Senior Design Final Project Document
December 4th, 2017

Portable Watering Device

Department of Electrical Engineering and Computer Science
University of Central Florida
Senior Design 2
Dr. Lei Wei

Group 9

Chris Havekost | Computer Engineering
Joan Henriquez | Computer Engineering
Peter Nachtigal | Electrical Engineering
Ronak Patel | Computer Engineering

Table of Contents

List of Figures

List of Tables

1. Executive Summary

Anyone who has a garden, loves their plants. The last thing you would want is for your plants to become unhealthy because you went on a business trip or vacation and had no one who could take care of them, or worse yet, you work so much and have such a busy lifestyle that you simply cannot take care of them properly anymore. Well, this is where our idea of the Portable Watering Device comes in to the rescue.

The Portable Watering Device, is well, a portable watering device targeted at people who live in apartment homes and have a little garden in their balconies. As most people know, when you live in an apartment home (especially a rental one) most often than not the landlord does not want any structural damage to the property, which translates to big holes on the wall, any device that could potentially start a fire, or even a sprinkler system for the plants in your balcony. This is where our device comes in.

Our device is a self-sustained system composed of a solar panel, a microcontroller with a few sensors behind it and lastly, a water tank and pump. The solar panel is used to charge a battery on the device which is used to power the system, meaning that the device does not need to be plugged into anything unless there is no sunlight for a couple days. The microcontroller behind the solar panel is connected to a few sensors (temperature, humidity, pressure, water tank level, etc.) which are used to predict when the best time to water the plants will be. Using our advanced algorithm, the microcontroller is able to detect when the best time to water the plants will be throughout the day.

Installation of the Portable Watering Device is very simple. All the user has to do is pick a spot on the balcony rails where the solar panel will get the most sunlight and place the device there. Then they will adjust the sprinkler nozzle so that it is aiming towards the plants. The last step is to take out their phones and install the app that comes with the device which allows the user to monitor each of the sensors, check the water level remaining in the water tank, enable the automatic watering mode, or if they prefer, they can set manual settings based on a preferred time of day, humidity level, or even temperature. The user will be able to access those settings from anywhere once the Wi-Fi Module in the PCB is paired to their home network, given them peace of mind knowing they can water their plants from anywhere at any time if they wish to do so.

2. Project Description

With the creation of an automated and self-sustaining watering system the need to continuously check up on plants or have people come over and water them while you're away significantly decreases. This not only will save the consumer's time, but money as well.

This section goes over:

- The motivation behind the design and application of the project.
- What the team plans to accomplish with the creation of this device.
- The requirements and specifications that must be met in order to achieve a satisfactory product.
- That house of quality analysis of the project.

2.1 Project Motivation

Planting is a very popular hobby, many people like to have plants at their residence for growing food, landscaping, or making their home look beautiful. The people who own a house have easy access to all the sources required to maintain the plants such as a water sprinkler system. However, the people who rent an apartment tend to grow plants on a balcony and have no access to an automatic water sprinkler system that waters the plants, and due to rental agreement restrictions, the residents are sometimes not allowed to plant outside. Here, the house owner could set the timer on the sprinkler system as desired, but the apartment owner does not have this advantage and they would have to water the plants themselves.

When we look at large fields, farms, or grounds full of crops, grass, or plants, we realize that they are well maintained, at least in the United States. This is due to large sprinkler systems that are costly, not only when they're initially set up in fields but also when they are being used. These systems are not useful for users that have low-budget and small living space for just a few plants, so the primary focus here is to develop a low-cost, highly efficient system for these groups of people.

One of the group-mates grew mint plants in his apartment balcony and they were healthy plants until summer arrived. For a busy college student, he or she can water their plants at most once a day due to education or work constraints. When it is too hot, watering plants once a day isn't enough for the plants to survive. This is a problem, we brainstormed to find a solution for this problem, and we thought what if there was something that could water the plants for us when we are out at work or school. We thought of a solar powered and portable water sprinkler system that is smart enough to understand the weather conditions and waters the plants independently. We thought portability was important because people who rent are more likely to move to another apartment frequently and it will be best for them if they were able to move the system with them easily rather than getting a new one. So, the difficulty of growing plants in an apartment balcony motivated us to build a solar powered and portable water sprinkler system.

2.2 Goals and Objectives

For the people who are unable to stay at home and take care of their plants as required, our goal is to present an automated water sprinkler system that is easy and simple to set up. This system allows the users to manage the system via a mobile application. However, the system is smart enough to understand the weather conditions and water the plants accordingly without an action taken by the user. The application notifies the user every time the system takes an action, and if it is almost out of water, so that the user knows when to fill up the tank. The goal here is to create a solar powered system that uses a solar panel and consumes very little power, this is so that the user doesn't have to worry about wires. We have made the system easily portable by taking advantage of the system's simplicity, which makes moving from one apartment to another very easy.

2.3 Requirements Specifications

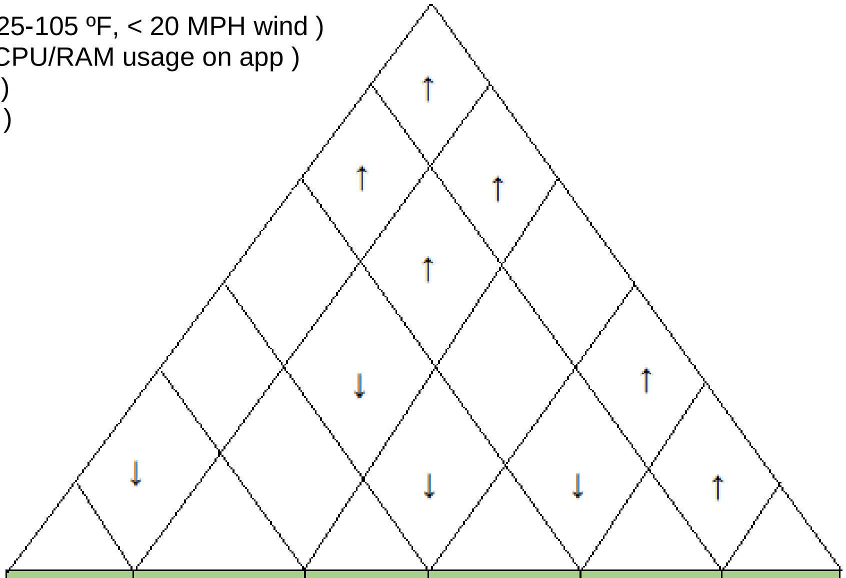
1. Water pump shall have enough pressure to force the water out to a nozzle.
2. Nozzle shall be able to propel water at least 5 feet and in a cone formation.
3. Water pump shall be low-voltage drawing to save on battery power.
4. Water pump shall be low-current drawing to save on battery power.
5. Water pump shall be switched on via GPIO and relay circuit.
6. Power circuit shall step down voltage to 3.3 volts.
7. Power circuit shall supply enough current out to sustain sensors and microcontroller.
8. Power circuit shall use MicroUSB input for voltage.
9. Solar panel shall be lightweight and small in size.
10. Solar panel shall output enough energy to charge a battery and power the device.
11. Solar panel shall endure the elements (weatherproof).
12. PCB shall be large enough to handle all the sensors and solar panel equipment soldered on.
13. PCB shall be small enough to be portable.
14. Microcontroller shall be able to handle all the input and output from sensors.
15. Microcontroller shall be able to drive a water pump.
16. Microcontroller shall be able to accept power from battery/solar panel power supply.
17. Microcontroller shall be small enough to fit into a portable container.
18. Microcontroller shall be able to enter low-power mode when not in use.
19. Temperature sensor shall be able to accurately detect surrounding temperature within ± 5 degrees Fahrenheit.
20. Barometric sensor shall be able to accurately detect atmospheric pressure.
21. Humidity sensor shall be able to detect humidity from 0 – 100%.
22. Depth sensor shall be long enough to reach bottom of the tank.
23. Depth sensor shall send data that water is low.
24. Depth sensor shall talk through GPIO.
25. Water tank shall hold at least half a gallon of water.
26. Water tank shall be made from inexpensive material.
27. Battery shall be able to power the device while there is no solar power.
28. Battery shall supply enough current for entire device.
29. Device shall be able to charge via USB when there is no solar power and a charge is needed.

30. Device shall be small enough to be considered portable.
31. All device components shall be within main unit (except for water tank and solar panel).
32. Device shall be easy for anyone to assemble.
33. Device wall-mount shall be sturdy enough to hold device.
34. Device wall-mount shall be able to sustain device for long periods of time.
35. Device wall-mount shall be efficient enough as to cause the least amount of damage to the wall it is being mounted on.
36. Device wall-mount shall be lightweight to maintain the portability of device.
37. Device shall have mechanism to adjust orientation of solar panel.
38. Device shall have a smartphone app to go along with it.
39. Smartphone app shall allow input from the user.
40. Smartphone app shall have an easy to use UI.
41. Smartphone app shall at least run on Android.
42. Device shall have two different watering modes, automatic and manual.
43. Device shall be able to communicate with smartphone app via Wi-Fi.
44. Wi-Fi Module shall have low power consumption.
45. Wi-Fi Module shall communicate through UART.
46. Device shall not cost more than $500 to build.

2.4 House of Quality Analysis

ESTIMATES:

- Weather Resistance (25-105 ºF, < 20 MPH wind)
- Performance (< 15% CPU/RAM usage on app)
- Setup Time (< 5 mins)
- Spray Distance (~ 5 ft)
- Power (< 10 W)
- Cost (~ $350)



		Low-Power	Performance	Setup Time	Portability	Spray Distance	Cost
		-	+	-	+	+	-
Cost	-	↑↑	↑↑	↓	↑	↑	
Functionality	+	↓	↓↓	↓	↓↓	↑	↑↑
Durability	+	↓	↑	↑	↓		↑↑
Weather Resistance	+		↑↑	↑	↓↓		↓

↑ Positive ↓ Negative ↑↑ Strong Positive ↓↓ Strong Negative

Figure 1: House of Quality Analysis

2.5 Hardware Flow Diagram

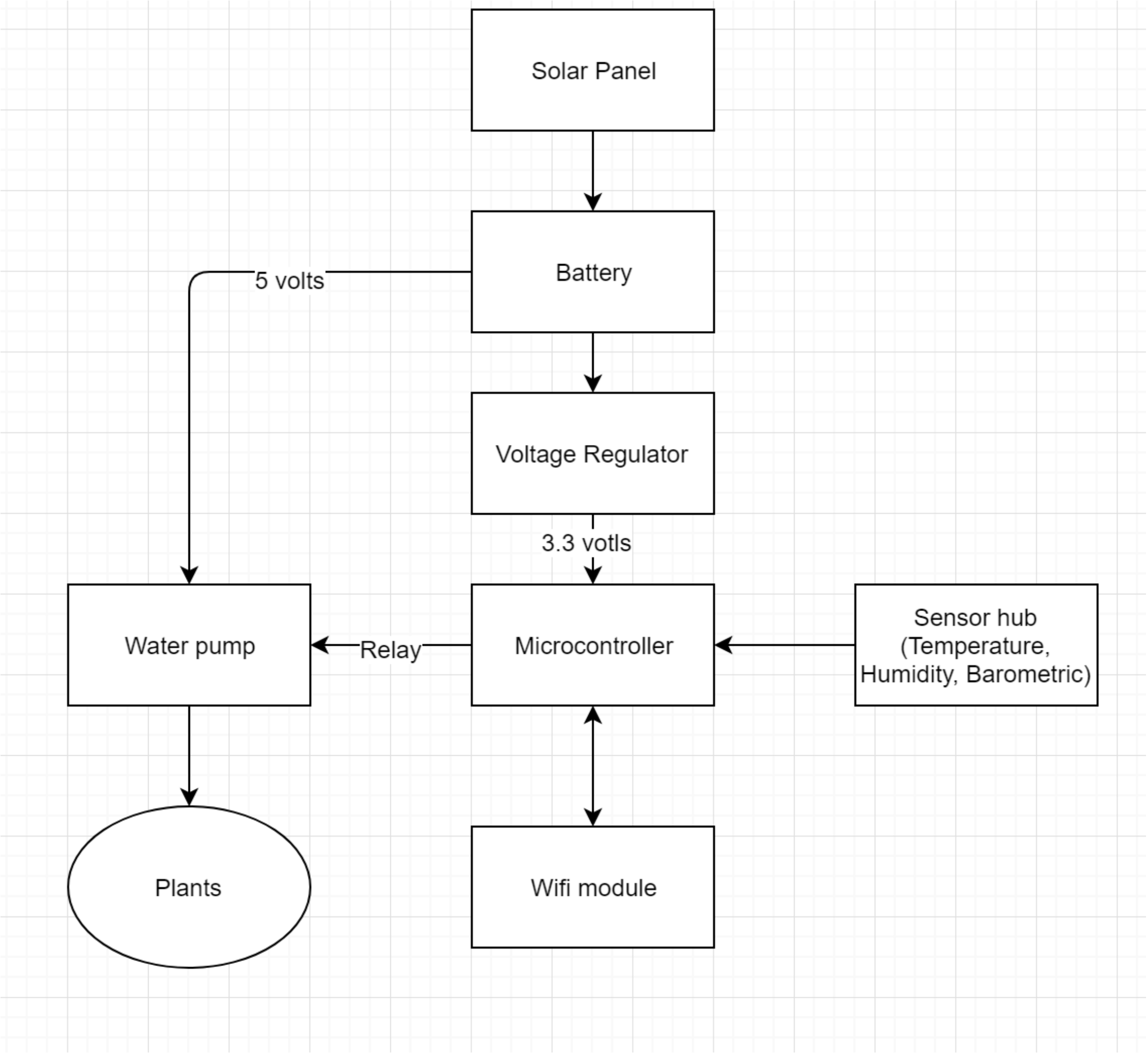


Figure 2: Hardware components flow diagram

The figure above is an estimate of how the PCB and the various components will be connected and how they will interact with each other.

2.6 Physical Device Design

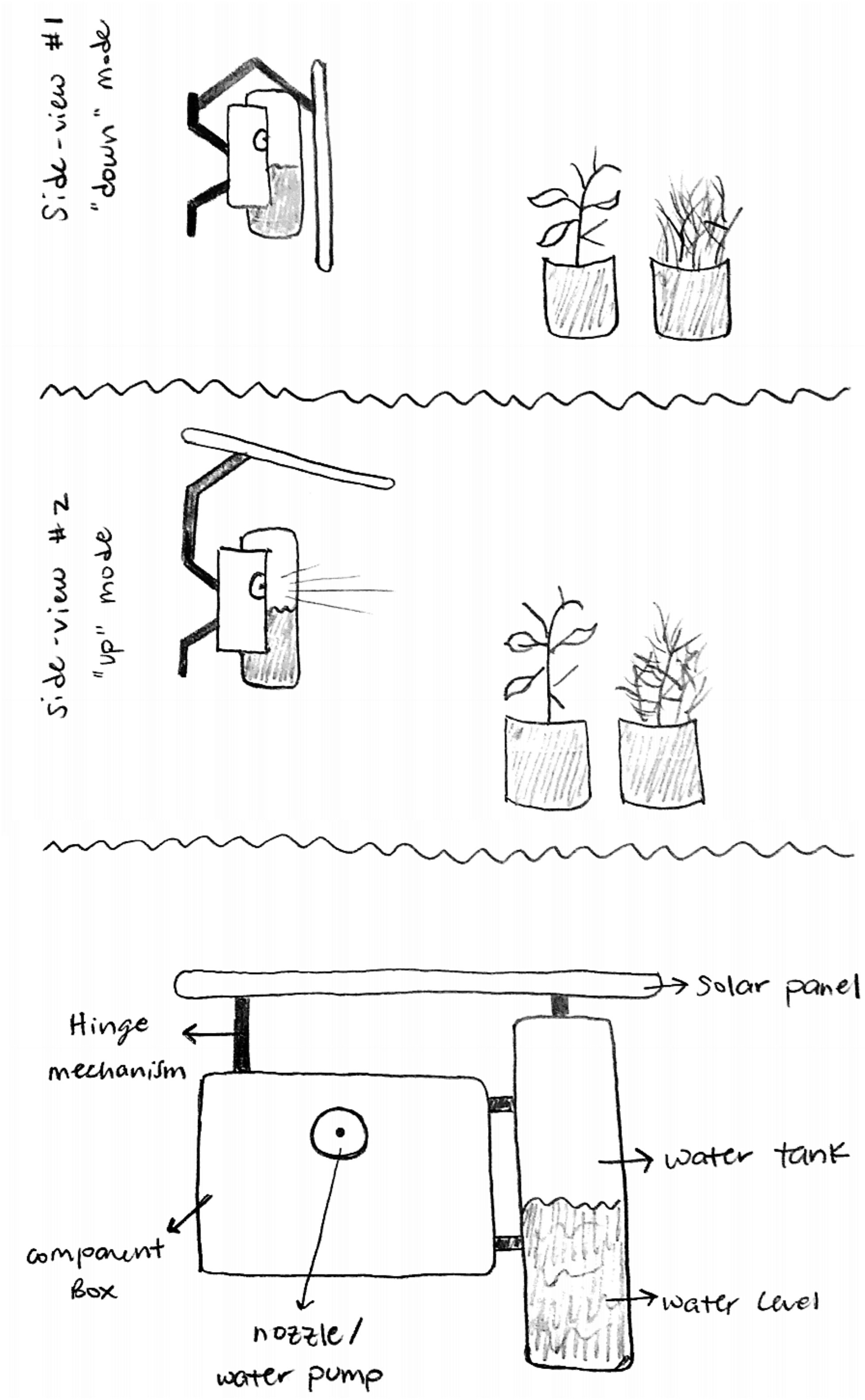


Figure 3: Physical Device Sketch

As seen in the sketch above, our portable watering device consists of four main physical components. The first component is the rectangular box that houses the water pump and the PCB with all the supporting sensors. This box makes sure that all the electrical components are safe and protected from all the outside elements (i.e. water, dust, excessive wind, etc.).

The second major component is the water tank. This tank is attached to both the wall mount and the PCB box. The wall mount gives it structural support to stay in place and the PCB box connection allows the water tank to talk to the PCB with its integrated water level sensor. For the first few prototypes, the water tank is a fixed height and width determined by us. However, in the future we hope to replace that with a modular system that can adapt to a variety of bottles so that is it up to the user what kind of size tank they want to have.

The third component is the solar panel. The solar panel is the biggest physical component, and also the most expensive, but we'll get to that later. This component is attached to a hinge mechanism that allows it to move up and down. This mechanism allows the entire device to have a smaller footprint when in its "down" position. This smaller footprint allows the device to be easier to carry and move around and also to package if we want to think that far ahead. In the "down" position, the solar panel sits right in front of the "component" box and the water tank. In the "up" position, the solar panel sits on top of the component box and the water tank at an angle. This will no longer block the water pump and allows the solar panel to receive the highest amount of solar energy.

The last component is the wall mount itself. The wall mount has three main attachment points on the side of the device. The first one is where the component box attaches to the mount. The second is where the water tank attaches to the wall mount. And last but not least, is where the hinge mechanism attaches to the wall mount. Our goal is to have as little structural damage as possible to the mounting wall since this device is mostly targeted towards apartment homes which usually need to be put back to their original state upon moving out. With that being said, the device still needs to have enough structural support to hold all the other components for an extended period of time.

3. Research

Since people today are becoming more and more busy with their work lives and have less and less time to do tasks that are important back at home, it has become necessary to create technology that can help do this work for us. Automation at home has become more accessible with the development of IoT (internet of things) devices. This development has allowed companies to create devices that can automate home gardens across the world simply.

This section goes over:

- Products that currently exist on the market as well as do-it-yourself models created by hobbyists.
- Relevant technologies that deal with the project and research about them.
- Comparison of the types of technology in order to select the best components for design.
- Selection of parts based upon the comparison research.

3.1 Existing Projects and Products

Commercial products for self-watering plants come in many shapes and sizes. They can come in the form of a water tank that has a hose and has to be attached to the pot the plant is in itself or come in the form of a flower pot that can water the soil it contains itself (Figure 3). The designs for the products range from small and simple to large and complex. Our goal is to get a balance of both of these designs, small but complex.

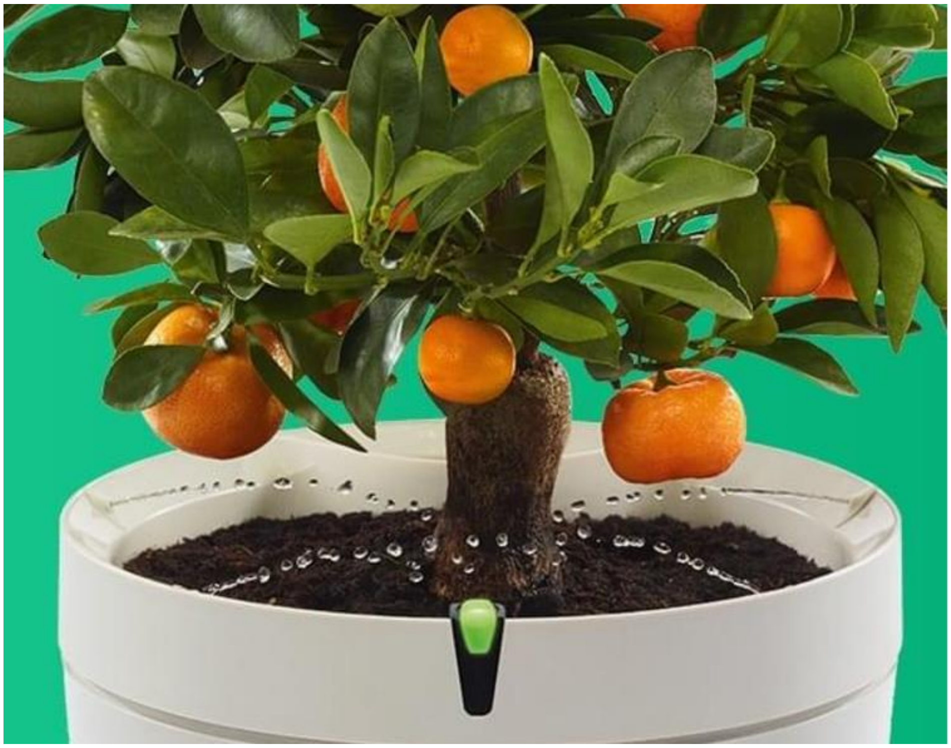


Figure 4: Commercially sold self-watering pot

Source: https://www.parrot.com/us/connected-garden/parrot-pot#parrot-pot-details

The do-it-yourself community has also taken a swing at the self-watering plant systems by making their own homemade designs. The two most popular use the already put together systems of an Arduino or Raspberry Pi (Figure 4) and program them to work sensors from their GPIO on board ports. This design is very similar to the way we wish to build our system but instead of having a bulky already designed system of an Arduino or Raspberry Pi we soldered our own board together and created our own all-in-one board that utilizes the sensor we need and reduces the size as much as possible.

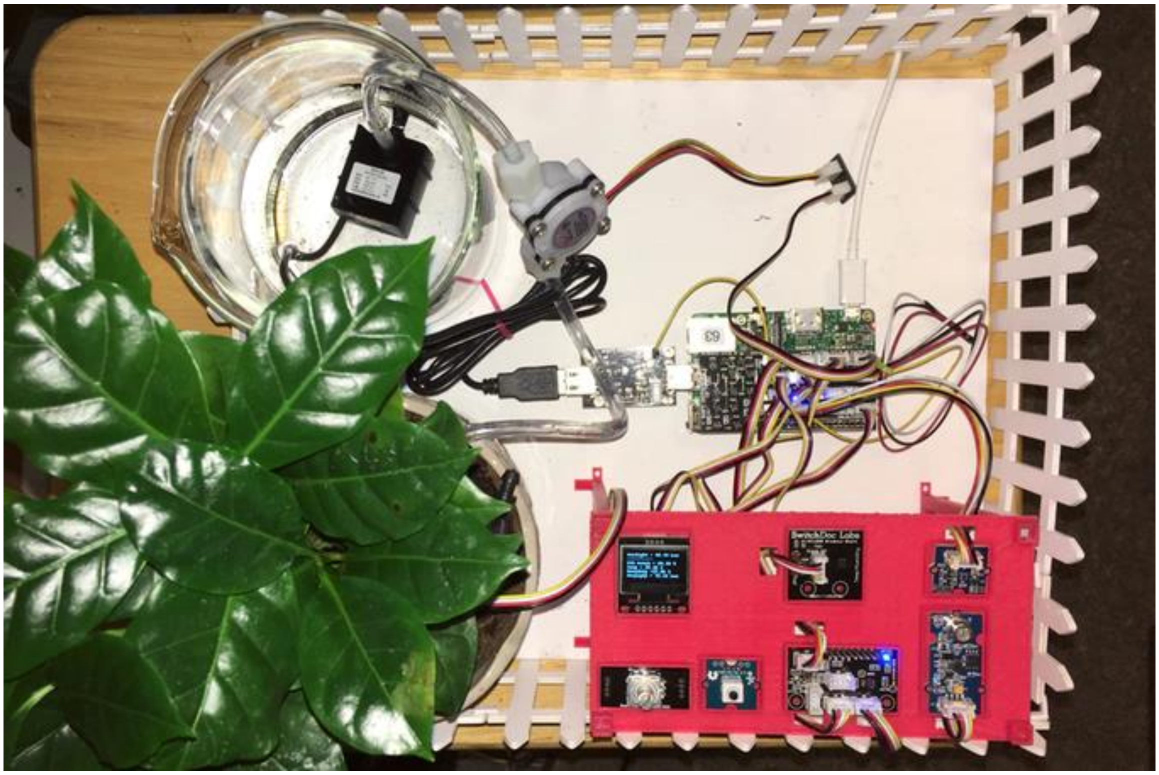


Figure 5: Homemade IoT-based plant watering system

Source: https://www.kickstarter.com/projects/sunair/smartplant-pi-smart-garden-with-your-raspberry-pi

3.2 Relevant Technologies

The following section deals with the relevant technologies of our project and the research done on them. In the research, it is necessary to point out all relevant functions of the technology and the importance of the parts of the components. This research should also lead to new knowledge that can help expand or improve upon the project.

3.2.1 Microcontroller

The Microcontroller that will be selected must follow some guidelines in order to be considered for this project. Some of those include: a low-power mode function, a fast wake up/response time, the ability to handle multiple inputs, and have a fast-enough clock-rate to handle up-to-date information. The low-power mode on the microcontroller allows the device to have a sleep function when it is not ready to be used to save power and know when to wake up without an alert from a user to start up. The fast wake up or response time is used to make sure the plants are watered without a delay each day. The system being designed handles multiple inputs from different sensors, it is a major necessity to handle these inputs without a problem. A fast clock rate allows the controller to check each sensor as well as data sent into the chip quickly and accurately, the faster the rate the faster the refresh of each sensor and the data. Each of the controllers below have been selected in order to fit the needs mentioned.

Low Power Mode

Low-power mode is designed to have a microcontroller unit run as efficiently as possible. Power leakage and dissipation through the controller has been significantly reduced since the devices were first designed. As technology becomes more advanced the ability to operate microcontrollers using less energy has become easier to develop. [1]

By turning off the clock in sections of the processor that isn't being used lowers the power being used by that section. As well as higher performance transistors that have less leakage and low voltage functionality, the microcontrollers of today have significantly reduced power consumption.

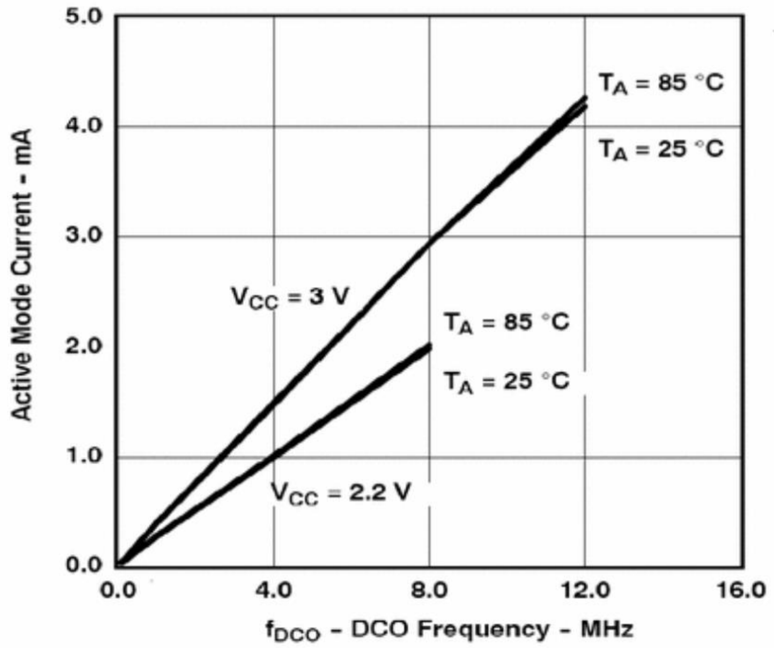


Figure 6: Current draw compared to frequency of clock

Source: http://www.ti.com/ods/images/SLVSAF6A/g_iam_fdco_las694.gif

In the figure above, the correlation between current drain (power consumption) and the increasing frequency is linear. By reducing the frequency of the clock, we are able to limit the Active mode drain current. Using this logic by putting parts of the CPU to “sleep” using a low-power mode function we are able to reduce the rate at which the clock cycles. If the section of the CPU is up all it needs is a “wake up” trigger and the clock rate can increase to activate it.

Some of today’s microcontrollers have been able to lower down current pull to 9 nanoAmps and run currents down to almost 30 microAmps/Hz. With typical rechargeable batteries readily available the ability to keep microcontrollers running long and stable over time has become a thing of ease.

Microcontroller Communication

The microcontroller has many forms of communication to take input from a user as well as output to the user. Some of these are SPI, I2C and UART. Each of these three types of communication use different forms to transmit their information. [2]

Some of the aspects to look at for these forms of communication are whether it is parallel or synchronous. Parallel types are able to transmit bits of data at the same time usually in groups of 8 or 16, and relying on busses to transmit. Groups of data are sent at the same time at the edge of a clock pulse. Synchronous forms of communication use one wire and send bits at the pulse of a clock one bit at a time. While parallel is able to send data faster, it is more costly and takes up more pins for the form of communication compared to the synchronous line of communication.

Asynchronous communication also exists, this means that the data can be sent and received without the need of an external clock. This style helps reduce the amount of lines and pins needed for communication. This requires more precision in order to transfer the data carefully though.

SPI	I2C	UART
Sends a lot of data at once with running the risk of losing some data and un-syncing it	Safely sends data one bit at a time based off the internal clock cycle	Sends data as a parallel shift between two communicating UARTS
Requires some precision because the noise from other lines may interfere with others	Least precision required less lines means there will be less chance of noise interference	Requires more precision, noise is relevant here as well since there are more wires
Uses internal clock	Uses internal clock	Does not rely on clock
Uses 8 lines to send 8 bits and one line for clock	Uses one line for data and one line for clock	Uses 8 lines for data transfer
Supports 10-20 Mbps	Supports 100KBps-3.4Mbps	Supports up to 430 Mbps depending on standard in both UARTS

Table 1: Serial communication comparison

Microcontroller General Purpose Input/Output

For the project being designed input is taken from sensors on board the printed circuit board and output comes from data sent to an application and water being sprayed. This form of input and output can only be handled by pins registered to the general-purpose input/output ports on the microcontroller.

The general-purpose input/output, or GPIO, ports typically have eight pins that can handle input and output. These pins are able to be set by the user to be input or output, nothing is set permanently by the controller. GPIO pins allow peripherals to be added to the device and increase the functionality of the microcontroller on the PCB. [3]

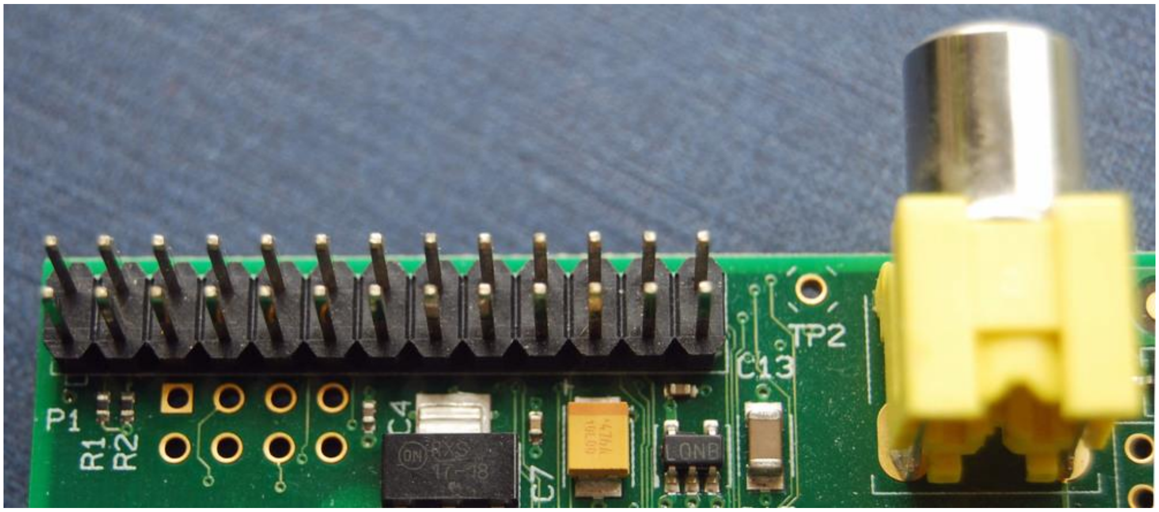


Figure 7: Standard PCB pin headers for GPIO pins

Source: https://www.raspberrypi.org/documentation/usage/gpio/

General purpose ports handle voltages that can be high or low to represent ones or zeros in data transfer. Physical components such as temperature sensors only deal with forms of voltages, or in analog, and it's the responsibility of the controller to do an analog to digital conversion. This means it takes the analog voltage that is sent into the pin and converts it into a digital representation of zero (low) or one (high).

Arduino Uno is the board that offers GPIO and this is the board that is used for running this system as it powers the sensors and other modules of this system, this board acts as a source of communication between all of these important components of this system. By exploring this microcontroller, we learned that:

1. Initially, microcontroller's GPIO pins have a default voltage signal that they output when told to turn on. The value of this voltage signal depends on the maximum power supply supported by a board. Arduino Uno GPIOs by default shootout a 5V signal. Therefore, when a programmer uses the Arduino IDE to describe a particular GPIO pin as a 'HIGH' it will have an output of 5V, and if described as a 'LOW' the pin will have an output of 0V.

2. For this system, we needed different types of components with different specifications. Here, if a system needs less than 5V supply and if we deliver 5V then we will damage the component. For instance, some component might need a 3.3V supply to operate correctly, now we have learned that we can't use the Arduino IDE to program a GPIO output of 3.3V. We found that there are different ways to make this work, by building different circuits on the breadboard with the GPIO pin as input. As discussed previously, a voltage regulator is used to this job and having it in the circuit would definitely prevent some safety issues and will be more reliable than other circuits, because if we completely rely on a circuit such as below then varying current passing through the resistor and the zener diode can have an impact on the life of the circuit. Also, this circuit would take a lot of breadboard space unlike a voltage regulator that does the job efficiently by taking very less space.

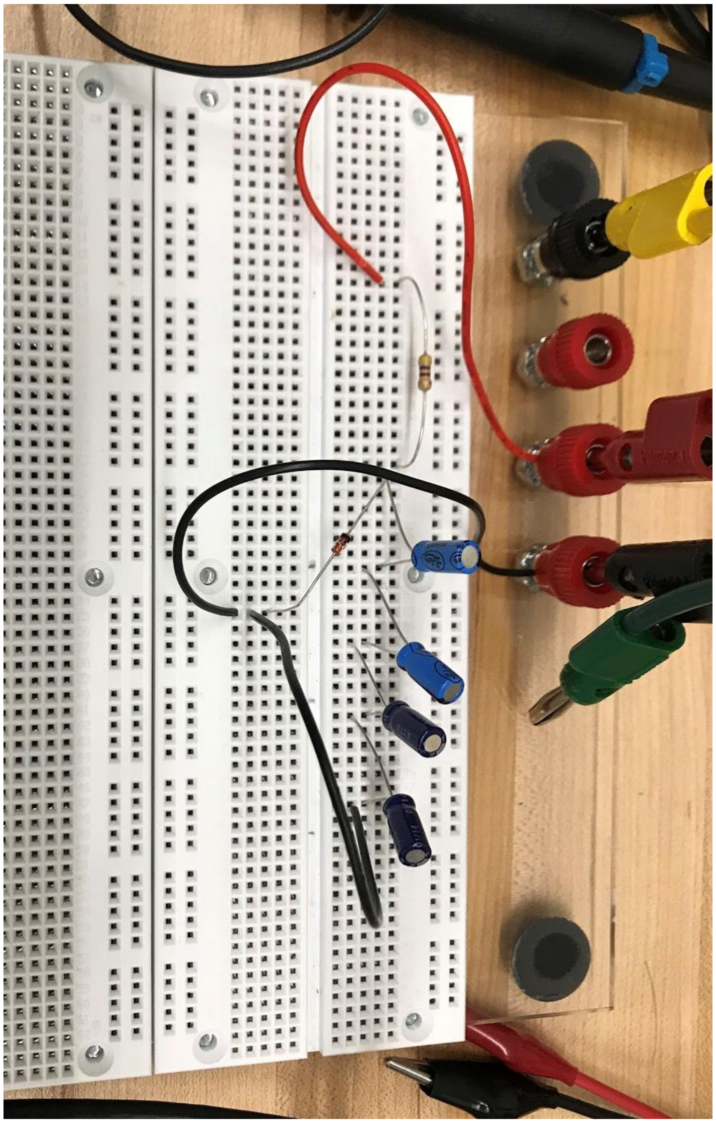


Figure 8: Breadboard testing of 5V to 3.3V

Before connecting these circuits to the actual components, these circuits were first tested on the breadboard. Power supply was used to generate an input of 5V, and voltmeter was used to verify 3.3V at the output.

Microcontroller Architecture

Major processors today use the layouts of their internal components to become more efficient and faster, this is known as the computer's architecture. There are two architectures mainly used in modern technology. The complex instruction set computer and the reduced instruction set computer, or CISC and RISC architectures respectively.

CISC architecture utilizes mainly hardware to complete its program. This helps reduce the amount of lines of assembly code necessary to create the program using complex instructions. Simple instructions are combined in designed ways to form more intricate instructions. For example a multiplication function uses typical add and shift instructions without having to type them out.

The benefit of CISC is having a multi-clock cycle which allows for faster execution of code with its reduced amount of lines. Unfortunately this design also takes up a lot of space. Using memory to store these complex instructions means that space is reduced on the chip so it loses room for other functions like communication functions or GPIO ports. [4]

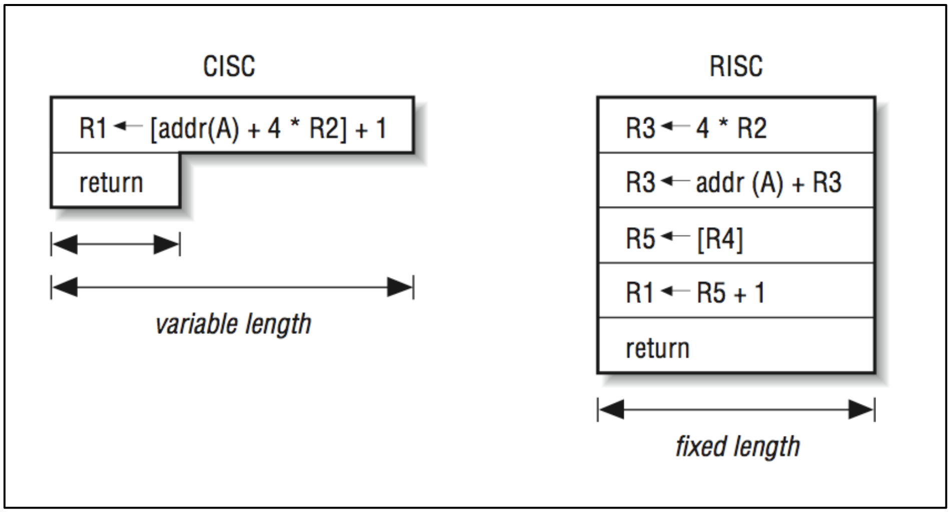


Figure 9: Instruction set architecture comparison

Source: http://archive.cnx.org/contents/818b2af4-c1a1-422e-9c47-b5c2fb0f8d9a@3/what-is-high-performance-computing-fundamental-of-risc

RISC architecture, on the other hand, uses simple instructions and relies mainly on the software to complete its functions. This means that the clock design runs on one clock and has a smaller clock that executes one instruction per clock cycle. This architecture means that each instruction has to be written out which means the length of the code will increase.

The space on RISC based processors allows more space for non-architecture based hardware to be incorporated. This includes communication and GPIO pins. Since this is a single datapath that executes over each cycle it is possible to build in a pipeline which has instructions executes parts of instructions in parallel.

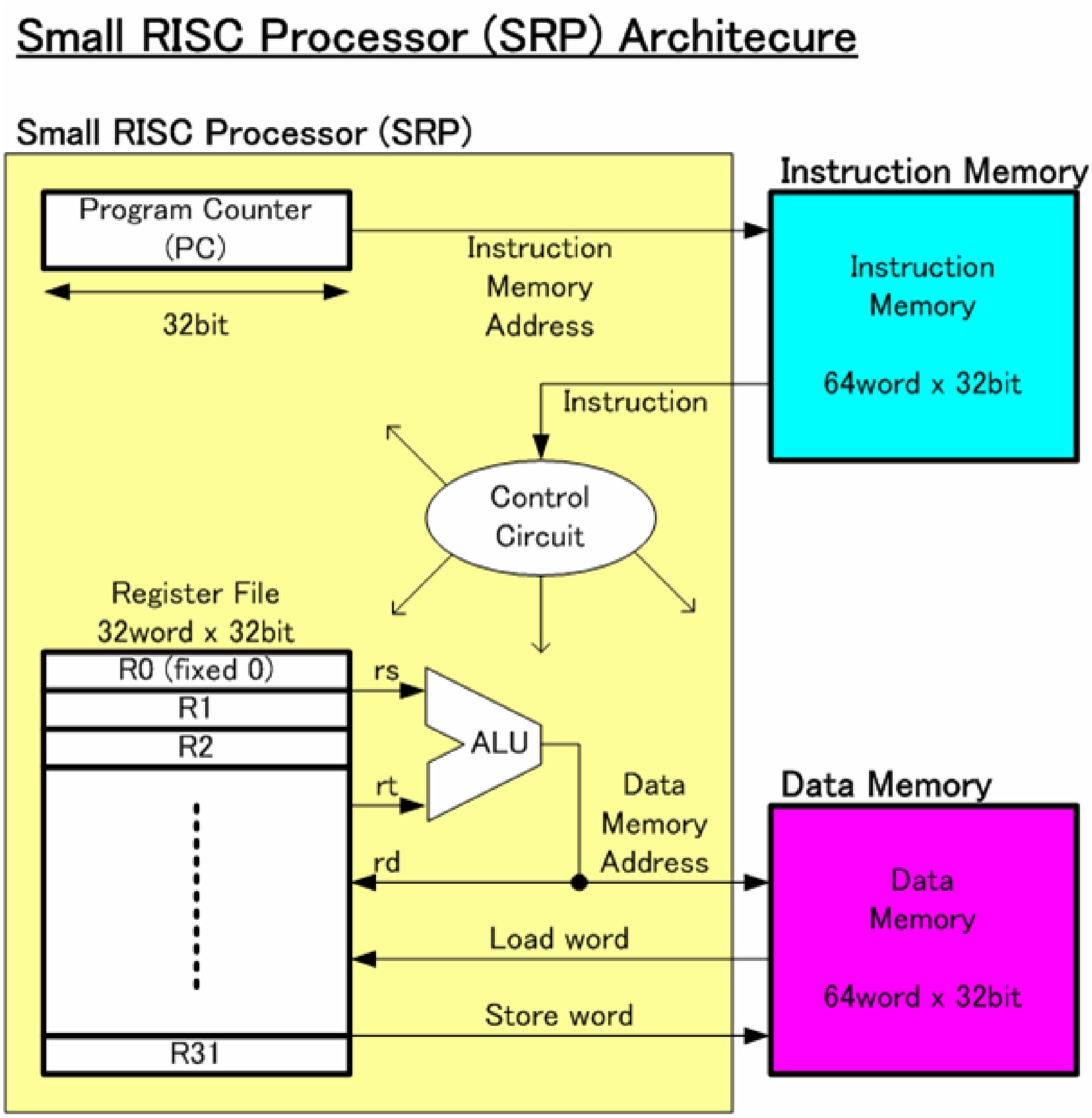


Figure 10: Basic layout architecture for RISC Processor

Source: http://www.lsi-contest.com/2009/siyou-1e.html

Each architecture has benefits and downsides but the research done in this section has helped to determine the architecture that can provide us with the best outcome. While the CISC can complete very complex functions and programs it is not necessary for the amount of work our processor needs to handle and therefore should not be used in our project.

CISC	RISC
Hardware based	Software Based
Many instruction sets	Small amount of instructions used to create more complex ones
Less registers used	More registers used
Multiple cycles to execute one instruction	Instructions executed in one cycle
Uses microprogramming to convert languages between levels	The conversion in languages is added in the compiler not hardware

Table 2: CISC vs RISC architecture comparison

The simplicity of a reduced instruction set computer is perfectly suited to the tasks we need. The reduced instruction size also means that the size of the chips we have selected is smaller compared to the CISC architecture chips since most of their instructions are executed with extra hardware. This includes the need for storing data, writing data, reading data and performing our output of driving a water pump. With this information, we can begin to narrow down the search for our MCU.

3.2.2 Solar Panel

Solar energy is proving to be a very good source for powering machines today. We can observe that many companies, schools, and organizations are trying to be economical and environmental friendly by creating and taking advantage of solar powered systems. The solar panels today are used in both large and small scales; big solar panels are used in large scale projects that require lots of power like powering houses, and small solar panels can be used in small scale projects like powering a small battery that powers a phone. For powering the sprinkler system, we are using a small solar panel as it generates enough power for the system to operate. We also use a battery to power the system in case the weather conditions don’t allow the solar panel to generate enough power.

In order to decide which solar panel type is best for the sprinkler system, we studied various different kinds of solar panels available that have different manufacturing processes and use different materials. Because they use different materials, efficiency results vary as some elements gather heat and convert heat to electricity faster than other elements, and also that some save more power than the others. With a solar panel generating power to operate this system, we also researched batteries as a backup to power this system in case of poor weather conditions where there is lack of sunlight.

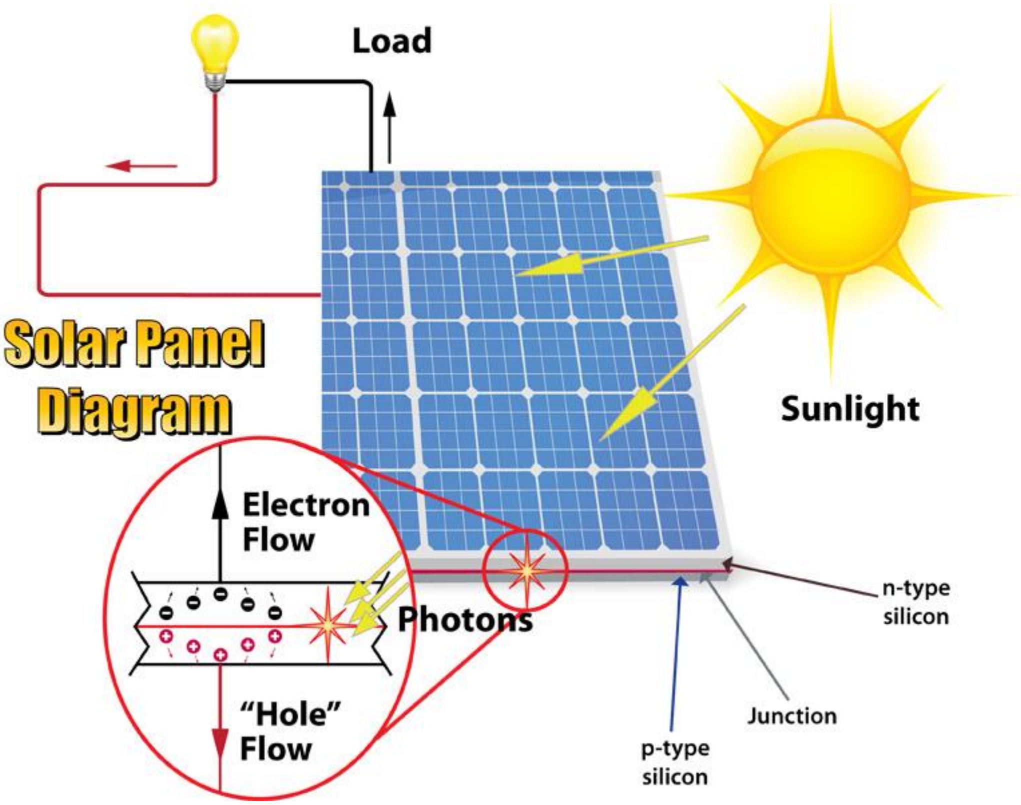
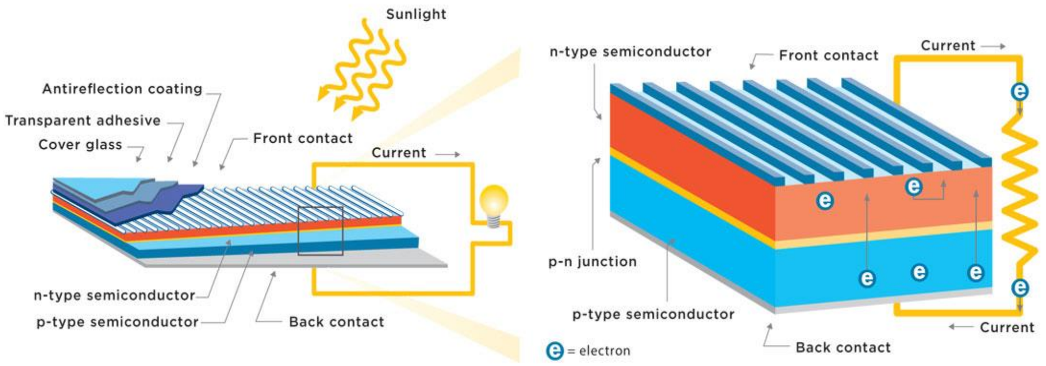


Figure 11: Composition of solar panel

Source: https://etap.com/renewable-energy/photovoltaic-array-fundamentals/

Building a Solar Panel

So, what is a solar panel? Many smaller units called photovoltaic cells connected to each other create a solar panel. Similar to what used in microelectronics, a photovoltaic cell is also made up of two slices of silicon. In order to work; to create a electric field, the two slices of silicon need to generate opposite poles of charge. To do this, material science engineers mix both the silicon slices with other elements to get some charge (positive or negative) that allows the cell to have proper conductivity, this process is called doping. To be specific, generally the top layer of each of the silicon slices is dipped in phosphorous to add more electrons which have a negative charge (N). Now to create opposite poles, the bottom layer is dipped in boron to have less electrons creating a positive charge (P). This creates an electric field between the silicon slices; p-n junction, which means that when the photons of sunlight hit the solar panel or a photovoltaic cell, the electrons in the field will be pushed out to the silicon layers and generate motion in electrons; current. This current flow from the p-type layer to the n-type layer. Each photovoltaic cell generates very little power, panels are formed of multiple cells interconnected so more power can be delivered. However, to make the solar panel work as a useful source of energy, the cells on the side of the panel are connected to the wires that act as an outlet to connect to any other device; like a battery or motor. This is how heat energy is converted to electrical energy by a photovoltaic cell.



Solar cells are composed of two layers of semiconductor material with opposite charges. Sunlight hitting the surface of a cell knocks electrons loose, which then travel through a circuit from one layer to the other, providing a flow of electricity.

© AARON THOMASON/SRPNET.COM

Figure 12: Role of Photovoltaic Cell

Source: http://www.ucsusa.org/clean-energy/renewable-energy/how-solar-panels-work#.WV4h5jOZORt

Types of Solar Cells

As mentioned earlier, there are different types of solar cell technologies available, each manufactured differently and having different materials used. There are three types of solar cell technologies available today which have their own advantages and disadvantages.

Single-Crystal/Monocrystalline Cells: These are thin ingots of silicon and cylindrical in shape. However, they are not perfectly cylindrical since four sides of these ingots are cut out to make a silicon wafer, this done to obtain optimal performance and lower cost of a single cell as the cost of a cell is determined based on its size in terms of area, similar to microprocessors in computers. This process is known as Czochralski process. About ⅓ of the global market for solar panels consist of single-crystal solar panels. These cells have a very high efficiency mainly because of their manufacturing process and the use of the highest grade silicon. These cells are able to convert 15 – 20% of all solar energy received to electrical energy. Thanks to their efficiency, they require less space than other solar panels which makes them perfect for small scale projects. Panels containing these cells are usually the most durable and live the longest compared to the other types of solar panels. They also perform better than other solar panels in low sunlight conditions.

The main drawback to Single-Crystal/Monocrystalline cells is their price. Since these panels provide the best performance, they are also the most expensive. Also, the entire circuit in the panel can break down if dirt, water, or snow gets in there. While cutting the sides of the cylindrical ingots improves efficiency, it also created large amounts of wasted silicon. And lastly, these panels are not as efficient in colder weather as they are in warm weather.

Polycrystalline Cells: Unlike the monocrystalline solar panels, polycrystalline solar panels don’t go through the Czochralski process, which helps them waste less silicon. In this case, a square shaped mold is filled with melted raw silicon, then it is cooled to be cut to form perfect square silicon wafers. Polycrystalline panels make up to ½ of the global market of solar panels. This manufacturing process is simpler and cheaper than the one needed for monocrystalline cells. This is an ideal solar panel for homeowners since it requires less investment compared to the other cell types and still does a decent job. The main drawback of these cells is that they have a lower ability to convert solar energy (about 13 – 16%) because of their lower silicon purity. This also means that they require more panels in order to match the output of monocrystalline solar panels, which means more they take up more space.

Thin-Film Cells: The manufacturing process for these cells is a quite unique. It is spraying or depositing photovoltaic materials on glass or metal surfaces in thin films, which creates a complete module at a time instead of connecting and assembling cells individually. There are different types of thin-film cells and they can be categorized depending on the combination of photovoltaic materials used; Amorphous Silicon, Cadmium Telluride, Copper Indium Gallium Selenide, Organic photovoltaic cells. The efficiency of these panels depends on the combination of the photovoltaic materials used. Thin-cell solar panels are about 10% of the global solar panels market. Manufacturing large amounts of thin-film panels is simple, which makes them cheaper than monocrystalline and polycrystalline solar panels. The thinness of these panels enables them to be flexible and more visually appealing. High temperatures and shade on the panel have a very low impact on efficiency and performance of the unit.

Like the other panels, thin-film cell solar panels also have a few drawbacks. The first one is their efficiency per area. This solar panel would not be very efficient, for example, for a homeowner because they require a lot of space. In the same amount of space, a monocrystalline solar panel can generate almost four times the power that a thin-film cell solar panel would. The more space required, the more the cost of the equipment needed to operate these panels (cables, support structures, etc.). Lastly, the longevity of a thin-film panel is shorter than that of monocrystalline and polycrystalline panels. Manufactures also provide shorter warranties for this kind of panels.

In the table below we compare all the solar panel types to discover which type is the most cost-effective and efficient for this project.

Solar Cell Type	Advantages	Disadvantages
Single-Crystal/Monocrystalline Cells	1.Highest efficiency 2.Space Efficiency 3.Life Longevity 4.Better performance in low sunlight conditions	1.Expensive 2.Not resistant to outside elements 3.Creates large amounts of silicon waste 4.Only efficient during warm weather
Polycrystalline Cells	1.The manufacturing process is simpler and cheaper 2.Wastes less silicon than monocrystalline silicon 3.Ideal for homeowners	1.Lower ability to convert energy 2.Require more space 3.Not as visually appealing
Thin-Film Cells	1.Cheaper 2.More appealing look 3.Flexible 4.High temperatures and shade on the panel have very low impact on efficiency and performance	1.Requires lots of space 2.The more the space required, the more the cost of the equipment needed to operate the panels; cables, support structures, etc. 3.Shorter life longevity and manufacturer warranty

Table 3: Solar panel technology comparison

3.2.3 Sensors

Sensors are used in our project to measure the relative humidity of the air around a plant, the barometric pressure, temperature of surrounding air, and water level of the tank. The readings from the humidity sensor are combined and analyzed with readings from the temperature sensor and barometric pressure sensor to determine whether or not the plant needs watering. To achieve this we added a range of acceptable times to water plants into our code. We need to keep in mind that a drop from high pressure to low pressure can indicate that rain is imminent, that a relatively high humidity reduces how frequently the plant needs water, and other combinations of temperature, pressure, and humidity. The water level sensor alerts users when the amount of water in the tank is running low.

To make our decisions as to which sensors we would choose for this project, we considered many different factors. One factor we looked at was whether the sensors produced an analog or digital output. An analog output is favored for its accuracy, its signal directly corresponds to its voltage level and it can output any number in its operating range of voltages. Digital signals have discrete values that can be thought of as steps. One digital step can correspond to a certain range of analog values and have its output be the same over that entire range, not changing steps until the analog signal jumps above or dips below some predefined value. We also value stability of readings over long term use, so it was important to find out how each sensor's measurements change over the course of years. In particular, Honeywell humidity sensors experience a very small change in their output of relative humidity over five years. With only a 1.2% change, their stability in this category leads the industry.

Size is another important factor to consider when figuring out how to design your PCB. Many of the sensors we looked at were very close in size, however, there were some outliers. One example was the 083E temperature and relative humidity sensor. While it is much more accurate than other sensors we looked at, its sheer size means that it would be impossible to integrate it onto our PCB, and including it at all would severely diminish the portability of our project.

One of the most important factors we had to keep in mind when designing this project was power draw of the components. In a project where portability is a major focus and a solar panel is being used to charge a battery, power consumption needs to be kept to a minimum. Luckily, sensor manufacturers know that a large portion of the market cares about power consumption and many of them have focused on bringing low power sensors to market. Many sensors we looked at only consume in the range of ten to a few hundred µA when in use. When not in use, those sensors dial down to almost no power consumption. This is great for our project because it allows us to maximize the battery life and not have to constantly rely on solar or USB power.

Humidity Sensor

Our design incorporates humidity sensors so that we can keep track of the relative humidity of the air. We want to be able to measure the relative humidity of the air so that we can better control the flow of water into and out of the plant. After reading about a process called transpiration, where water is taken up through the roots of a plant and evaporated through its leaves, we learned that plant tissue can be damaged when there are sudden shifts in relative humidity, and plants are healthiest when in a certain range of relative humidity.

Aside from measuring the relative humidity of the air around plants, humidity sensors are used in a number of other things, including, but not limited to:

- Incubators
- Sleep apnea machines
- Ventilators
- Greenhouses

We were able to find humidity sensors ranging from 3mm x 3mm to a whopping 2.1cm x 16cm that were on average less than $3.26. We also researched the different types of materials that the sensors themselves could be made out of and found that each material had its own advantages and disadvantages. Thermoset polymer-based capacitive sensors perform well in almost all cases, except when the sensor is not at an ambient temperature. This is a rare case, but it does happen. They also provide better protection against certain chemical liquids, vapors, and allow the sensor to operate in higher temperatures. Capacitive relative humidity sensors are the only sensors that are capable of providing accurate measurements as the relative humidity approaches 0%. This gives them the widest range of operation and a commanding monopoly in their use for measurements. [5]

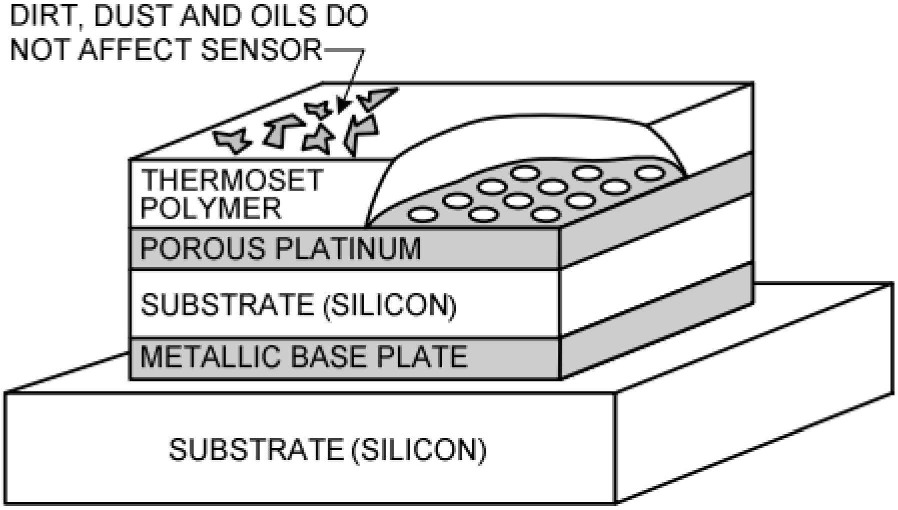


Figure 13: Sensor coatings

Source: https://sensing.honeywell.com/index.php?ci_id=49925&la_id=1

Another type of humidity sensor is called the thermoplastic polymer sensor. These sensors have slightly lower operating temperatures than thermoset polymers and a far worse resistance to vapors and chemicals in both gaseous and liquid forms. A third type of sensor we found made use of a Lithium Chloride film spread over a porous, usually ceramic, medium. The pores allow water vapor to be absorbed into the substrate, changing its conductivity and allowing the humidity of the environment to be measured. We found, however, that response times of these sensors are very slow and that they cannot be used in very humid environments without losing a lot of their accuracy.

So, in our opinion, we believe that thermoset polymer-based capacitive sensors are the best because of their operating range, superior accuracy, and their ability to rapidly detect changes in sensor capacitance. Their cost may be a little higher, but performance wise it is the best value, especially when considering that these types of sensors have great long term stability in their readings.

The chart below compares the three types of sensors just discussed, and it uses some terms that the reader may be unfamiliar with, so we have defined them here. RH stands for relative humidity. This is the amount of water vapor in the air divided by the total amount of water vapor the air can hold at that temperature, expressed as a percentage. The next term we would like to define is hysteresis. In this case it refers to the relative humidity value of the sensor lagging behind the actual relative humidity value of the air. If the new value of relative humidity in the air is very close to the old value that the sensor had then it will take some time for the new value to be reflected in the sensor’s readings. This may not happen within one cycle. As an example, thermoset polymers have a hysteresis range between 1% and 3%, so changes within that range will take longer to register in the sensor. The last term we will be defining is the risetime of the sensor. This is just how long it takes the sensor to reflect and change in relative humidity.

Active Material	Thermoset Polymer	Thermoplastic Polymer	Lithium Chloride Film
Substrate	ceramic or silicon	ceramic, silicon, or glass	ceramic
Changing Parameter	capacitance	capacitance	conductivity
RH Range	0% to 100%	0% to 100%	15% to <100%
RH Accuracy	±1% to ±5%	±3% to ±5%	±5%
Hysteresis	<1% to 3%	2% to 5%	very poor
Linearity	±1%	±1%	very poor
Risetime	15s to 60s	15s to 90s	3 minutes to 5 minutes
Operating Range	-40°C to 185°C	-30°C to 190°C	Varies. Cannot be exposed more than 3 to 5 minutes
Variance Over Time	±1% in RH over 5 years	±1% in RH every year	>1% in RH per °C

Table 4: Humidity sensor technology comparison

Temperature Sensor

Temperature sensing in one of the more important functions of our project. At its most basic level, we are able to measure the ambient temperature around the plant and adjust its watering cycle accordingly. Temperature sensors are used in almost every industry for many different types of control and fail-safes, they are not limited to merely measuring outside temperatures and relaying that information. Today, temperature sensors are commonly used in computers, smartphones, cars, and many other everyday objects. When temperatures become too high in a computer, for example, the hardware could become damaged. So while the sensor doesn't always provide meaningful data to the end user of the product, it does help keep that product safe by limiting the energy it can use once its internals heat up to a certain temperature or by turning that product off altogether. A big consideration in determining what would be a good temperature sensor for us was current draw. Not only would a large current draw from the sensor negatively impact the battery of the device, it would also impede the ability of the sensor to accurately measure the ambient temperature because the current it draws generates heat around it. Luckily, many of the sensors manufactured nowadays do not draw too much current, and many sensors we saw had a self-heating factor of about 0.2°C.

There are a few ways to design a temperature sensor, with four being prominent in the industry today. One way is to design something called a negative temperature coefficient (NTC) thermistor. These NTC thermistors have properties that exhibit large and precise changes in resistance that can be mapped to fluctuations in temperature. NTC thermistors are very accurate, usually within 0.05°C to 1.5°C, and because of their sensitivity they can quickly detect small changes in temperature and output accordingly. They have an operating range, from -50°C to 250°C, that is more than sufficient for our needs.

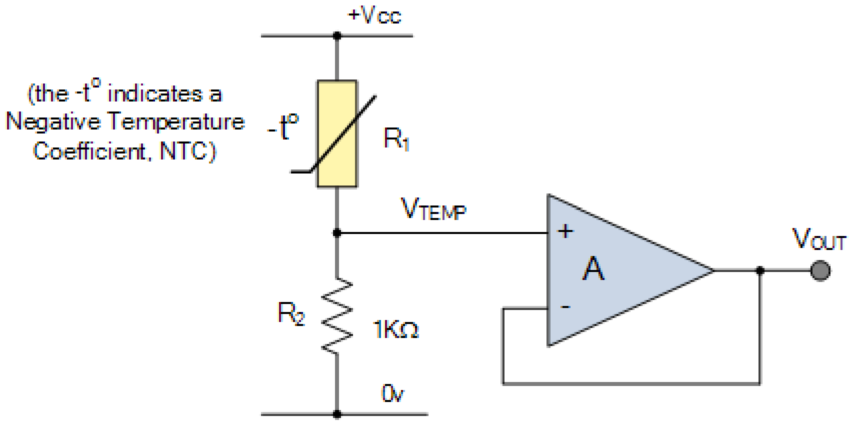


Figure 14: Negative Temperature Coefficient design

Source: http://www.electronics-tutorials.ws/io/io_3.html

There exists a sensor that is even more accurate than the NTC thermistors just mentioned, and it is called a resistance temperature detector (RTD). By mapping the resistance of the RTD to changes in temperature, we can get extremely accurate readings across an even wider range of operating temperatures. An RTD would give us great accuracy, but it’s operating range could be considered by many to be too large for this project at -200°C to 600°C. RTDs also happen to be some of the most expensive temperature sensors available, so we easily passed on this technology.

A third way to create a temperature sensor is to measure the difference in voltage of two pieces of wire that are made out of different materials and connected at two points. Because it is known how different metals react to different temperatures and these wires are having their voltages measured at the same points, you can use a lookup table to compare the varying voltages and convert that into a temperature. The accuracy of this method is not very high, temperatures vary from 0.5°C to 5°C, but these types of sensors are viable over a relatively large temperature range. For our project, we want to avoid using a thermocouple sensor, our operation would take advantage of less than 5% of this sensor’s typical operating range and its inaccuracies are too large to ignore.

The fourth and most promising temperature sensing technology we saw was a semiconductor-based sensor. These sensors have a relatively small range compared to the other three technologies mentioned, -40°C to 120°C, but this range is actually closest to where our device will be operating and it will still have room to spare. Semiconductor-based sensors make use of integrated circuits to measure temperature. The sensors are inherently small, and are actually touted by many as being useful for microcontrollers, embedded devices, and other electronic devices. These sensors provide a linear response to temperature, they do not have to be converted like some other technologies, but are not very fast or entirely accurate. However, for our purposes this type of sensor will do just fine to measure ambient temperatures in the air.

Sensor Type	Advantages	Drawbacks
NTC	1.Change in resistance mapped to change in temperature 2.Good accuracy	1.Self-heating increases with temperature increases 2.Sometimes requires calibration
RTD	1.Best accuracy we've seen	1.Operating range is easily too large for scope of this project 2.Expensive
Thermocouple	1.Good accuracy 2.Can find low power sensors	1.Like RTDs, the operating range is much too large for us
Semiconductor	1.Perfect operating range 2.Low power consumption 3.Low cost 4.No calibration 5.Linear response	1.Not as accurate or as fast as other types, but good enough for this project

Table 5: Temperature sensor technology comparison

After reviewing the advantages and disadvantages of the various sensor types, we decided to choose a semiconductor based sensor. These make use of an integrated circuit to measure temperature, and the sensor we chose performs an analog to digital conversion for us and provides an easy way to communicate with our microcontroller.

Barometric Sensor

A barometric pressure sensor has all sorts of applications and can be found in many things from cars to phones. Mainly, the pressure sensor is used to determine the pressure on the atmosphere around it. Atmospheric pressure is how heavy the air is at ground level. Intuitively, this means that because there is more air above a point at sea level compared to the same point at any higher elevation that the barometric pressure at that point is greater at sea level. Barometric pressure is useful for our project because it allows us to predict how likely it is for there to be precipitation in the near future. This works because areas of high pressure push air downwards, forcing it to descend. This air warms up as it descends in altitude and this makes it unlikely for clouds to form, making storms even less likely. On the other hand, when a pocket of air flows through an area of low pressure, that air rises. While it rises it cools off, this causes it to condense and form clouds, and possibly precipitation. This is why it is useful to be able to measure and keep track of changes in air pressure over time.

There are a few different types of pressure sensing technology. Sensors that measure an applied force over a known area are called force collector types. This is the most common type of pressure sensor, and there are a variety of ways that it can use to make this measurement, two of them being capacitive technology and piezoelectric technology.

Capacitive barometric pressure sensors work by making use of a small conductive surface, called a diaphragm, that is able to flex in a certain direction inside of a pressure cavity. This flexing results in a change of capacitance of the overall device, which can be amplified and measured as it changes. The capacitive output is measured linearly and many sensors use common materials for their diaphragms so it is generally a lower cost compared to other technologies.

Piezoelectric sensors use the electric charge that accumulates in crystals and other materials due to stress. The strain induced on quartz, for example, generates electricity that can be measured and directly related to the pressure on the sensor. Piezoelectric sensors aren't just used pressure sensing though, the effect can be utilized to generate and measure sound waves and voltages and is even used in watches to measure time.

Sensor Type	Advantages	Drawbacks
Capacitive	1.Can be made from simple materials	1.Smaller operating range is okay for our project, but makes other technologies a better fit for other applications
Piezoelectric	1.Measurements are not largely affected by radiation or electromagnetic fields 2.Versatile, can be used for different kinds of measurements	1.Relatively expensive compared to capacitive 2.Higher temperatures cause drops in sensitivity

Table 6: Barometric sensor technology comparison

In the end, we decided to go with a capacitive based pressure sensor. Since they can be made from simple and easy to access materials, they are generally cheaper than many of their counterparts while still remaining an accurate way to measure pressure.

Water Level Sensor

Our portable watering system must have a way of sensing how much water is in the tank so that it can inform the user when to add water. This is an important addition because it makes day to day operations less stressful, makes the product easier to use, and reduces the chance of plants not being watered. Through our research we uncovered a few different types of water level sensors, the ones we most thoroughly considered were ultrasonic sensors, floats, and laser sensors.

Laser sensors work by shooting out a thin beam of light into the substance in a container, and measuring the time it takes for that light to be reflected back. Since the beam is coming from a laser there is not much spread, meaning these sensors can be used in small spaces and measure long distances while still giving accurate readings. For us, accuracy in a small space is desirable, but unfortunately, laser sensors work best when measuring solids or opaque liquids. It seems like they wouldn't work well inside of a water tank.

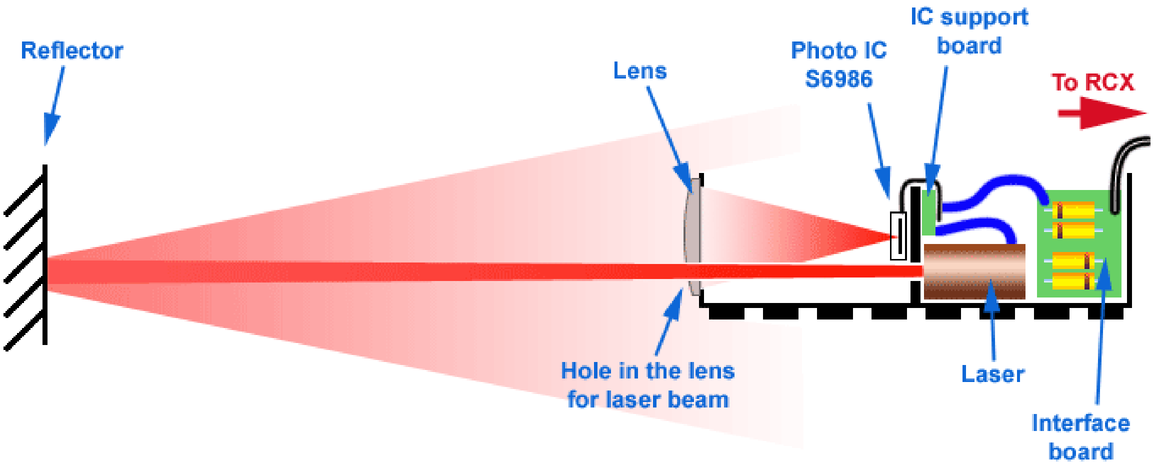


Figure 15: Diagram of basic laser sensor

Source: http://www.philohome.com/sensors/lasersensor.htm

Floats are a relatively low-tech solution to the problem of sensing depth. Floats are buoyant objects that would float on top of the liquid in our tank. Because they will always float on the liquid, if we could measure their position somehow then we would know the amount of water in our tank. There are many ways to go about achieving this, we thought of some simple solutions that made use twine, for example. We were unable to figure out how we could take this physical measurement and send it to the PCB so that we could automate the level sensing. After doing some further research we found that magnetic floats are common today and transmit using networks of switches and resistors, so this helped solve our earlier problem, but we decided against having to set up a float system for our project.

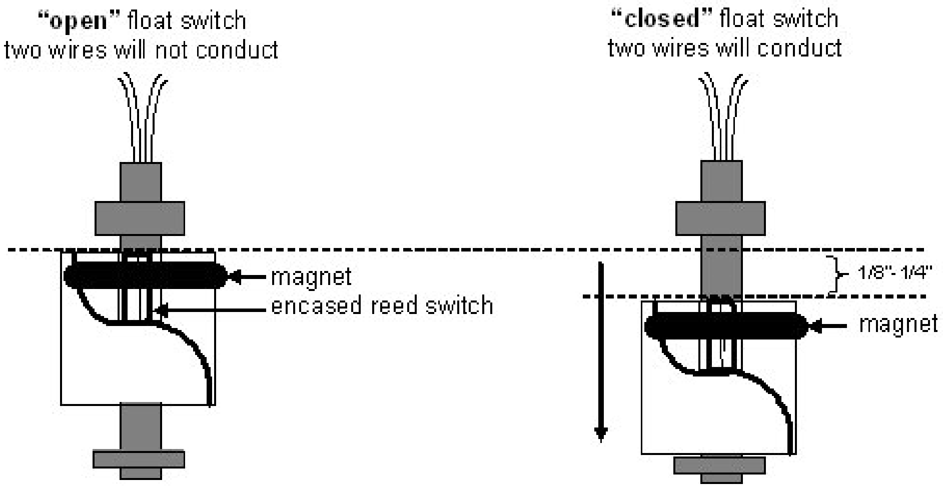


Figure 16: Float sensor diagram

Source: http://www.aquahub.com/store/media/floatwork-2.jpg

Another type of water level sensing technology is a strip sensor. These sensors usually provide continuous level sensing through means of variable resistance. As the level of the measured fluid changes, so does the resistance of the strip sensor. In our case, the strip sensors we looked at had a resistive output that was inversely proportional to the level of measured fluid, so higher liquid levels mean lower resistance output.

This leaves us with ultrasonic sensors. These work similarly to the laser sensors mentioned earlier, but they transmit sound waves instead of light. Because these sensors use sound, it takes a little bit longer for the wave to travel back to the source, increasing the time it takes to measure the distance, and thus, the level of water in the tank. For our purposes, the difference in speed between light and sound is negligible. The size of our tank is relatively small, and we don't need extreme speed for our measurements because the tank cannot be depleted so quickly. We can also afford to sacrifice some accuracy compared to laser sensors because we will be alerting users that they need to add water long before it actually runs out.

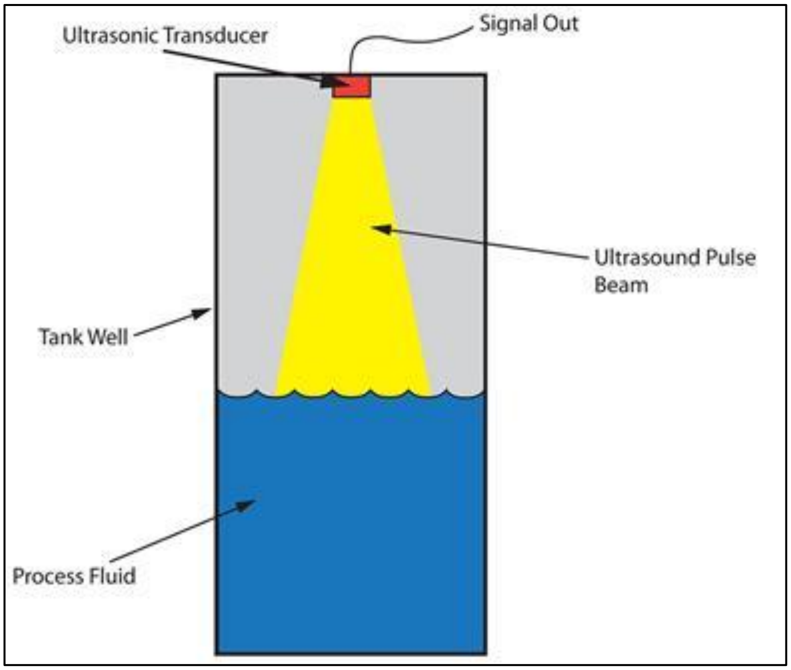


Figure 17: Ultrasonic sensor

Source: http://www.azom.com/article.aspx?ArticleID=12457

Ultrasonic sensors also have another minor drawback: there exists a problem where the sensor won’t return any data if the object it is sensing is too close. This is because the time it takes the pulse to leave the sensor and come back into it is too small for the sensor to make a measurement on since it takes time to switch from transmitting mode to receiving mode. We can work around this by making sure the sensor is placed far enough away from the water we are measuring in the tank, that way the sensor always has enough time to send and receive a pulse. We will also add a recommended fill line in the tank and let users know that filling above this line can cause the sensor to produce inaccurate readings or even none at all.

Sensor Type	Advantages	Drawbacks
Ultrasonic	1.Cheap 2.Low current draw	1.Accuracy 2.No measurement returned if water level is too close to sensor
Laser	1.Small beam spread 2.High accuracy	1.Work best measuring solids or opaque liquids

Table 7: Water-level sensor technology comparison #1

Sensor Type	Advantages	Drawbacks
Float	1.Low tech solution 2.Low cost	1.May be cumbersome for user to set up and makes our project less portable
Strip	1.Easy to read level using microcontroller 2.Low power 3.Adequate temperature range	1.Cannot be bent without affecting measurements 2.Pins cannot be soldered directly, requires a connector

Table 8: Water-level sensor technology comparison #2

Based on factors such as cost, size, and ease of implementation, we were leaning more towards a small ultrasonic sensor, however, we decided to try out a strip sensor instead. The strip sensor can communicate easier with the microcontroller because its resistance value directly correlates with the level of the water we are sensing. We also decided to try out the strip sensor because it is more accurate and doesn't have any dead zones like an ultrasonic sensor does. We ordered some ultrasonic sensors as backup just in case something goes wrong with our strip sensor, we were okay with this because of their cheap cost and their reduced accuracy is still tolerable for this type of project.

3.2.4 Wireless Module

When devices need to be connected without the use of wires, a wireless network is needed. Wi-Fi and Bluetooth are two of the most popular wireless networking technologies. They each have their pros and cons and are used based on the requirements of the user and the product. We also came across a third wireless technology called ZigBee that seems like it could compete with Wi-Fi and Bluetooth for a spot on our PCB.

Wi-Fi uses radio frequency (RF) technology to provide wireless high-speed internet and network connections. When an RF current is supplied to an antenna, an electromagnetic field is created which is then able to propagate through space. This is how computers are able to communicate with a network inside a home or office building. There are many different Wi-Fi standards ranging from IEEE 802.11a all the way to IEEE 802.11ac. These standards have progressed over time, with a new one being released every time a major new feature has been added to the Wi-Fi protocol. The main difference between the various Wi-Fi standards is the frequency in which they operate (either 2.4 GHz or 5.0 GHz) and their respective signal range. The higher the frequency, the faster the speeds they can provide but the smaller the signal range. [6]

For a while, 802.11b was the go to standard for Wi-Fi technology because of its perfect implementation for household and office scenarios, the two most popular places where Wi-Fi is commonly found. In the past few years, 802.11n has started to pick up in popularity thanks to its combination of both frequencies. With an 802.11n Wi-Fi router, users are able to enjoy the increased speeds of a 5.0 GHz network when they are in close proximity to the router and then transfer over to the 2.4 GHz network when the 5.0 GHz network starts losing signal strength. This way users are able to get both ends of the spectrum with one product.

On the other hand, Bluetooth is a wireless technology standard for exchanging data over short distances from fixed and mobile devices, and building personal area networks (PANs). Bluetooth uses short-wavelength UHF radio waves in the ISM band from 2.4 GHz to 2.485 GHz. The main advantage Bluetooth has is its low-power wireless connectivity technology which is mostly used to stream audio, transfer data and broadcast information between devices. There are two different types of Bluetooth technology, Basic Rate/Enhanced Data Rate (BR/EDR) and Low Energy (LE). Basic Rate/Enhanced Data Rate enables continuous wireless connections and uses point-to-point network topology to establish one-to-one device communication. On the other hand, Low Energy enables short-burst wireless connections and uses multiple network topologies, including point-to-point, broadcast and mesh. [7]

ZigBee has been on the scene since 2003 when it was first standardized, although the concept was born back in 1998. Like Bluetooth, ZigBee also makes use of Wireless Personal Area Networks (WPANs). It can be considered a low cost, low power alternative to the likes of Wi-Fi and Bluetooth. It has a relatively slow transmission speed, only about 250 kbit/s, but this is fine for the likes of sensor data transport. Because of its low power consumption its range is limited from 10 to 100 meters, which seems far, but line of sight is needed for communication. It is possible for ZigBee devices to communicate over larger distances through the use of a mesh network of intermediate devices. Ideally, ZigBee isn’t meant to compete with Wi-Fi in terms of speed, but since it low power we are seriously considering trying out this technology for our project. As an added bonus, ZigBee networks are automatically secured and communication takes place on an encrypted network.

ZigBee was developed with battery life of components in mind, so it draws very little current and its low latency communication keeps its current draw low. ZigBee operates in the ISM (industrial, scientific, medical) band of radio waves. In most countries this is the 2.4 GHz band and it offers quicker speeds. In the United States, however, ZigBee networks operate in the 915MHz band with slower data rates, a little more than 20 kbit/s compared to 250 kbit/s. [8]

Technology	Advantage	Disadvantage
Wi-Fi	1.Provides high-speed access to the internet 2.Long signal range 3.Once device is connected to the internet, could be operated from anywhere in the world	1.Harder to setup 2.Walls and other obstacles interfere with signal strength
Bluetooth	1.Easy to connect devices 2.Easier to setup in microcontroller environment	1.Only works over short distances (less than 30 ft) 2.Cannot connect a device to the internet
ZigBee	1.Low cost 2.Low power 3.Encrypted network	1.Mainly works over short distances. 2.Longer distances are possible but you won’t be able to access your device anywhere as if it was connected to the internet

Table 9: Wireless technology comparison

For this project, we have decided to go with a Wi-Fi Module for our device. Although Bluetooth would have been a lot more convenient and easier to program, the range issue would have greatly impacted our device. Since our end goal is to have a portable, automatic sprinkler system, it would be nice for the user to be anywhere in the world and still be able to connect to the device and see its status and make any changes if necessary. This also eliminates ZigBee since, even over its mesh network, you cannot truly access it anywhere in the world. This only leaves us with Wi-Fi as an option. Although it is a little more difficult to work with, once setup properly, its advantages are evident as to why it is the better option over Bluetooth for our device.

3.2.5 Voltage Regulator

When constant voltage is needed across a load with a higher voltage given than wanted a voltage regulator can be implemented to help step down the voltage to a more manageable range. A voltage regulator is able to accept a DC voltage on its input side and reduce it down to a fixed voltage or adjustable depending on how you design your circuit. There are two types of typical voltage regulators: linear and switching.

This component is necessary for reducing our voltage from our battery down to have our microprocessor and other sensors work properly. A voltage regulator reduces the voltage safely to help protect our components and can also have a built-in feed-back system so that if a current is being drawn where it shouldn't the system can protect itself. [9]

The basic circuits of regulators typically contain an Op-Amp, BJT transistor, a Zener Diode, resistors and some capacitors. The op-amp helps drive the BJT by giving it more current which helps to have a slightly higher output voltage. While the transistor itself in common collector mode can have a fairly stable output voltage and function as a regulator on its own

The resistors labeled 1, 2 and 3 in the figure below can be used to adjust the output voltage. By doing the voltage divider rule you're able to adjust the values of the resistors to gain a favorable output.

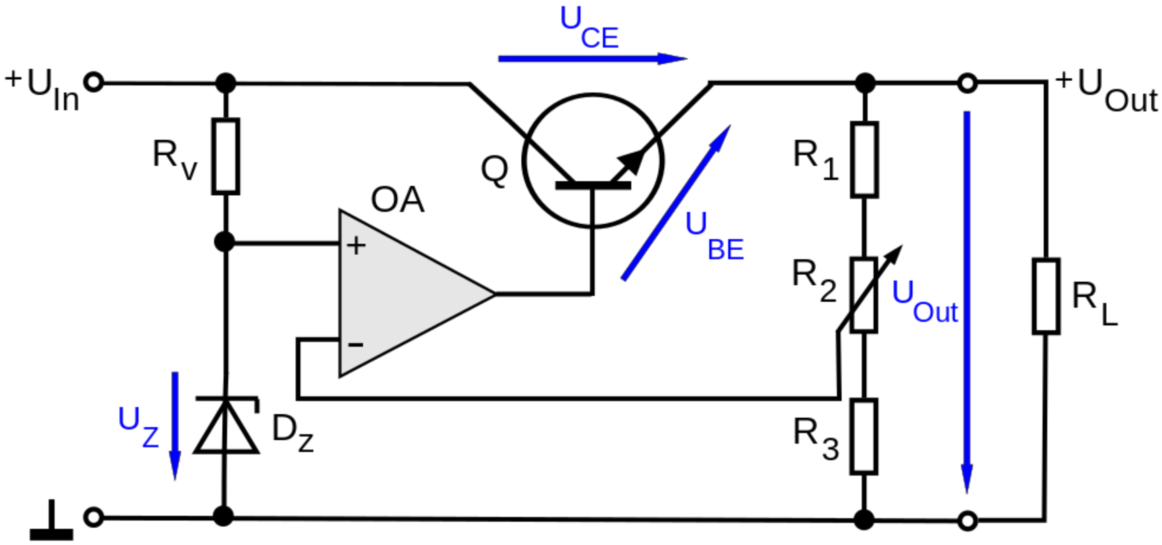


Figure 18: Voltage regulator using an Op-Amp and BJT model

Linear Voltage Regulator

This type of voltage regulator works by the voltage divider rule, by adjusting the resistance of the regulator based on feedback voltage. This allows the regulator to handle any amount of current placed onto it up to its limit and still output at a constant voltage.

The main problem with linear regulators is there power dissipation. Since the regulator needs such a relatively large amount of input voltage to reach its output goal its voltage drop can be large as well. This means that the greater the voltage drop, the greater the power dissipation. The efficiency of these types of regulators therefore suffer greatly. Having a small output voltage over the large input causes its efficiency to drop.

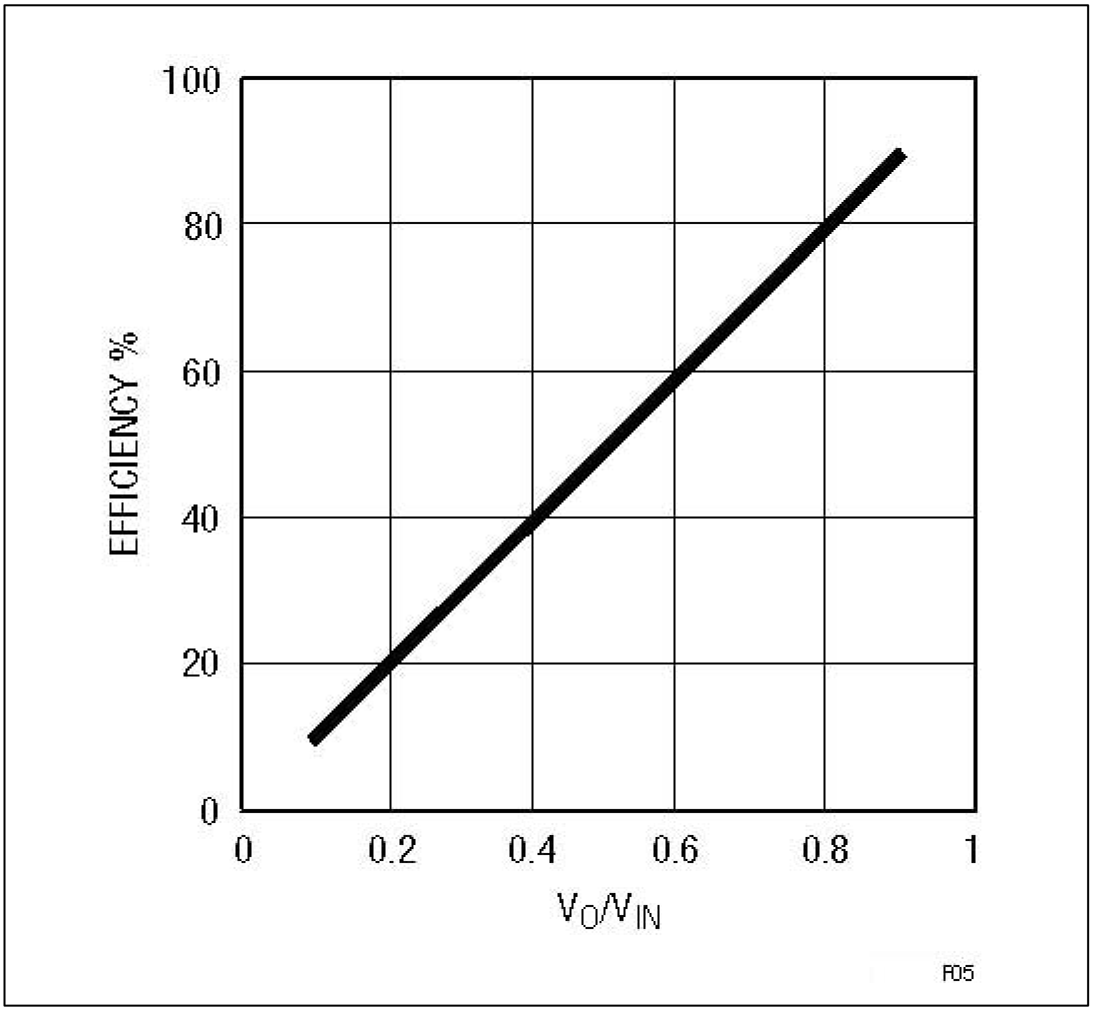


Figure 19: Linear Voltage Regulator

Source: http://www.edn.com/Home/PrintView?contentItemId=4420313

Switching Regulator

Where linear voltage regulators were not great with efficiency switching regulators are. By utilizing more components they're able to raise their efficiency to almost 80% better than linear regulators. The downside to this is that with more components comes a greater design challenge and reliability on all components working properly.

The Buck converter is a step-down voltage regulator that drops a DC voltage to a lower and same polarity DC voltage. This converter uses an inductor to drive a current into a load resistor to get the proper output voltage and hold it at the desired amount. A capacitor is used to be charged and help keep the inductor current constant across the load resistor to keep the voltage stable.

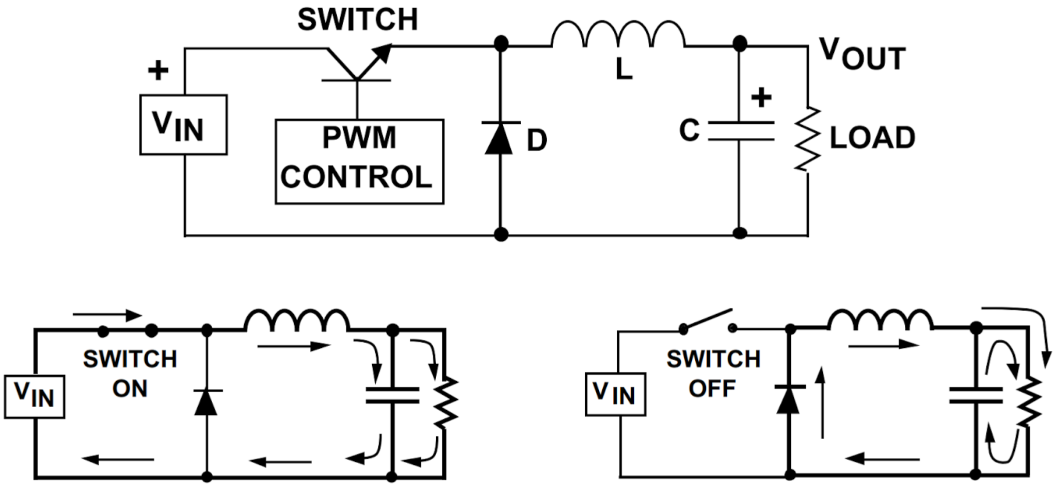


Figure 20: Buck converter circuit using PWM and BJT

Source: http://www.ti.com/lit/an/snva559a/snva559a.pdf

With our project being a low-power driven module, it makes more sense to go for the most efficient of the two regulators. That would be the switching regulator. While it may take up more space on the PCB designing a switching regulator will definitely help our understanding of the circuit as well as keep our power requirements met. With a more efficient regulator we can help make our battery supply last longer and power our other components more effectively. [10]

3.2.6 Voltage Booster vs Relay Circuit

A voltage booster, or boost converter, is a circuit or components that is able to take low input voltages and step the voltage up to a much larger number sometimes double or triple the input. This circuit is utilized in many applications especially those in RF technology to help amplify their signals. In our utilization though, it is used to help power a 12V water pump.

Basic boost converts consist of a MOSFET transistor being used as a switch to help store current in an inductor and then switches off to send that current to a load and effectively boost the voltage. The output voltage is reliant on the duty cycle of the switching MOSFET. The faster a duty cycle the higher the output voltage because of the almost constant current given to the load. It's easier to think about a booster converter as a reverse buck converter. Instead of needing our output voltage less than our input, we need our input less than our output. [11]

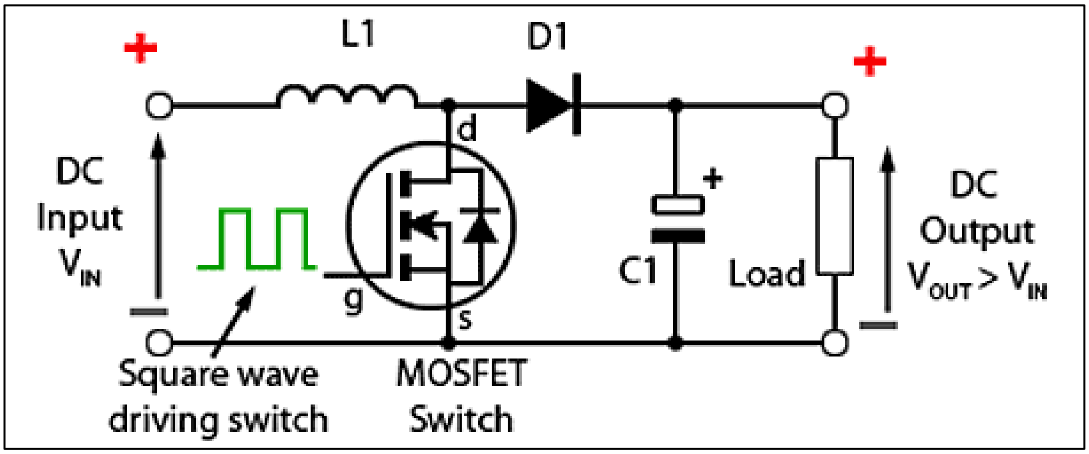


Figure 21: Basic switching boost converter

Source: http://www.learnabout-electronics.org/PSU/psu32.php

A relay circuit uses an input to saturate an npn BJT transistor and drive current through the zener diode to switch the relay. Doing this turns the voltage at the top to be driven into the load attached to the relay. In our case our input would come for a general purpose input/output pin from the microcontroller. The GPIO is able to deliver 3.3 volts which is enough to saturate the BJT. The load in this case would be the water pump that can operate from 5 volts delivered from our battery source. This circuit works better for loads that will be turned on for a short amount of time. [12]

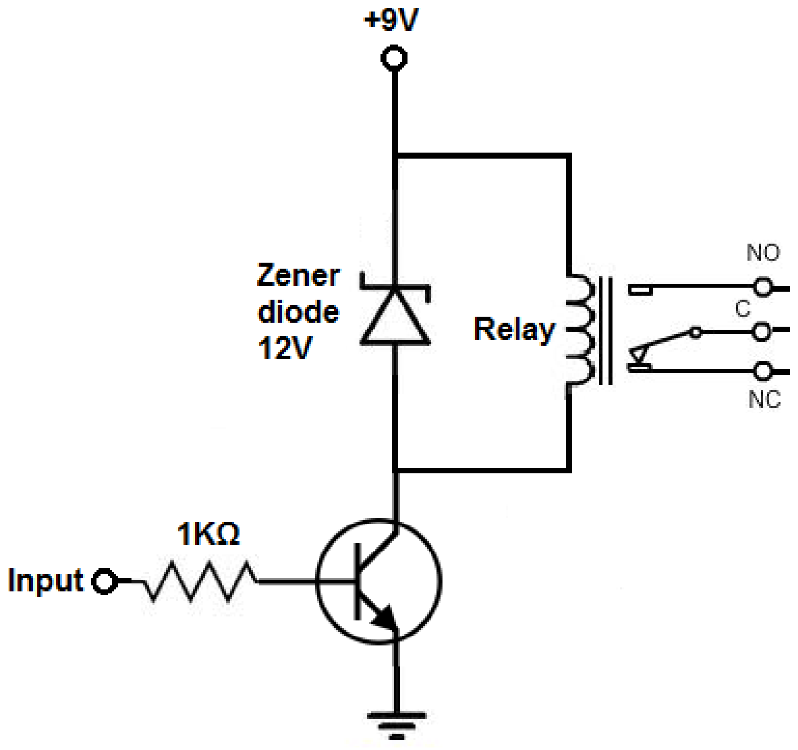


Figure 22: Basic relay circuit

The boost converter would be a good selection if we were to run the pump for long durations, but since the watering function of the device won't take too long another route may be possible. A simple relay would allow the water pump to be activated by 5 volts (component willing) and not use the boost converter and potentially waste power. This simple circuit is switched using a GPIO port and driven with the battery output.

Voltage Booster	Relay Circuit
Conserves power on long usage	Uses power on short usage
Harder to build	Easier to build
More space on PCB	Less space on PCB
Requires PWM	Requires GPIO input voltage
One driving voltage	One driving voltage and one input voltage

Table 10: Voltage Booster vs Relay Circuit comparison

3.2.7 Battery

There are many types of batteries available in the market today, however the one that we are using for this project is a portable battery pack or a power bank. This is because power banks are small size batteries that allow us to charge small devices like phones, or tablets. A power bank is a very popular device today as it provides instant charging with portability, consumers mainly use it for charging their cell phones. The power banks consist of a lithium ion or a lithium polymer battery, the purpose of a power bank is to store charge in it by having it initially charged, and once charge is stored, it can be used to power other devices.

How does a lithium-ion battery work? A battery has two ends called electrodes, one is called cathode (- charge) and the other is anode (+ charge). The role of a battery is to store and release energy by moving electrons from anode to cathode. The moving of electrons is current. We can use this energy by connecting the battery to another device and deliver power to it. Here, cathode is made from a metal oxide like cobalt oxide, and anode is made from the element carbon. Between the electrodes, there are electrolytes in the battery. Electrolytes contain lithium ions. When a battery is placed to operate a device, the positively charged lithium ions from the anode move to the cathode. Once the cathode is full of these ions, electrons are attracted due to cathode becoming more positively charged than the anode. We can let our device use the energy from the battery because we force the electrons to move to the device first and then to the cathode, as shown in the figure below.

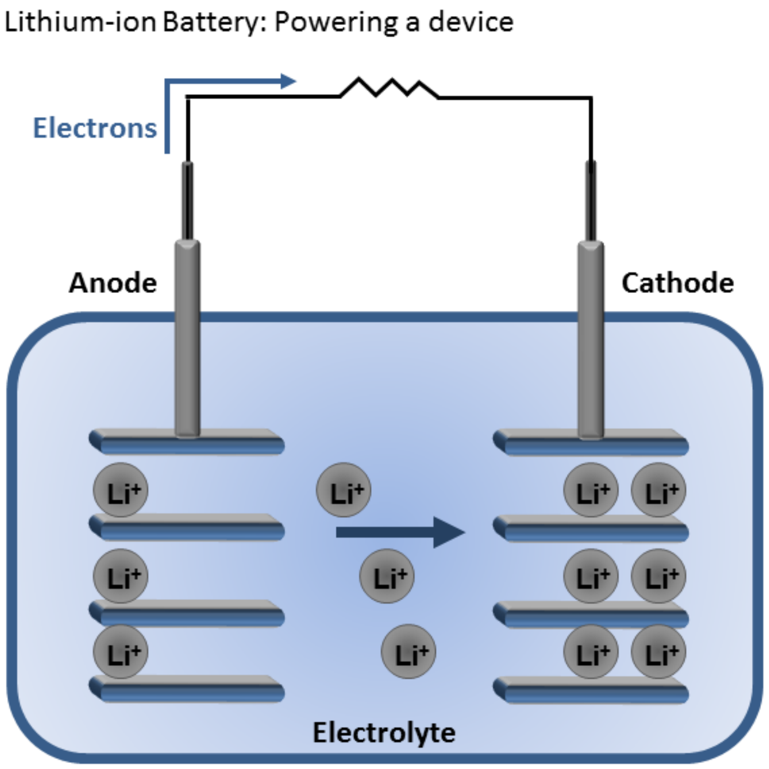


Figure 23: Activity in Lithium-Ion battery while powering device

Source: http://sustainable-nano.com/2013/10/15/how-do-lithium-ion-batteries-work/

The reason why the lithium-ion batteries are popular because these are rechargeable batteries and they're used in many devices like phones, laptops, iPod, etc. The lithium ion produces a lot more electrical power/unit of weight compared to the other batteries, so lithium ion batteries store the same amount of charge as other batteries but in a smaller size. When we connect a charger to the battery, the lithium ions now move to the opposite directions; from cathode to anode. This causes the battery to recharge. One disadvantages of lithium-ion batteries is that they require special protection to control the voltage and the current flowing through the battery so that it doesn't overcharge and discharge. Another disadvantage is that they age quickly, depending on the cycles completed by the battery and how the consumer uses it. After is ages, the battery loses charge quickly. Lastly, they are very costly when compared to other batteries.

Another type of battery used today in handheld devices is the lithium polymer, which uses a polymer electrolyte instead of a liquid electrolyte. This battery is also rechargeable, has an ultra-thin design that is usually compared with that of a credit card and is very lightweight since it does not require a robust body to protect it. Lithium polymer batteries are more resistant to overcharging, which lowers the chances of electrolyte discharge, and as bonus, they can be made in several sizes since manufacturers are not forced to make them in standard cell formats. The only disadvantages to these batteries are that the manufacturing cost is very high and they have a lower energy density than lithium-ion, which translates to a smaller charge storage.

Nickel cadmium is another type of battery which uses nickel oxide hydroxide and metallic cadmium as electrodes. They can be used just like we use AA or AAA alkaline batteries but they have their own properties. These batteries are rechargeable, provide fast charging, have a high number of charging cycles and have a charging temperature lower than most of the other batteries. They also don't get damages very easily when overcharged and are the most affordable batteries based on their cost per cycle. While these batteries have a lot of advantages over the others, they also have quite a few disadvantages. The first disadvantage is that they have a low energy density (small charge storage) and are made of toxic materials, which are harmful for the environment and are limiting the use of these batteries in some countries. Another big set of disadvantages for these batteries is that they have a very quick discharge rate and must be repeatedly used in order to prevent the memory effect where the battery loses charge quicker than it should. Another form of the nickel cadmium battery is the nickel-metal hydride. This battery is developed for advanced hydrogen energy storage concepts and used in high-end portable electronics where performance parameters such as runtime is important. They can have up to 40% higher capacity than nickel cadmium batteries and are less prone to the memory effect. They are also made out of recyclable material, making them environmental friendly. On the other hand, these batteries generate a noticeable amount of heat and take a longer time when charging, they discharge very quickly when not in use and their performance deteriorates if operated in high temperatures. They also require higher maintenance.

The next battery technology we are going to discuss is lead acid batteries. These batteries are usually larger in size than the previous ones we've mentioned and are mostly used on automobiles. They are rechargeable, simple to manufacture and inexpensive in terms of cost per Watt. They have a low self-discharge, are reliable and provide durability. They also have very low maintenance costs and have no memory effect. The main disadvantage of this battery type is the fact that it is not applicable to handheld devices. It also has a low energy density, only a limited amount of full discharge cycles is allowed and improper charging can increase the temperature and damage the battery.

Lastly, we have alkaline batteries. These batteries are extremely popular, with 80% of the batteries manufactured in the US being alkaline batteries. They come in many different sizes and can be found in all kinds of consumer electronics and household items. The main reason for their success is their list of features. Alkaline batteries, like all the other we have discussed, are rechargeable. However, they have four times the capacity of an equivalent nickel cadmium or nickel-metal hydride battery, can function at a very low temperature, are safe for the environment and their charge can last up to 7 years without being used. The only disadvantages to this kind of battery are that it has a very high internal resistance, which leads to low power output, and when they are kept in the devices for too long, they can leak the materials they are made of causing damage to the devices' circuits. They are also heavier, bulkier and have lower energy density compared to the lithium-ion batteries.

After carefully analyzing each type of battery technology, we concluded that the Lithium-Ion battery is the most promising battery for this project. It has the highest charge storing capacity, it provides high current output, and that it has a very slow discharge rate so that it doesn't runout of charge quickly when it is not being used by the sprinkler system to get power. Below is a quick comparison of all the battery technologies discussed in a table form.

Battery Type	Advantages	Disadvantages
Lithium-Ion	1.Rechargeable 2.High energy density 3.Slower self-discharge 4.No priming required 5.Provide a high current density	1.Require special protection 2.Age quickly 3.Costly
Lithium Polymer	1.Rechargeable 2.Ultra-thin design 3.Light weight 4.Manufacturers could make it in several sizes 5.Safe	1.Lower energy density 2.Manufacturing cost is high.
Nickel Cadmium	1.Rechargeable 2.Fast charging 3.High number of charging cycles 4.Charges at lower temperatures. 5.Robust battery 6.Affordable	1.Low energy density 2.Made of toxic material 3.Quick discharge 4.Must be repeatedly used to prevent memory effect

Nickel-Metal Hydride	1.Rechargeable 2.Higher capacity compared to Nickel Cadmium battery 3.Less prone to memory effect 4.Made of recyclable materials	1.Generates heat during charging and requires more time to charge than Nickel Cadmium battery 2.Discharges very quickly 3.Performance deteriorates 4.Requires high maintenance
Lead Acid	1.Rechargeable 2.Inexpensive 3.Reliable and durable 4.Low self-discharge 5.Low maintenance	1.Not applicable for handheld devices 2.Limited amount of full discharge cycles 3.Environmental damage 4.Improper charging
Alkaline	1.Rechargeable 2.Has four times the capacity of Nickel batteries. 3.Very slow discharge 4.Can function at very low temperatures. 5.Safe and environment friendly.	1.Heavier, bulkier, and has lower energy density compared to the lithium-ion batteries 2.High internal resistance 3.Leak the materials causing damage to the circuits

Table 11: Battery technology comparison

Battery Specifications

Many power banks normally have a microUSB port in order to charge its battery when connected to a wall outlet or another source of power such as a laptop. The power bank being used for the sprinkler system needs to have a DC input port as it will be powered via a solar panel, and the solar panel would have a DC output. The power banks also have USB port(s) that allows other devices to be plugged in to receive power. Another feature that the power banks generally have is a LED battery indicator, to show the user approximately how much charge is left in the battery.

So there are many types of power banks available, how do we know which one is the right one for us? Like any other gadget, power banks also have specifications. Battery capacity and current output are very important specifications to be considered. Battery capacity basically means the size of the battery, this is measured in units of mAh (milliAmpere/hour). For example, if a consumer’s phone battery capacity is 3,000mAh, and they use a fully charged power bank that is 4,500mAh, they can charge one phone at a time and completely for a total of 1.5 times, after which the power bank will run out of charge.

Another spec that can be found important to some consumers is the current output. Some power banks have a high current output of about 4A, whereas most of the power banks have the current output of 2.1A; the power bank with a 4A output will charge a device almost twice as much as faster than the power bank that has a current output of 2.1A. So, from a consumer point of view, if you’re a heavy user of your cell phones, it will be beneficial to buy a power bank with a large battery capacity so you can charge your phone often on the go. However, the disadvantage of having a power bank of large battery capacity is that the power bank will be large physically also, thus making it less portable.

3.2.8 Printed Circuit Board

In order to have our electronics and sensors working together across the board a PCB, or printed circuit board, is needed to cleanly and efficiently have everything connected. A PCB is a layered combination of a silkscreen, solder mask, copper and a substrate (FR4). The combination and layering of these together allow for a circuit to be wire together without physically needing to use wires to connect them together like on a breadboard.

Designing a PCB

The first step in designing a PCB comes in the form of selecting the required parts for the project to function properly. The best solution to making sure each part works is to order all the parts and function properly before designing our board on software.

The next step in design is sketching our layout on software. The software should be efficient and user friendly and have a wide library of components to be utilized in the modeling. Prior knowledge of circuit design will help in this process. Below is a table of potential software to help design and export our printed circuit. The cost as well as content in their libraries will be compared.

Software	Cost	Features
Eagle	$820 (student edition Free)	-Thousands of premade libraries -Widely used so a lot of support -Available premade templates
OrCAD	$1500	-Circuit Simulation -Auto-routing program -Creatable database to store component values
PADS	$5000	-In program walkthroughs -Easy FPGA design for digital circuits -Thermal analysis to see heat radiance from circuit
ExpressPCB	Free	-Simple design and order process from same company -Active online community -Open-source libraries

Table 12: Comparison of PCB Design Software

From the table above it is clear to select Eagle as the software for design. With the university using the software as well it adds another layer of support. The free version that is available to students includes all items necessary for designing the appropriate PCB for this project.

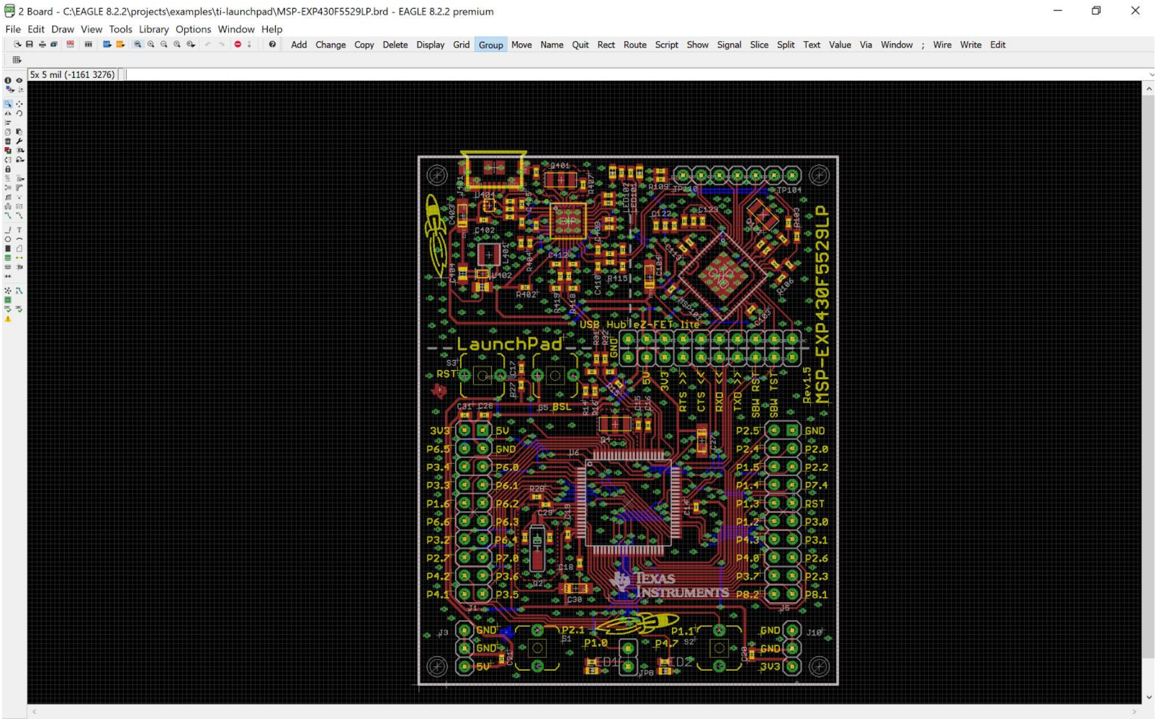

2 Board - C:\EAGLE 8.2.2\projects\examples\ti-launchpad\MSP-EXP430F5529LP.brd - EAGLE 8.2.2 premium
File Edit Draw View Tools Library Options Window Help
Add Change Copy Delete Display Grid Group Move Name Quit Rect Route Script Show Signal Slice Split Text Value Via Window ; Wire Write Edit
5x 5 mil (-1161 3276)
USB Hub|eZ-FET lite
LaunchPad
MSP-EXP430F5529LP
Rev1.5
TEXAS INSTRUMENTS

Figure 24: Eagle Software PCB board layout representation

With the Eagle software, the layout needs to be formed by sketching the size of the PCB on the modeling software. Within the sketched-out boundaries of the board the user will design the proper circuit design. The parts to the design for the circuit can typically be found in premade libraries from the program designer, otherwise open-source libraries with proper components can be downloaded.

After the proper design and simulation of the circuit it's time to export the file and send it to a manufacturing company. The company will take the design and print out the circuit properly by designing each layer and component of the PCB precisely. After the manufacture confirms the board is functioning and conducting properly the board is sent back to the consumer for final completion of the board by soldering on the final components of the circuit.

PCB Manufacturing

When it comes to the creation of our PCB we want it to be a quality build but also come at a cheap price in order to reduce costs. It's also important to find a cheap prototyping builder so that if we have issues with one design were able to get a cheap build setup and sent back out quickly. While most manufacturers must have the designs created in order to get a full price some places offer a set cost for a PCB. Designs that get to pricey would be perfect to go to this type of manufacturer. A mix of two manufacturers may be best also. One for prototyping and one for final build, although this runs the risk of a difference of quality between the companies messing with the design. The user rating is very important to us for the quality of build and customer service aspects as well.

	Price	Manufacturing speed	Layers	Rating
4PCB	$33	5 days	2	5 stars 5 ratings
PCBCart	quoted	1-5 days build 5 days - 2 weeks shipping	1-10	5 stars 5 ratings
OSHPark	$26	5 days	2	5 stars 5 ratings
PCBWay	$5 + shipping	2 - 5 days	2	5 stars 5 ratings
Seeed Studio	quoted	1 day build 5 days - 2 weeks shipping	1-16	4 stars 4 ratings

Table 13: Comparison of PCB manufacturers

While each company is able to satisfy the number of layers we need for our design the major downside to Seeed studio is its rating and the amount of PCB styles it can produce. The 4PCB deal is a nice deal for major designs but since our PCB won't be too expensive it doesn't make sense to use it. Therefore, we will use PCBCart to make both our prototyped and final design PCBs for the project.

3.2.9 Water Pump

Pumps today are what help drive modern man's average day. From refrigerators to cars, and from cooling computers to heating pools today's pumps are what help drive the world today. A pump in general is a machine that is able to move and deliver fluids from one area to the next. These fluids can range from a gas or a liquid.

In order to get our proper spray distance we need to have a high enough rate of flow to have a strong stream. By having a higher voltage and current being drawn across the pump were able to generate more power (Power= Voltage x Current). With a higher power the pump is able to drive a more significant stream of water.

Flow rate is the amount of volume of a fluid being moved at an amount of time, this is typically measured in either liters per hour or gallons per hour. With a higher movement of water at the output nozzle of a pump the more force can be delivered through a nozzle. This nozzle helps to control the spray distance and angle at which the device delivers. Nozzles can be designed and developed using a 3D printer and online templates to help deliver the best angle for our cone and distance as well the consistency of our droplets of water. [13]

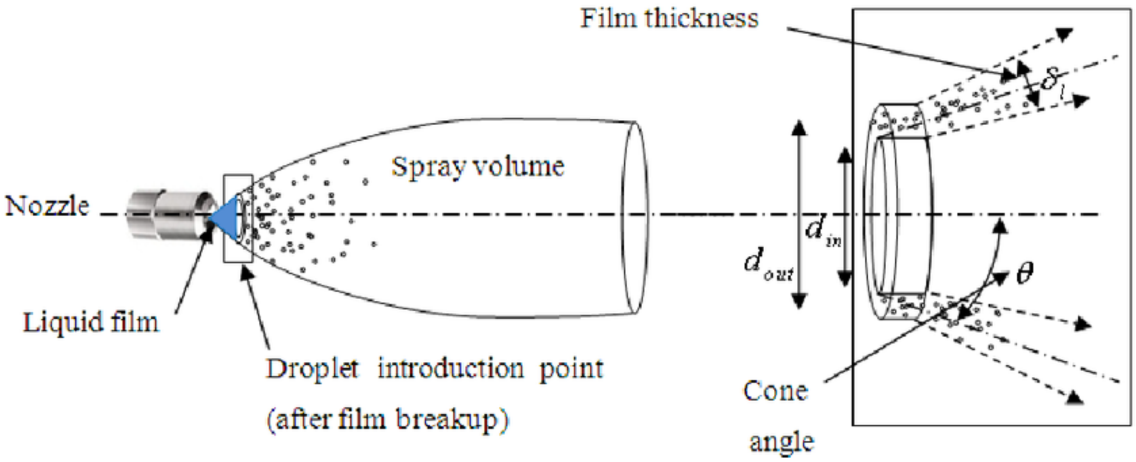


Figure 25: Demonstration of direction of spray and angle

Source: https://www.researchgate.net/figure/281436185_fig1_Fig-1-Schematic-diagram-of-the-syst...

For our project design, we will be looking at water specific pumps seeing as though our design will be watering plants. When we look at pumps in general there are typically two types: Reciprocating and Rotary pumps.

Reciprocating pumps can be as small as a bike pump to the major pumps located aboard large class ships. These pumps use a function of cylinders to push and pull a fluid from its resting location to the desired one. The design uses multiple valves in order to continually move the fluid forward through the system. A reciprocating design that has 2 stages. The first stage will be an intake stage, to draw the fluid in. The second stage is the discharge stage where the same piston that's driving the intake stage will reverse the flow and send the fluid that's been taken in out through a different opening in the system through the systems of valves stated before.

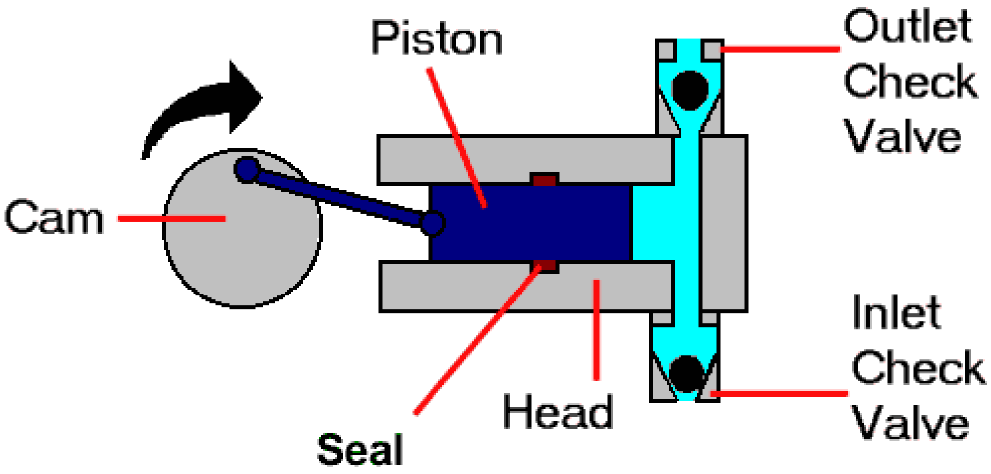


Figure 26: Reciprocating pump

Source: http://www.instructables.com/id/Simple-reciprocating-pump/

Rotary pumps or centrifugal pumps, use motors in order to trap fluids between each rotation of the motor driving the pump. By storing fluid in each cycle of the inside compartments of a motor the energy from centrifugal force helps to drive the fluid to the outlet of the motor. Typical designs can range from a small aquarium pump to larger pumps that help fill retention ponds or reduce flooding in areas. Some designs for rotary pumps include the Vane and Impeller pumps. Vane pumps use blades that slide back and forth in order to help drive the fluid. While impeller pumps have a circular propeller inside of the pump that spins around and collects the fluid in each compartment and flings it out the outlet side of the pump. The figure below represents a Vane pump that has two sides that move back and forth to help the fluid moving process. [14]

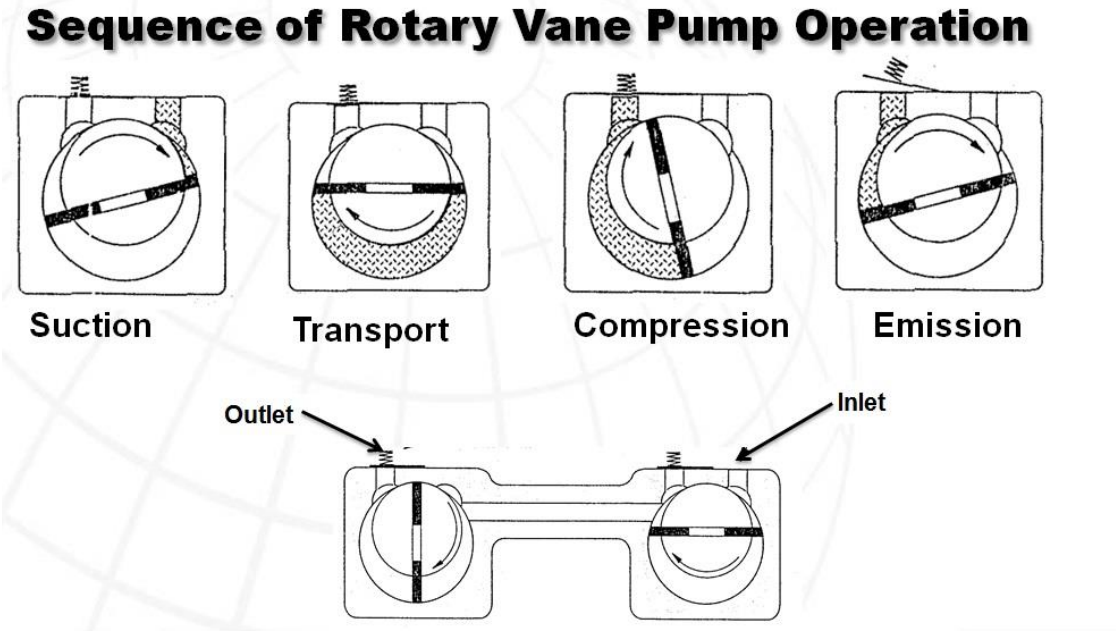


Figure 27: Basic Vane pump

Source: http://camblab.info/wp/wp-content/uploads/2015/01/Rotary-Vane-Pump-Operation.jpg

After researching the types of fluid pumps that there are and how each one is available in the market today, the team decided to go for a rotary based water pump. The price difference between the devices was a major deciding factor as well as the commercial uses. Reciprocating pumps are either used for large machinery or aren't efficient enough to satisfy the needs of our design.

Within the rotary section of water pumps the best and most readily available would be a brushless motor impeller pump. These are cheap, more efficient and able to be replaced easily if something should go wrong with the pump at any time. The amount of flow that each pump can produce would be sufficient enough but the impeller satisfies any needs we have.

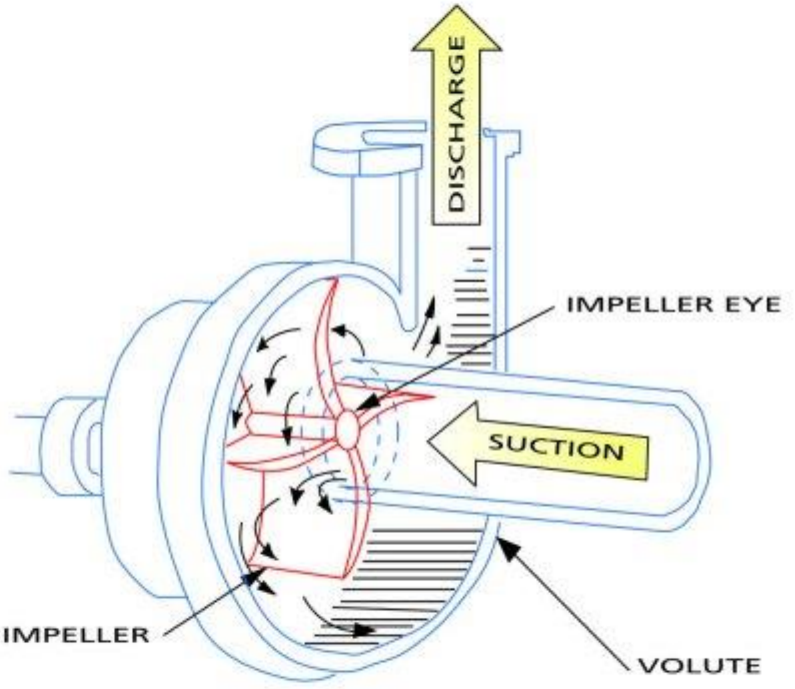


Figure 28: Impeller motor diagram

Source: http://www.pumpfundamentals.com/pump_glossary.htm

Reciprocating vs Impeller

While the reciprocating pumps are great at moving massive amounts of fluid they aren't very compact. Most small reciprocating pumps are almost double the cost of impeller pumps that would work. They are also significantly larger compared to impeller. The impeller motor is also relatively quiet compared to the reciprocating pumps. This being said we are using the impeller form of water pump.

Reciprocating	Impeller
Complex	Simple
Larger/take more space	Smaller/less space
Heavy in maintenance	easy to maintain
Costly repair	Cheap repair
Slow running	fast running
Pulse based output	Continuous output
More efficient	Less efficient

Table 14: Reciprocating vs impeller pump comparison

3.3 Strategic Component and Part Selection

This section focuses on specific components that we came across while doing research. Generally, we will describe a few different components for each type of technology and make comparisons between each of them. We will detail any advantages or disadvantages a certain part has, and end each subsection with what part we ultimately selected.

3.3.1 Microcontroller

AT32UC3L0128 (http://www.microchip.com/wwwproducts/en/AT32UC3L0128)

The AT32UC3L0128 is a 32-bit Atmel AVR microcontroller. This controller uses a RISC (reduced instruction set computer) architecture which has a smaller instruction set but can operate at a higher speed. It is a very low-power using module that is capable of reducing its current consumption to 165 microAmps per Megahertz and a leakage of only 9 nanoAmps. It supports I2C, SPI and UART serial communications to allow our pick for the best communication to the controller. The controller comes with support for up to 36 GPIO pins. With 128 KB of nonvolatile memory and 32 KB of RAM the controller is able to store all of the functions necessary for the project.

MSP430FR6989 (http://www.ti.com/product/MSP430FR6989)

The MSP430FR6989 is a 16-bit TI made microcontroller. Similar to the AT32UC3L0128 it uses a RISC architecture to operate at a higher speed compared to other controllers of its class. The low-power aspect of this controller has a drain of only 101.25 microAmps/MHz and a fast wake up time of 7 microseconds. This controller supports I2C, SPI and UART and supports 5 16-bit timers for constant refresh of interrupts for the sensors. It has 128 KB of nonvolatile memory and 2 KB of RAM which should be plenty of on board storage in order to not have a memory cache added. While it is a bit more on the expensive side at $9.64 it supports all the functions necessary for our device. The chip boasts a total of 83 GPIO pins for a large amount of I/O functions.

MSP430F6659 (http://www.ti.com/product/MSP430F6659)

TI also produces the MSP430F6659 which is a RISC based architecture controller. With 512 KB of nonvolatile memory and 66 KB of ram it has enough on chip storage to save the functions necessary. It supports all three serial communications of I2C, SPI and UART. The MSP430F6659 has 4 16-bit timers and has on-chip USB communication available. With 74 GPIO pins this controller can handle all of the I/O necessary. The downside is that this controller is pricey at $12.25.

ATmega328P (http://www.microchip.com/wwwproducts/en/ATmega328P)

The ATmega328P is the microcontroller commonly found on the arduino uno boards and is commonly bootloaded with the Arduino IDE (a common and easy to use IDE). The 8-bit AVR controller uses a RISC based architecture. With 23 GPIO pins and the ability to communicate via UART, SPI and I2C this chip can handle the necessities of the project.

This chip has 32 KB of Flash memory and 2 KB of SRAM(static random access memory) so while it wouldn't be able to hold an OS it would be able to store pre-programmed functions. Running at 20MHz would be usable for our project but wouldn't give a substantial amount of extra refresh time.

ATSAMG51 (http://www.microchipdirect.com/ProductDetails.aspx?Category=ATSAMG51)

The ATSAMG51 is the microcontroller seen on the new Raspberry Pi Zero boards. With a wakeup time of 3.2 microseconds, wasting power isn't a problem. At only 103 microAmps/MHz this chip is the most efficient one researched. Supporting 38 GPIO pins and having UART, SPI and I2C communication takes care of the necessary I/O and communication requirements. With 256 KB of memory and 64 KB of RAM this controller has plenty of memory but also sports 512 Bytes of Flash as well.

After researching each of the microcontrollers each showed a benefit whether it be the highest clock rate, the lowest power draw, most on chip memory or lowest power consumption. Below is a table comparing all of the controllers and some of the more important aspects of the important requirements.

	AT32UC3L0128	**MSP430FR6989**	**MSP430F6659**	**ATmega328P**	**ATSAMG51**
Clock Rate	50 MHz	16 MHz	20 MHz	20 MHz	48 MHz
Low-power Mode	165 uA/MHz	101.25 uA/MHz	295 uA/MHz	0.75 uA/MHz	103 uA/MHz
Cost	$6.39	$9.64	$12.25	$2.18	$3.14
Memory	128 KB	128 KB	512 KB	32 KB	256 KB
Pin Size	36	83	74	23	38
Communi cation Types	UART, SPI, I2C	UART, SPI, I2C	UART, SPI, I2C	UART, SPI, I2C	UART, SPI, I2C
Package Type	18 by 18 pin on board	18 by 18 pin on board	18 by 18 pin on board	DIP	18 by 18 pin on board

Table 15: Comparison of Microcontrollers

After doing initial research it was realized that memory would also have to be considered in this process to help reduce size and cost of the printed circuit board. While architectures were considered in our choice, the microcontrollers all had similar reduced instruction set computer architectures. While GPIO pins are also very important it was found that all of the controllers had enough pins to support all of the peripherals used by the project. The design of this project does not execute too many crucial instructions. It receives data from three sensors, transmits that data to an application, and drives a relay if necessary. Since the data is not too heavy, the controller doesn't need to be too extreme. That being said the ATmega328P-DIP package will be used to controller our design.

3.3.2 Solar Panel & Battery

Each type of solar panels discussed can be used depending on the project scale. After comparing the solar panels, we can say that it is be best to use a single-crystal/monocrystalline solar panel to power the sprinkler system. This is because we know that the water sprinkler system is a small scale system that only waters a few plants sitting in an apartment balcony. For a small scale system, we need a panel that is small in size and that can generate enough power efficiently otherwise if we have a polycrystalline panel, it can take up quite a bit of space in the balcony. The monocrystalline solar panel meets all of these requirements.

Based on this decision of using a monocrystalline solar panel for this system, research was conducted to find the most suitable and efficient monocrystalline panel. The comparisons of combinations of solar panels and batteries sold by Voltaic Systems are listed in the following table:

Panel Model	Panel Power (W)	Panel Peak Volt. (V)	Panel Size (in.)	Weight (lbs)	Cost
V15	6	6.0	8.7 x 6.9 x 0.2	0.77	$89
V44	9	6.0	8.7 x 10.1 x 0.2	1.35	$159
V72	17	17.5	15.5 x 10.75 x 0.25	2.90	$259
RNG-20D	20	12	13.5 x 18.5 x 1.0	4.8	$59.99
RNG-30D	30	12	23.8 x 13.5 x 1	6.2	$65.99
RNG-100D	100	12	47 x 21.3 x 1.4	16.5	$149.99
SOL-030P-01	30	12	27.3 x 17 x 2	6.6	$49.99
Venture 30	7	8-9	9 x 6.5 x 1.5	1.56	$169.99
Boulder 30	30	12	18.11 x 20.87 x 0.98	7.93	$137.13
SP-100W	100	17.5	45.5 X 1.5 X 26.5	20	$129.99

Table 16: Comparison of Solar Panels and Batteries

Specifically, Voltaic Systems claims that these monocrystalline solar panels have an efficiency of 19% and that they are waterproof. Also, each of these solar panels come with an output cable of 8.5 inches in length that connects to the battery.

To compare, Renogy (RNG-20D, RNG-30D, RNG-100D) is another reliable brand, however the sizes of their panels and their weights are larger than the panels of Voltaic Systems. Clearly, Voltaic Systems offers light and compact design which would suit the portable sprinkler system better than Renogy.

Another close brand is Goal Zero Venture 30, that is close to the size of Voltaic Systems however, their battery size is smaller and the weight is slightly more. And also, that Voltaic Systems panel with V44 battery is still slightly cheaper than Venture 30. The other panel SOL-030P-01 and Goal Zero Boulder 30 are about 6-7 times heavier and significantly larger in size so we can reject it.

Thus, Voltaic Systems was found to be the best panel we could use for this sprinkler system. It is also important to decide the right battery for this system.

The tables below describe the interface and features of each battery, and cost with the panel and battery:

Battery Model	Features	Protection	Battery Capacity (mAh)	Battery Output (V)	Battery Size (in.)	Cost
V15	1.Lithium Polymer 2.Power button 3.Charge Indicator 4.1 USB output port to charge devices 5.1 USB input port to charge the battery	1.Short circuit 2.Over charge 3.Over discharge 4.Over current 5.Over temperature	4,000	5	4 x 2.5 x 0.5	$89 (with panel)
V44	1.Lithium Polymer 2.Power button 3.Charge indicator 4.2 USB output ports 5.DC input to charge the battery		12,000	5	4 x 4.25 x 0.75	$159 (with panel)
V72	1.Lithium Polymer 2.Power button 3.Charge indicator 4.DC input to charge battery 5.DC output to charge laptops 6.12/16/19V Selector 7.1 USB output to charge devices		19,800	5	7.3 x 5 x 0.6	$259 (with panel)

Table 17: Features and costs of batteries #1

Battery Model	Features	Protection	Battery Capacity (mAh)	Battery Output (V)	Battery Size (in.)	Cost
Venture 30	1.Lithium-Ion 2.1 microUSB input to charge the battery 3.Built-in charging cable 4.Built-in charging tip to connect solar panel 5.2 USB output ports to charge devices 6.Charge indicator	1.Waterproof 2.1 year warranty	7,800	5	4.5 x 3.25 x 1	$169.99 (with panel)
RZ60G-A	1.Lithium-Ion 2.1 microUSB input to charge the battery 3.2 USB output ports to charge devices 4.Charge indicator	1.1 year limited warranty	6,000	5	4.5 x 2.63 x 0.5	$39.99

Table 18: Features and costs of batteries #2

Out of all the combinations listed in the table, we have decided to use the 9W solar panel and the V44 battery. This is because the 6V solar panel generates enough volts to power all the components of the sprinkler system such as the PCB, microcontroller unit, water pump, and all the sensors. And the V44 battery provides enough USB outputs and capacity of 12,000 mAh for a long-lasting charge and to power the system in case if the weather conditions are not ideal for the system to generate power via the solar panel.

In order to receive the current generated by the solar panel, it was mentioned earlier that the photovoltaic cells on the side of the panel are connected to a wire that lets us connect the solar panel to the other devices. Elaborating on that, this solar panel from Voltaic Systems comes with a battery and with all the cables required to connect these two devices. The V44 battery comes with a DC input that allows the cable from solar panel to connect to the battery. From the battery, depending on the device to be powered, a specific cable is required; if we were to connect the battery to an iPhone 7, we would need a USB to lightning cable where the USB part of the cable would be inserted in one of the USB ports of the battery and the other end of the cable to the iPhone. So, in order to power the PCB, we would need a USB to microUSB cable, where the USB part of the cable will be inserted in the USB port of the battery and the microUSB end to the PCB. Since the other components of the system are connected to the PCB, powering the PCB will power the whole system.

3.3.3 Wi-Fi Module

In order to communicate with our device from our smartphone application, we need a way to communicate with it wirelessly from long distances. This includes set up, scheduling for watering, power on/off and more. While Bluetooth was considered for this project, it ultimately did not fill all the needs we had for the device, especially since Bluetooth is a short-range communication. Therefore, we needed to add a Wi-Fi-module onto the board. For an Arduino-based microcontroller, there are two popular options for Wi-Fi modules.

The first one is the Arduino Wi-Fi 101 Shield. This component is a powerful Wi-Fi module with crypto-authentication (developed with ATMEL) that connects an Arduino-based board to the internet wirelessly in a very simple and effective way. It connects to the PCB via SPI port, supports TLS 1.1 (SHA256) and WEP/WPA2 Encryption types and uses the IEEE 802.11b/g/n standard for its connection. It also supports both 3.3V and 5V operating voltages.

The second option is the ESP8266 Wi-Fi Module. The ESP8266 is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to a Wi-Fi network. With its default firmware, it has the same functionality as the Arduino Wi-Fi Shield, but at a lower cost. It is also designed to occupy minimal PCB area, making it smaller than the Arduino Wi-Fi Shield. The only disadvantage of the ESP8266 is that it is not capable of 5-3V logic shifting and requires an external Logic Level Converter in order to work.

	Arduino Wi-Fi 101 Shield	**ESP8266 Wi-Fi Module**
Operating Voltage	3.3V and 5V	3.3V
Wi-Fi standards supported	IEEE 802.11 b/g/n	IEEE 802.11 b/g/n
PCB Connection	SPI Port	SDIO 1.1/2.0, SPI, UART
Size	2.1” x 2.5” x 0.93”	1.0” x 0.6” x 0.04”
Price	$49.95	$6.95

Table 19: Wi-Fi Module comparison

For this project, we have decided to go with the ESP8266 since it has a lower cost ($6.95), takes up less space and still maintains all the same functionality from the Arduino Wi-Fi Shield.

3.3.4 Water Pump

For our water pump selection it was imperative to have a pump that was able to move enough water at a good rate to have a great projection of water which could be controlled to cover the patio or balcony of an apartment. While taking into account the possibility that the voltage being produced for the pump may not be able to reach the highest max it can handle. Therefore, we had to find a pump that could produce a great flow with a wide range of turn on voltage.

Cost is a major factor to be considered when looking at any of our components and our range of prices for one spanned from $5 to $25. Within this range it was obvious the amount of water being moved would be nearly the same at around 65 gallons per hour. This creates a good flow and will allow us to direct the spray direction and distance with ease by purchasing or 3D printing a simple nozzle.

With flow rate considered to be a non-issue with our choice in pumps, the comparison of turn on and operational voltages began. Most water pumps are able to operate around 12 Volts input which may not be able to be completed at all times. With the potential of battery failure or a part of the PCB drawing too much power it is important to take into account a range at which pumps can operate. The two best at doing this were the Lightobject EWP-7L9 and the 3M Water Circulation Micro Brushless Water Pump. The Lightobject could operate at small voltages which is perfect for our low-power design but would essentially still need a boost converter to operate it. With a greater range of operation voltage and a difference of only $3.00 compared to the Lightobject it was the obvious choice for us to use the 3M Water Circulation Micro Brushless Water Pump to drive our plant showering aspect of the design.

Pump	Voltage	Cost	Size	Gallons per hour	Power Consumption
Lightobject EWP-7L9	6-9V	$8.95	30 x 78.3 x 10 mm	65 G/H	12W
Mavel Star 12 volt	12V	$10.99	10 x 6 x 8.8 cm	63 G/H	6W
Anself Ultra-quiet Mini	12V	$10.35	5.2 x 4.6 x 5.5 cm	63 G/H	4.8W
Uniclife 80 GPH Submersible Water Pump	110-120V AC	$7.99	53.3 x 43.2 x 33.02 mm	80 G/h	4W
3M Water Circulation Micro Brushless Water Pump	5-12V	$12.00	50.8 x 45.7 x 55.9 cm	63 G/H	2W
FORTRIC ZKWP01 60GPH DC 12V	12 V	$10.99	51 x 34 x 42.7mm	60 G/H	4.2W

Table 20: Water pump comparison

3.3.5 Sensors

Humidity

TE Connectivity Measurement Specialties makes a humidity sensor (HPP845E031R5) that also sports a temperature output. It comes in a small footprint, no bigger than 3 x 3 x 1 mm, and their sensor is made with microcontrollers in mind due to their low power and small physical footprint. These sensors only cost $1.33 each, but unfortunately a minimum order of 5000 is required for purchase.

Honeywell develops a great sensor with the HIH9000 series. With a ±1.7% relative humidity accuracy, ±0.3°C accuracy within its temperature range, and a perfect operating range, this sensor would be perfect for us. It would come out on top and be our choice were it not for the $68.34 price tag. So we decided to sacrifice some of this accuracy for a lower cost alternative.

Aosong Electronics makes a good sensor, the DHT22, that combines temperature and relative humidity sensing and has a digital output. This digital signal is calibrated and the sensitivity of humidity measurements is 0.1% of the relative humidity. This sensor is larger than the two previously mentioned sensors which makes it easier to solder. It is also a relatively cheap $9.95 with no minimum quantity needed to order, so we are happy to sacrifice some size advantages and accuracy for this cost.

Sensor	Humidity Range	Operating Temperature Range	Measurement Accuracy	Current Consumption (in-use)	Cost
HTU21D	0% to 100% RH	-40°C to 125°C	±2% RH	0.45mA	$1.33
HIH9000	0% to 100% RH	-40°C to 125°C	±1.7% RH	0.65mA (I^2C) 0.75mA (SPI)	$68.34
DHT22	0% to 100% RH	-40°C to 80°C	±2% RH	1.25mA	$9.95

Table 21: Humidity sensor comparison

Temperature

DFRobot makes an interesting waterproof temperature sensing kit that works from -10°C to 80°C.This temperature range is good but the catch is that the sensor comes as part of a three-conductor system and has a long length of wire. Unfortunately, it's too big for us to use and. had it been chosen, may have made it more difficult for a prospective user to set up.

TDK Corporation makes an NTC thermistor (B573xxV2 series) that's available for sale, however we ran into the same problem of not meeting minimum order quantities on it. It has a good operating temperature, but since we haven't fully designed our PCB yet we were unsure as to whether or not it's 10000-ohm resistance would negatively impact performance. Since it is a thermistor, it is extremely accurate and temperature values can be attained through a lookup table.

The DHT22 sensor from Aosong comes with a DS18B20 digital temperature sensor. This is based on semiconductor integrated circuit technology and can provide between 9 and 12 bits of temperature measurements. At 9-bits of resolution, the DS18B20 can measure increments of 0.5°C where at its highest (and default) resolution of 12-bits it can measure changes as small as 0.0625°C. After powering up to a low-power idle state, a simple command can invoke a temperature measurement and an analog to digital conversion for us. We can then read this signal using the microcontroller. For the temperature range we are operating in, the temperature reading variance is just ±0.5°C. By deciding to use the DHT22 for its humidity sensing capabilities we are also going to take advantage of its temperature sensing capabilities, so the cost and ease of use for this two-in-one sensor makes it an obvious choice, especially since it is still respectably accurate.

Sensor	Resolution	Operating Temperature Range	Measurement Accuracy	Current Consumption (in-use)	Cost
Waterproof DS18B20	0.5°C to 0.0625°C	-55°C to 125°C	±0.5°C within -10°C to 80°C	1mA	$6.90
B573xxV2	-	-55°C to 125°C	Excellent	-	$0.14
DHT22 (DS18B20)	0.1°C	-40°C to 80°C	<±0.2°C	1.25mA	$9.95

Table 22: Temperature sensor comparison

Combined Sensors

Overall, the BME280 from Bosch Sensortec blew us away. At $7.58 per unit and with no minimum orders, the performance to price ratio of this sensor was too hard to ignore. This sensor combines temperature, pressure, and humidity into one discrete package. While we would have liked to keep each sensor separate on the PCB, having them all integrated in one mount saves us physical space on our PCB, allowing it to be more portable. It somehow still comes in a smaller package than other standalone and two-in-one sensors that we saw, measuring only 2.5 x 2.5 x .93 mm. It can communicate using either I2C or SPI, operates from 1.7 to 3.6V, draws 1.8 to 3.6 microAmps depending on sensors being used, and draws only 0.1 microAmps when in sleep mode. It can operate from -40°C to 85°C, 0% to 100% relative humidity, and can handle 300 to 1100 hPa of pressure. It has a low response and is relatively accurate. There were so many positives to this sensor that we felt it trumped finding separate sensors that may be worse to do the same jobs.

Unfortunately, for as much as we loved the Bosch three-in-one sensor it had one major drawback. After we received the parts in the mail we saw just how small the package really was. It became immediately obvious that this would be tough to solder, so we took to looking up the soldering recommendations. Our hopes of using this sensor were soon put down when we found out that the precision of our soldering would need to be on the level of micrometers. This was a major oversight on our part and we are currently looking to return these otherwise wonderful sensors.

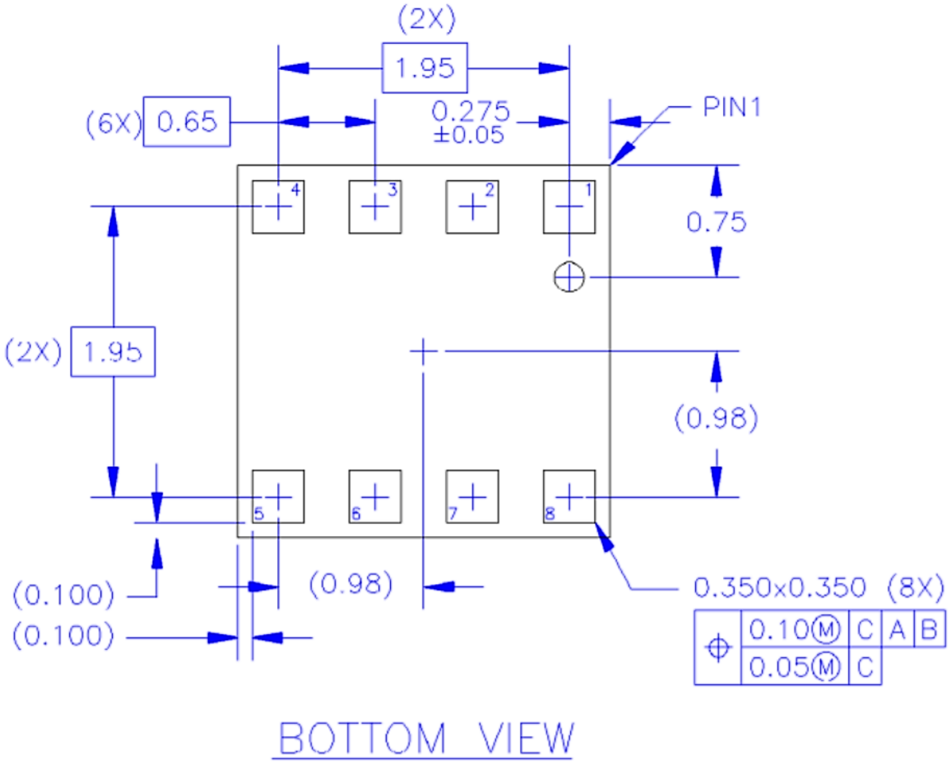


Figure 29: Soldering sections of BME 280

Source: https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280_DS001-11.pdf

Enter the DHT22, a two-in-one temperature and relative humidity sensor that helped keep our project on track with its easy to solder pins. It is actually more accurate when measuring relative humidity, ±2% compared to ±3% for the Bosch sensor, and more accurate when measuring temperature, <±0.2°C compared to ±1.0°C for the Bosch. When we sat down to think about why the DHT22 sensor seemed to beat out our previously favored Bosch sensor, we came up with two possible explanations: size and cost. The DHT22 sensor costs more money than the Bosch sensor and also possesses a larger footprint. This likely makes it easier for the DHT22 to have better sensing technologies since the focus isn't on making the sensor as small as possible.

Water Level Sensor

For the water level sensor, originally, we decided to go with a small ultrasonic sensor due to its cost, size, and ease of implementation. The ultrasonic sensor we had decided to go with was the ELEC Freaks Ultrasonic Sensor HC-SR04 which cost $3.95. However, after finding out about water level strip sensors, we decided that we wanted to try one out. We found that eTape offered a few different versions of these continuous fluid level sensors, and we decided on the PN-12110215TC-12. It cost a bit more money, coming in at $39.95 for one sensor plus a resistor and required pin connector.

We felt that, despite the cost, using this type of level sensor would be beneficial to our project. It is more accurate than an ultrasonic sensor, with none of the drawbacks. As long as the strip sensor fits into the tank, it can measure any amount of liquid in it, whereas the ultrasonic sensor needs a minimum distance between it and the liquid. This fact alone should make it easier for end users to set up in their tanks, they won't have to mess around getting the ultrasonic sensor mounted correctly to the top of the tank. The strip sensor can be placed on the side of the tank and attached with tape and works as long as the pins are not submerged.

Sensor	Sensing Range	Operating Temperature Range	Current Consumption	Cost
HC-SR04	2cm to 4m	-	15mA	$3.95
PN-12110215TC-12	31.5cm	-9°C to 65°C	50mA	$39.95

Table 23: Water level sensor comparison

3.3.6 Voltage Regulator

This component is necessary for reducing our voltage from our battery down to have our microprocessor and other sensors work properly. A voltage regulator reduces the voltage safely to help protect our components and can also have a built-in feed-back system so that if a current is being drawn where it shouldn’t the system can protect itself. [9]

There are several circuits we could have used to maintain a voltage of 3.3V, but the following are the three different types of examples we compared:

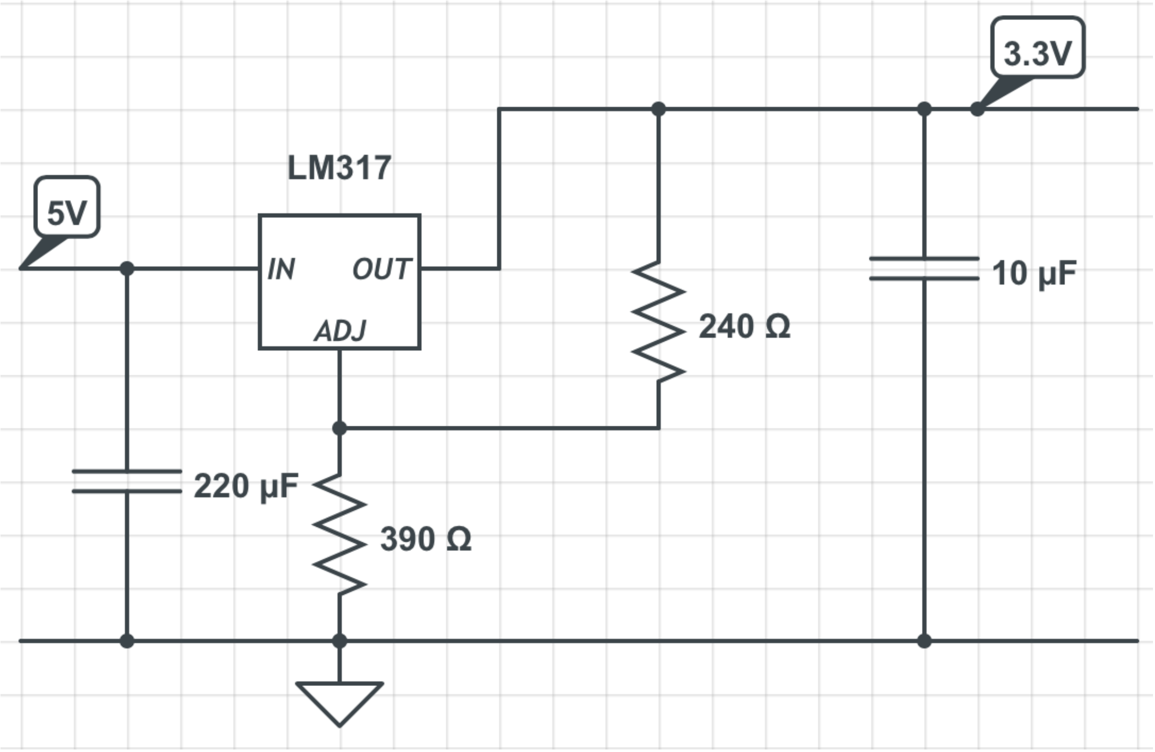


Figure 30: Circuit using LM317 voltage regulator (A)

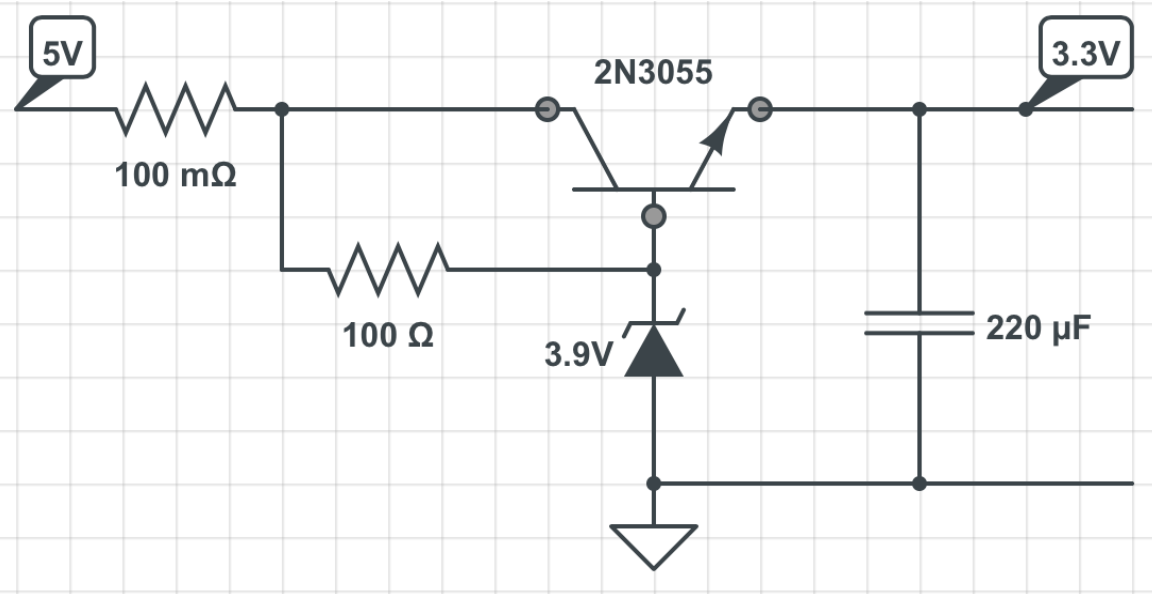


Figure 31: Circuit using N-P-N transistor (B)

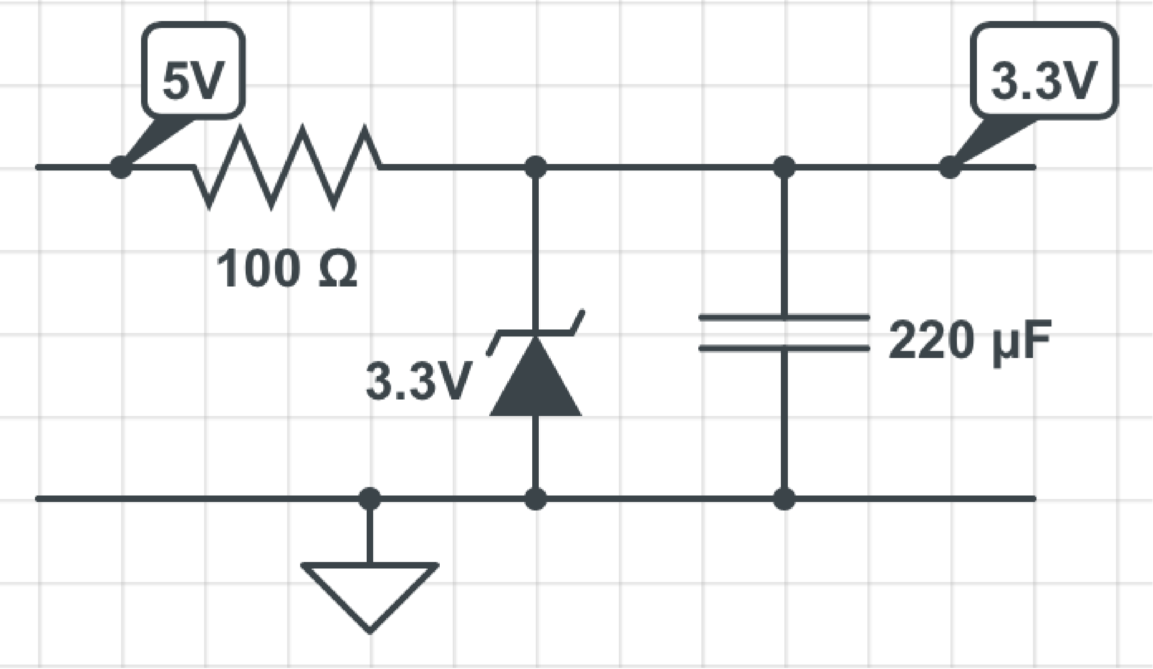


Figure 32: Circuit using Zener diode (C)

Circuit Type	Advantages	Disadvantages
A	1.Wide range of input voltage between 5V to 25V DC. 2.This kind of voltage range can accommodate most of the microcontrollers. 3.1.5A of current is decent amount of current flow to expect, and it will also cause no heat problems.	1.Current deliver not as high as some other circuit types.
B	1.Very fast circuit. 2.Can deliver current of 5A or more.	1.Very small input voltage range, only some microcontrollers can be compatible with this range. 2.Faster current flow means more heat; if there is no proper heatsink, then the material can be damaged.
C	1.Simple and easy to build. 2.Doesn't draw more power due to very minimal current flow. 3.Can still power up some controllers even though low-profile.	1.Very small voltage input range; 5V to 6V. 2.Slow circuit; only up to 300mA.

Table 24: Circuit type comparison

Based on the comparisons listed above, we can see that the best type of circuit to use for this system is a voltage regulator circuit. This is because it provides wide input voltage range and enough current flow rate, meaning that it will be capable of converting any voltage signal between 5V to 25V, to a 3.3V supply. So, more modules and microcontrollers can be safely powered by a voltage regulator. Also, even if the voltage regulator was used by itself to make the conversion, it would be able to do so without adding more components such as capacitors and resistors because voltage regulators come with built-in resistors. This would save lots of space on the breadboard and draw less power. The LM317 voltage regulator will be used to regulate a 5V supply to a 3.3V supply.

3.4 Part Selection Summary

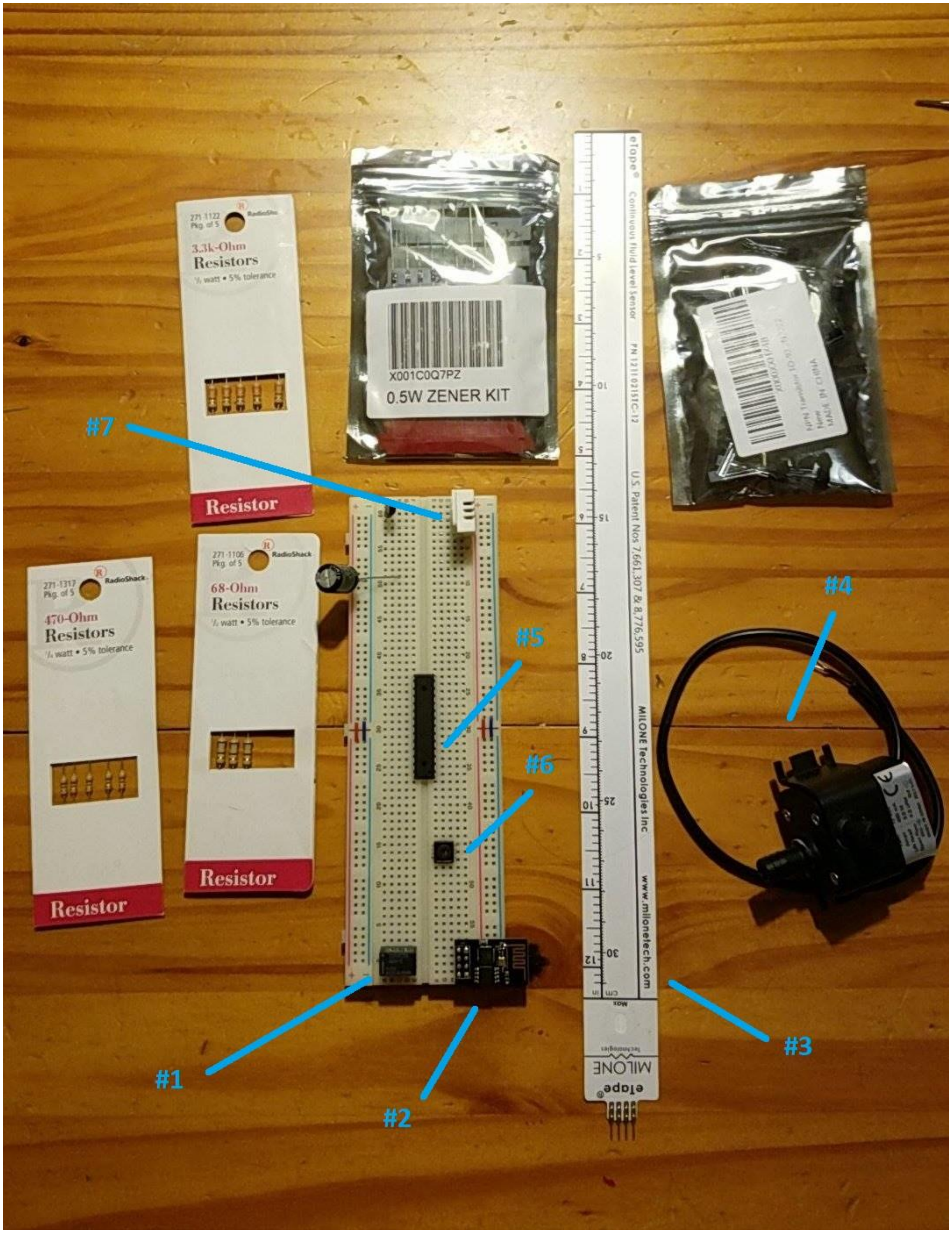


Figure 33: Received components

When starting out, we had to set some expectations for the project. Any water pump we would select needed to be low voltage and draw a low amount of current to save battery power while simultaneously providing enough pressure to propel a stream of water at least five feet. We had to find a solar panel that could output enough energy to charge the device's battery and be able to power the device on its own. We set expectations higher when we asked ourselves to find a microcontroller capable of being able to handle all sorts of input from sensors, accept power from the battery and solar panel, drive the water pump, and still be small enough to fit into a portable container. We also wanted to design a device that would be easy for anyone to assemble and be user friendly. With these expectations in mind, we set out to find the right components for our device while making the least amount of compromises.

For our microcontroller, the ATSAMG51 is the microcontroller that stood out among the rest because of its performance and price. It features a clock rate of 48 MHz, a low-power mode of 103 microAmps/MHz and only costs $3.14. (Item #5 in figure above)

For our solar panel and battery combo, of all the combinations listed in the comparison tables, we have decided to use the 9W solar panel and the V44 battery. This is because the 6V solar panel generates enough volts to power all the components of the sprinkler system such as the PCB, microcontroller unit, water pump, and all the sensors. And the V44 battery provides enough USB outputs and capacity of 12,000 mAh for a long-lasting charge and to power the system in case if the weather conditions are not ideal for the system to generate power via the solar panel. The price for the solar panel and battery combo is $159, making it our most expensive component.

For the Wi-Fi Module, we have decided to go with the ESP8266 since it has a lower cost ($6.95), takes up less space and still maintains all the same functionality from the Arduino Wi-Fi Shield. (Item #2 in figure above)

For the water pump, we saw that with a greater range of operation voltage and a difference of only $3.00 compared to the Lightobject it was the obvious choice for us to use the 3M Water Circulation Micro Brushless Water Pump to drive our plant showering aspect of the design. The price of this pump is $12. (Item #4 in figure above)

For the water level sensor, originally, we decided to go with a small ultrasonic sensor due to its cost, size, and ease of implementation. The ultrasonic sensor we had decided to go with was the ELEC Freaks Ultrasonic Sensor HC-SR04 which cost $3.95. However, after finding out about water level strip sensors, we thought it may be easier for our end users to set up so we decided to give it a try.

We found that eTape manufactures different versions of these continuous fluid level sensors, and we decided on one of their 12-inch versions. We felt that this may be large for some tanks, but the sensor has no problem sticking out the top of a tank and its size makes it usable across a wider variety of tanks. It cost a bit more money, coming in at $39.95 for one sensor plus a resistor and required pin connector but we felt that, despite the cost, using this type of level sensor is beneficial to our project. It is more accurate than an ultrasonic sensor, with none of the drawbacks. As long as the strip sensor fits into the tank, it can measure any amount of liquid in it, whereas the ultrasonic sensor needs a minimum distance between it and the liquid. End users won’t have to mess around getting the ultrasonic sensor mounted correctly to the top of the tank, the strip sensor can be placed on the side of the tank and attached with tape and works as long as the pins are not submerged. (Item #3 in figure above)

For the temperature, pressure, and humidity sensors we originally decided to go with the BME280 from Bosch Sensortec which combines them all into a discrete little package for only $7.58. Unfortunately, as we mentioned earlier we realized we would be unable to use the BME280 once we received them in the mail. Instead, we are using the DHT22 from Aosong for temperature and relative humidity sensing (Item #7 in figure above). For barometric pressure sensing we are using the KP235 analog sensor from Infineon which cost $8.43 (Item #6 in figure above). While each of these cost more money and take up more space than one of the Bosch sensors, we will actually be able to use them effectively for this project and take advantage of their increased accuracy.

Lastly, we have item #1 in the figure above. This item on the breadboard is a simple relay. This relay, along with the resistors and diodes seen around the breadboard are some of the other miscellaneous items needed to bring our device to life. Since those are general electrical components, they were not really discussed in the parts selection.

4. Standards and Design Constraints

In order to have a consistent design and building process a set of standards must be implemented and adhered to by each member. These standards are used to help navigate around the constraints that could be tied to this project. Constraints can range from financial, size, power consumption, to any of the numerous ways the project could be limited. It is the team's duty to be wary, research, and work around these constraints.

The following section goes over:

- A list of standards for the team to use when designing the project.
- Constraints tied to this project and how they may affect the final design.

4.1 Standards

An engineering standard is what specifies the requirements that must be met by the products, materials, processes, and services to fit their purpose. The purpose of standards is to promote safety, reliability, productivity, and efficiency. These standards are created by experts from particular fields of study. Standards are important and effective because following them creates a common language amongst engineers. Manufacturing costs can be lower if procedures are performed according to the standards.

4.1.1 Solar Technology

Standard	Description
IEC 61215	A standard used in the development of photovoltaic modules. This standard was implemented to decrease the field failures in development of crystalline silicon cells; failures such as broken cells, corrosion, broken glass used to occur due to poor manufacturing processes and methods. IEC 61215 standard consists of several qualification tests that the photovoltaic modules have pass through during manufacturing, and if the modules pass these tests then they are more likely to survive when they are being used and will not have design issues that lead to damages.
ISO 9488	A standard formed by a technical committee at International Organization for Standardization that consists of basic vocabulary that is to be used when we discuss solar energy in a scientific manner.
UL 1703	Developed by Underwriter Laboratories. This standard covers safety issues that occur while mounting and installing the modules.
UL 1741	A standard that covers the use of inverters, converters, charge controllers, and interconnection system equipment in a grid-connected or stand-alone PV modules.

Table 25: Solar Technology standards #1

Standard	Description
UL 2703	A standard for mounting systems and clamping devices that are used with photovoltaic module systems that has a maximum voltage of 1000V.
IEEE 1526	A standard that tests the performance of stand-alone photovoltaic systems.
IEC 61836	A standard that provides terms, definitions and symbols for solar photovoltaic energy systems.
IEC 60904-1	Measurements of PV current-voltage characteristics for photovoltaic devices.

Table 26: Solar Technology standards #2

4.1.2 Wireless Technology

Standard	Description
IEEE 802.11a	Wireless network bearer operating in the 5 GHz ISM band with data rate up to 54 Mbps.
IEEE 802.11b	Wireless network bearer operating in the 2.4 GHz ISM band with data rates up to 11 Mbps..
IEEE 802.11g	Wireless network bearer operating in 2.4 GHz ISM band with data rates up to 54 Mbps.
IEEE 802.11n	Wireless network bearer operating in the 2.4 and 5 GHz ISM bands with data rates up to 600 Mbps.
IEEE 802.11ac	Wireless network bearer operating below 6GHz to provide data rates of at least 1Gbps per second for multi-station operation and 500 Mbps on a single link.

Table 27: IEEE 802.11 standards

4.1.3 Sensors

Standard	Description
IEEE 2700	A standard for sensor performance parameter definitions.

Table 28: Sensor standards

4.1.4 Battery Standards

Standard	Description
UL 2575	A standard for lithium ion battery for use in electric power tool and motor operated, heating and lighting appliances.
IEC 62133	A standard that sets the rechargeable cell and battery safety standard for Li-Po batteries. This includes four tests: Molded Case Stress, External Short Circuit, Free fall, and overcharging. Easy to pass but still very important
UL 2054	This standard includes 18 tests in order to makes sure that batteries meet U.S. device standards applicable to household items. This increase the safety of each battery pack and lowers the risk of bringing it into public and private spaces.
BS EN 60086-4:2000	This is a safety standard for lithium batteries if they are used as a primary source. This helps us keep a safety check on our device that is being powered through the battery.
BS EN 61960-2:2002	This standard applies to portable applications. Since our device is supposed to be moved and stored in different environments it is crucial to understand how they work portably.
BS EN 60086-4:1996	This standard is a safety standard for lithium batteries if they are used as a primary source.
UL 1642	This is a safety standard implemented during the testing of lithium-ion batteries.
IEEE 1625	This is a standard for rechargeable batteries for multi-cell mobile computing devices.
IEEE 1725	This is a standard that checks the quality and reliability of rechargeable lithium-ion and lithium-polymer batteries to be used in cell phones.

Table 29: Battery standards

4.1.5 Java Coding Standards

In order to build good java programs, it is important for development teams to adopt proven design techniques and good coding standards. Following these standards allows for code consistency across the world, making it easier to understand, develop and maintain an application or program. [15]

File Headers:

- All source files should begin with c-style headers containing the title, version, date and copyright information.
- Headers should be followed by the package and import statements and then the documentation comments.

Code Formatting:

- Four spaces should be used as the unit of indentation.
- Indentation pattern should be consistently followed throughout the code.
- The starting brace should be at the end of the conditional.
- The ending brace must be on a separate line and aligned with the conditional.
- Lines longer than 80 characters should be avoided.
- When an expression does not fit in a single line, it should be broken according to these general principles:
    - Break after a comma or operator.
    - Prefer higher-level breaks to lower-level breaks.
    - Align the new line with the beginning of the expression at the same level on the previous line.
- Two blank lines should be used in the following circumstances:
    - Between sections of a source file.
    - Between class and interface definitions.
- One blank line should always be used in the following circumstances:
    - Between methods.
    - Between the local variables in a method and its first statement.
    - Before a block or single-line comment.
    - Between logical sections inside a method to improve readability.
    - Before and after comments.

Comments:

- Implementation comments are delimited by /*…*/ or //.
    - A double dash is recommended for commenting out code.
    - For multiple or single-line comments given as overview of code, /*…*/ should be used.
- Comments should have four styles of implementation comments as follows:
    - Block comments.
    - Single-line & trailing comments.
    - Commenting codes
    - Documentation comment.

Code Style Rules:

- Don’t ignore exceptions.
- Don’t catch generic exception.
- Don’t use finalizers.
- Fully qualify imports.

- Write short methods.
    - If a method exceeds 40 lines or so, think about whether it can be broken up without harming the structure of the program.
- Define fields in standard places.
- Limit variable scope.
- Order import statements.
- Use standard java annotations.
- Treat acronyms as words.
- Log sparingly.

These are just some of the standards that should be followed when programming in the Java language to make sure that the code follows the correct guidelines accepted by other developers.

4.1.6 Soldering Standards

The National Aeronautics and Space Administration(NASA) hold a high standard when it comes to soldering and surface mounting components. Adhering to these standards will allow us to create a better and smoother solder and printed circuit board job. Ranging from proper cleaning of equipment to soldering heads to proper solder, NASA creates a great references of standards in their paper "Soldered Electrical Connections: NASA Technical Standards". Some of the standards upheld from this paper are as follows: [16]

- Cleanliness of workstation and tools must be upheld in order to achieve the best working tools and environment.
- The parts will be mounted in the position on the PCB that has been made clear from the manufacturer as to how they should be placed. This makes sure the parts work in their proper order and do not short out due to wrong connections.
- Values and labels of parts that are printed onto the PCB shall be visible after soldering to the best of the operators ability. This allows for a more coherent PCB and so others can know what parts are mounted where after soldering is completed.
- The parts will be mounted in parallel to their mounting surface. This insures that the connections to each other are connected properly and not in series which would potentially cause the part to malfunction.
- The part shall be secured in place during the soldering process, which include the soldering cooling down and solidifying. With minimal movement at the conductors the part is able to set itself properly on its plate without moving and coming off its proper mounting station.
- Connections will be cooled at room temperature over time, without use of external pressurized air. The use of these tools could cause the hot solder to come off and burn a person or object near by. For safety this will be followed strictly.
- The solder should cover the conductor and the termination area on the PCB. Too much solder over the terminal and conductor could short out the area and cause the part to malfunction. This can also cause the connections to be mixed and have pins become connected together.
- Residue of solder will be cleaned off after a half hour of the solder cooling using solvent if necessary. This helps with creating a better connection between the terminal and the pins.

- If a PCB has a broken pattern or damaged connection it will not be repaired. This can damage the whole PCB, instead call the manufacturer and request a new board be sent out.
- For through hole connections heating can be applied at both sides of the whole but solder will be applied to only one side. This standard saves the board from potential shorting and damaging of a component.

The University of Cambridge produced a paper with safety standards that should be used under the paper titled “Soldering Safety”. While working on our printed circuit board, the teammates that will be soldering will stand by these safety standards and work around them at all costs. Not complying with these standards could result in harm to the group mate soldering or others in the area. [17]

- Never handle the soldering iron element at any cost. The iron can reach up to 500 degrees Celsius and can easily cause third degree burns.
- Any element that will be soldered should be handled with tweezers and not the user so as not to burn one’s self.
- Keeping the iron clean using a wet sponge reduces the risk of splash back from residual flux that may be on the element.
- When the iron is not in use return it to its stand immediately.
- Do not use damaged irons, this can cause an electrical shock to the user.
- Work on fire-proof workbenches to reduce risk.
- Collect soldering waste and dispose of it in a hazardous waste bin on campus.
- Work with lead-free solder as to reduce the risk of lead exposure and dangerous fumes in the air.
- Wear eye protection to prevent splash back from flux getting into the team members eyes.
- Always wash your hands when you’re done soldering to get rid of any potentially harmful chemicals on you.

4.1.7 International Software Testing Standards

Standard	Description
ISO/IEC 25000:2005	Provides guidelines for software quality requirements, known as SQuaRE. It helps organize requirements and provides a way to evaluate them.
ISO/IEC/IEEE 29119-2	This standard covers test processes and aims to define a generic model for software testing. This allows it to be used in almost any situation almost anywhere within the software development life cycle. This is a three layer model detailing organizational tests, test management, and dynamic testing.
ISO/IEC/IEEE 29119-4	This standard defines one international standard that envelops testing techniques for software design. These techniques can be used to help uncover test cases which can then be used to collect data about system requirements. See list of specific techniques following this table.
ISO/IEC/IEEE 29119-5	This defines a standard for keyword-driven testing. This is beneficial because it defines test cases using predefined keywords. These are associated with specific actions needed to perform a step in a test case, helping to identify and compose test cases.
ISO/IEC 33063	Contains indicators that can help when implementing process improvement programs or evaluating models, methodologies, and tools.

Table 30: International Software Testing Standards

The following is the testing list for the ISO/IEC/IEEE 29119-4 standard: [18]

Specification-Based Testing Techniques:

- Equivalence Partitioning
- Classification Tree Method
- Boundary Value Analysis
- State Transition Testing
- Decision Table Testing
- Cause-Effect Graphing
- Syntax Testing
- Combinatorial Test Techniques, including:
    - ➔ All Combinations
    - ➔ Pairwise Testing
    - ➔ Each Choice Testing
    - ➔ Base Choice Testing
- Scenario Testing (including Use Case Testing)
- Random Testing

Structure-Based Testing Techniques:

- Statement Testing
- Branch Testing
- Decision Testing
- Condition Testing, including:
  - ➔ Branch Condition Testing
  - ➔ Branch Condition Combination Testing
  - ➔ Modified Condition Decision Condition (MCDC) Testing
- Data Flow Testing, including:
  - ➔ All definitions
  - ➔ All-c-uses
  - ➔ All-p-uses
  - ➔ all-uses
  - ➔ all-du-paths

Experience-Based Testing Techniques:

- Error Guessing

4.1.8 Hardware Testing Standards

Although we did not find any IEEE, ISO, or IEC backed standards for testing hardware, there are still many applicable principles that can be used and are used by many individuals and organizations to test their hardware. These tests usually include unit, integration, functional, regression, and acceptance testing.

Unit testing involves our low level design and code structures. Each individual piece, or unit, was tested here. This means any and all methods inside our code were explicitly called and inspected to make sure that they perform their intended functions. If they are not returning the correct values they will need to be modified. Each piece of hardware also needs to be tested individually. For sensors and other electrical components, we tested them on a breadboard with a known, quality power supply to make sure they are functional. Once each part passes its unit test we can begin moving on to the integration phase of tests. [19]

Integration tests are one step above unit tests. Once we have made sure each piece of hardware and each method in the code works individually, we start combining them all and testing to see how they start working together. Before we finalize our PCB schematic and send it off to be built, our hardware integration tests will be done on a breadboard. Once again we have a steady power supply and will begin to slowly add components onto the board, testing each one every step of the way to make sure nothing breaks or interferes with another piece of hardware along the way. Once we get all the components hooked up we can start to move on to functional testing.

Functional tests continue the trend of moving away from the lower levels of hardware and software. Functional tests include examination of requirements and making sure proper tests are run to validate those requirements. Like its name, we want to make sure the overall functionality of the hardware and software are working together to begin meeting our goals for the project. We also conducted stress tests. For hardware this may include testing in harsh environments, almost up to the sensor’s limits, if possible. We may also try running the device continuously even though that is not how it is meant to be run.

Regression tests were ran early and often. These tests were ran to make sure that any new component or code additions had not adversely impacted components or code that was already there and working before. Regression tests are better suited to software after major bug fixes are applied, but they can still be useful in our case for hardware to help us spot if we have wired anything incorrectly or miscalculated the amount of power we need or where something needs to be placed in our circuit.

Acceptance testing typically takes place once the customer receives the product. Acceptance tests are usually formal tests performed by the end user, but in our case since we are the first end-users of this device, we will be the ones performing acceptance testing. We tested to make sure all of our technical and marketing requirements have been met, especially the ease of use of our device and its performance and portability. What we want to do here is think of as many real world usage scenarios as we can and then test them, noting whether or not or hardware and software work as expected, if either or both of them need a redesign or experience any glitches, leakages of power, or anything unusual. In a real world scenario we would want to make sure our hardware and software are ready to experience everything we said they would and more.

4.2 Realistic Design Constraints

4.2.1 Time Constraint

Time is a very precious item a lot of us take for granted. With every group member taking multiple classes, working and some even having extracurricular activities to attend for an organization, time is definitely something we could use a little more of. With so little free time, time management and planning are two of the main values we try to improve everyday so we do not fall behind with the project. By setting up deadlines and milestones, we have been able to stay on track and focused during this planning stage.

4.2.2 Funding Constraint

As previously seen in the “Strategic Components and Part Selections” section of this document, cost was a big factor in the decision of which parts to go with. Of course choosing the best parts in each category would have been awesome and possibly made things easier in the long run since nicer parts usually have better features, however, being college students and not having any sponsors made this difficult for us. Even though we had to settle for the cheaper alternatives to some parts, we believe we have all the tools needed to make this project a reality with the least amount of money possible. In the end, we needed to prioritize cost just a little bit more than function. These budget-conscious decisions should help keep the total cost of the project under $500, which means no group member will pay more than $125 out of pocket.

4.2.3 Size Constraint

The goal of this project is to make it as compact is possible. In some cases, this was not possible, as we figured out after receiving our first shipment of sensors. The Bosch BME280 three-in-one sensors that we originally received and planned to use as temperature, pressure, and humidity sensors were too small. While they would save a massive amount of space on the PCB compared to other individual sensors, once we had them in our hands we saw that it would basically be impossible for any of us to solder the pins of the sensor. Once we realized this we immediately took to researching alternative sensors. These ended up being larger and more expensive, but it wasn't such a big hit to our PCB space or our wallets, and we would much rather have a working project. Another factor playing into the size of our project is, specifically, the solar panel. The length and width have to be a certain size given the dimensions of the solar panel, as a group we made sure not to design these features. Depth of the project we were able to manipulate. We wanted enough room between the water tank, electronics box and solar panel so that condensation from the water tank won't affect either of them. From the solar panel we are given a size of 10.1 by 8.7 inches and we determined a depth of 6 inches under the solar panel to house the water tank and electronics box was sufficient enough to satisfy the concerns we had. Therefore nothing outside of these dimensions can be added to this project.

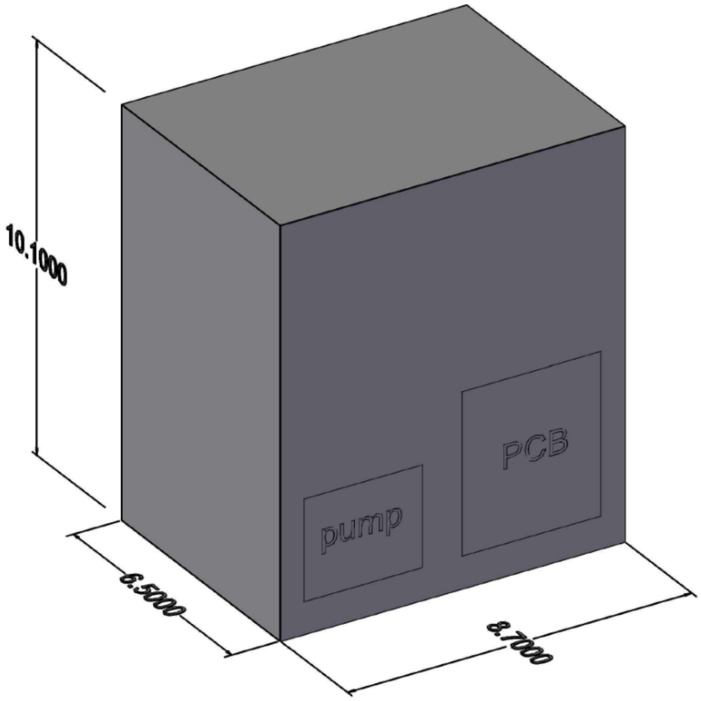


Figure 34: Size constraints associated with project

4.2.4 Power Constraint

A major factor of this design is the fact that it is self sustaining. In order to achieve that, the solar pane provides a certain amount of power to keep the system running. With the solar panel we have picked out a max power output is at 9W. This means we have to keep the power consumption of all the parts of the design to a minimum. Anything exceeding this maximum amount of power draw and the design will fail to work at all. This is also an important factor in keeping the battery charged longer. The more power drawn the faster the battery drains. This limits the amount of peripherals added to the design and the types of components we can use. For instance a water pump limited to a lower voltage.

	Volts	Current	Power
Solar panel/battery	5 V	2 A	9 W
Microcontroller	3.3 V	9.46 mA	11.8 mW
Wi-Fi Module	3.3 V	80 - 215 mA	264 - 709.5 mW
Water Pump	5 - 12 V	0.4 A	2 - 4.8 W

Table 31: Power consumption of each major part

4.2.5 Part-Availability Constraint

One constraint that was not really thought of at first is the availability of the parts and components we planned to get. While doing research, we were able to narrow down our selections for each component technology and then proceeded to finding the actual components that would fit our needs. To our surprise, a lot of the components we looked at were out of stock temporarily. This would not be a big deal if we had a lot of time to create this project. However, with a deadline of just a little over 4 months, parts need to be ordered as soon as possible so they can start being tested and implemented. We don’t have the luxury of waiting for parts to become available again so we can order them. For this reason, a few components had to be chosen based on availability and not necessarily because they were the absolute best option.

4.2.6 Soldering Constraint

Since the parts will be hand soldered onto the printed circuit board it is important to select parts that are able to be handled properly and soldered correctly onto the board. Pins should have a fair amount of distance between them so the risk of soldering the pins together or incorrectly is down to a minimum. Also, the parts themselves should be large enough to be handled by the person soldering. To small of a part and the device might not be able to be soldered to its ports properly.

Since the soldering will be done by teammates accuracy of soldering may be considerably lower than a machine soldering job. This requires us to use precision and patience in order to achieve the best form of solder connections. While this saves on cost for the project it leaves room for a lot of errors with the connections between all of the parts on the PCB. One small mistake can short the component or worse the board and the group would have to restart their PCB with a fresh board.

4.2.7 Water Constraint

Since we are integrating water with electronics it is imperative to be careful while using great caution with handling them. This means that the freedom of using any form of watering container is out of the question. Using a random one runs the risk of spilling water into the electronics box next to it. Waterproofing was also be a major task taken by the team seriously, if water is to get into the electronics box the system will fail. This narrows down the type of boxes we can use to secure our design. Also, a small gap between the solar panel and water tank is necessary so that evaporated water isn’t affecting our solar panel above it at all.

4.2.8 Microcontroller Memory Constraint

The atmega328P has a small internal memory which blocks it from doing heavy processing. It’s able to read and store data from sensors as well as drive the relay for our pump, but any major calculations must be done outside of the processor. This means we have to find a way to communicate with a device that can process the data our processor can’t, such as converting the digital data to the proper units. Therefore, we must use a Wi-Fi module in order to communicate with those heavy devices.

4.2.9 Battery Constraint

The battery that is selected is a 12000mAh Li-Po battery. While this holds a charge well and can output current, it can’t supply a constant current and voltage to a great deal of components for long. This puts a limit on the amount of peripherals that can be added to the overall system. While we could have opted for a larger battery, the size of it would have become to demanding and may not have integrated with our solar panel correctly.

Another battery constraint could be that after many charge cycles or when the battery ages, the battery might become defective. The solar panel and battery both are from Voltaic Systems and they are directly compatible with each other because the V44 battery connects to the solar panel directly via its DC input. So, most of the batteries in the market do not come with a DC input, then the user must by the battery from Voltaic Systems only, or get a battery with microUSB input and use some kind of adapter to convert DC to microUSB.

4.2.10 Software Constraint

With the need to have a fully functioning and pretty form of interface for a user, the ease of design and trade offs made between the two wants typically becomes more complicated and more excessive as time dwindles down or as functions become more complex. This means our project won’t be able to handle high forms of processing from the main hardware and can’t be to over the top with design to help reduce the time it takes to complete the application. With the project deadlines it is imperative to create a simple function and fast design that can be completed with enough time before the design is due to complete testing.

4.2.11 Balcony Constraint

The main challenge about smart systems is that since these are mostly wireless or Bluetooth based systems, every consumer may have a different situation. This is obviously because they have a different size homes or in this case a different size balcony. Average deck sizes of apartments are around 4 feet by 10 feet. Using this information, we can restrict the size of our spray range. We wish not to spray the entire deck so this limits our range of spray field to around 5 feet long and two feet wide. Giving users ample enough room to navigate around the spray and have other furniture or items out on their deck with reduced risk of getting wet.

4.2.12 Height Constraint

The idea of this sprinkler system is unique then what we normally see in houses because this sprinkler has to be at a certain height in the balcony. This is because in balconies residents normally plant using pots, and if we place this tiny sprinkler on the floor it will not water the plant properly, the pot would obstruct the water. However, keeping it at a height is a challenge because in many apartment places residents are not allowed to drill the wall or glue something to the wall. People also wouldn’t have a perfect stand for it. We can design this system in a way that it just hooks to the railings in balconies, but different apartment places may have different railing shapes, some have square shaped and some are cylindrical. So, keeping this device at a height is a challenge.

4.2.13 Water Level Sensor Constraint

Once we decided to use the eTape level sensor we knew there would be some limitations when we needed to implement it. For one, the sensor can not be bent at all or else it will not function properly and we would get inaccurate readings from it. This is because the material inside of the sensor relies on being compressed by fluid to produce a resistance value from the top of the sensor to the top of the liquid. It makes sense that bending the inner material would then cause these values to vary.

The sensor imposes another constraint on us in that it has to be calibrated for each and every unique tank it is used in. The reason for this is the size of different tanks will change the amount of water pressing on the sensor at any given level, changing its resistance. For this reason, we would need to know the minimum and maximum outputted resistance values for each tank so that when we take a water level measurement we can express that value as an accurate percentage of how full the tank is. We can do this because our eTape sensor behaves linearly, so any measurement in between the two extreme values for the tank can easily be expressed as a percentage of fullness. For the purposes of our demonstration this is not really a problem, we will be using the same tank throughout the demonstration so once we get the calibration values we can just hard code them. We are, however, thinking of ways to automate this process in the future so that any user could easily calibrate the device for use any tank.

Finally, this sensor also violates one of our goals to keep the size of this project relatively small. Since the eTape is about 14 inches long, with the sensor itself being 12 inches long, we end up going over one of our self-imposed size constraints by about four inches. Instead of giving up on this sensor altogether, we decided the sensor was interesting enough to warrant a slight relaxation of our size constraints. Plus, once the device is calibrated for the tank it will be used in, its readings will be fairly accurate and quite easy to obtain. We also won't be drawing much power to take these readings, since we do not constantly need to know the exact value of water in the tank, these measurements are only taken and updated every minute or two for the demo (every 30 mins or so for actual consumer product).

4.2.14 Social Constraints

The goal of this project is centered around ease of use for the consumer. If the consumers deem this product to difficult to use it could cause the project to fail completely. Therefore, we need to make the user friendliness at the highest priority. Too many steps in set-up or the application is not easy to understand would cause the project to not succeed in the social aspect of our design.

Another social aspect of this machine is the watering spray. To wide of a spray could cause the entire balcony to be covered in water and overall cause a dissatisfaction with the product. To small of a spray and the total amount of plants wouldn't be covered leaving the user again upset. In order to solve this we must make a functioning nozzle design that can be adjusted.

4.2.15 PCB Fabrication Constraint

With many companies, the length of time can vary with the fabrication of the printed circuit board. Unfortunately, this can cause our design to be late if something goes wrong with the manufacturing at the plant or the shipping gets messed up. This puts a rush on the team to design the PCB and send it out to the company of their choosing so they get the finished product back in time for the final presentation.

The fabrication can also cause the project to be delayed if they send back the board with any defects. During large batch manufacturing of printed circuit boards, many times connections and leads can be crossed or shorted unintentionally. This fault would require the team to order another board and may have limited time to do so, therefore they should leave time for a first and second draft of a PCB whether the first one works or not.

4.2.16 Mounting Constraint

After the users have the system with them, one of the first things that they might think of doing is place it somewhere and set it up. Placing the item could be a real challenge for the user if they're not used to mounting things on the walls, because this can cause the system to be misaligned. This can affect the angle of the solar panel and due to this reason, it might not stand up to the expectations. This will lead to more usage of battery than the solar panel, which makes solar panel a useless device for this kind of setup.

4.2.17 Networking Constraint

As discussed earlier, every user might have a unique situation when we talk about smart devices. With people owning small homes/apartments, they’re able to interconnect devices wirelessly without any issues. But for large home/apartment owners, wireless signals tend to be weak in certain areas of the premises. This can be due to poor quality internet connection provided by the internet connection, or the user might not have a good router to get the right signal strength, or maybe just how their house was made. This could mean that the user will have to purchase a better internet connection, router, a wireless extender, etc., to make this system work. Without network connection to the mobile application, the user will lose the control over the system as they will not be notified about any decisions made by the system nor will the user be able to make any decisions for the system.

5. Hardware Design

Designing the hardware was done in stages. These stages are: power, microcontroller, sensors and pump. Each stage was done in that order to be able to be tested properly and confirm the design is working at each point and meets the requirements set forth earlier in the document. Schematics supplied are used in the final design of the PCB to connect each part correctly and allow software testing to be done on programs such as Multisim.

When we sat down initially to design our system, power was one of the main features on our minds. We knew that we would have to provide power to a barometric pressure sensor, temperature and humidity sensor, water level sensor, water pump, microcontroller, and Wi-Fi module. We wanted to be able to adequately power all of our components by combining a solar panel and a battery. We did not have the money to spend on a large battery or solar panel, so we consciously selected components that would keep the overall power consumption low and size of our device small. This allows it to become more portable and usable in many different situations. The combination of battery and solar panel allows the device to be used indoors, outdoors, and in shady conditions.

The microcontroller was just as important in our minds as the overall power consumption of our device. When researching which microcontroller fit our design the best, special care was taken to make sure it would be compatible with all of our analog and digital components all the while not drawing too much power. This was important because we could select all the best sensors but the power consumption of the microcontroller would always dwarf them so we needed to keep that in mind. Our goal is to integrate the microcontroller and sensors onto one PCB that interfaces with our water pump and mobile application.

5.1 Hardware Design Flow Overview

This section provides a quick walkthrough of our hardware diagram found in section 2.5. For a more detailed explanation on each part and how much it cost us, please refer to sections 3.2, 3.3, and 3.4 of this document.

Solar Panel and Battery
These two devices are intended to supply a steady 5V to the voltage regulator, that powers the sensors and the microcontroller, and all other components. The battery is included because it can hold a charge when there is no access to direct sunlight, keeping the integrity of our system intact. The battery is connected to a relay that helps operate the water pump, and a voltage regulator is connected between it and the microcontroller and sensor hub. The regulator keeps voltages at a constant level, preventing damage and unwanted brown or blackouts.

Water Pump
Controlled by a relay and the microcontroller, this pump is used to actually water the plants and is able to supply enough water given the smaller voltage of 5 volts.

Microcontroller
The microcontroller is the brains of our operation. It gets its power from the battery and voltage regulator, takes in serial data from all of the sensors, communicates through the Wi-Fi module, and regulates the pump that waters the plants.

Sensor Hub
This hub takes data from the environment and funnels it through the PCB to the microcontroller and later to the Wi-Fi module and our application.

Wi-Fi Module
The Wi-Fi module receives data from both the microcontroller and software application that we are developing and is our link between the two. It allows the user to monitor data from the sensors and send commands to the microcontroller.

5.2 Schematic Design

5.2.1 Power Schematic

The power of this design is taken from a 5 volts DC input from a micro-USB B type plug. The micro-USB plug is inserted into a female 5 pin on board receiver, the data pins are grounded since all we wish to use is the voltage input from the battery. The voltage and ground pins are the main supply and ground for the entire design. The 5 volts are used in two different ways, one being sent through a regulator to step down to 3.3 volts for the controller and sensor and the other being sent to a relay that takes care of powering the water pump.

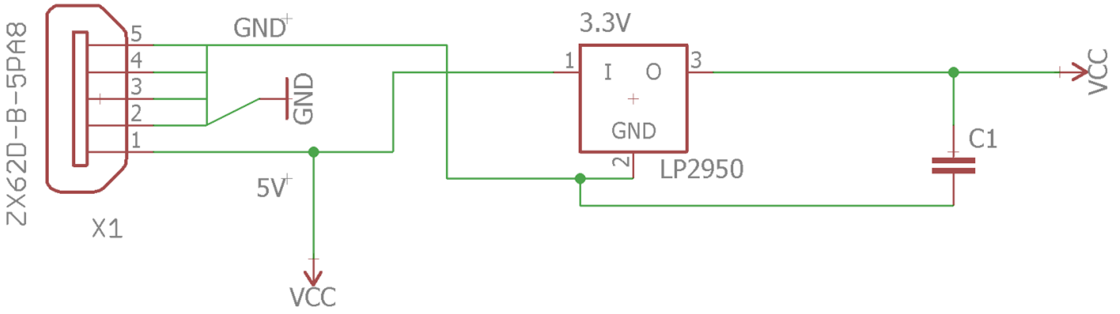


Figure 35: MicroUSB + 3.3-volt regulator + 5-volt junction schematic

What needs to be monitored the most in this section of design is both the voltage used by each component and the current draw from them. The voltage and current help determine the amount of power draw from each piece in the design. Limiting this allows our device to run.

5.2.2 Relay Schematic

Next is the design of the relay that relies off the 5 volts from the power supply and an input coming from a GPIO pin. The GPIO is able to send out a 3.3 volt pulse to turn on a BJT in order to saturate it and have current pass through it. This current and voltage than saturates the diode which allows current to pass to the relay switch and turn it on. We're able to use the 5 volts from the original input and not boost it because the water pump selected is able to operate at a 5 volt input.

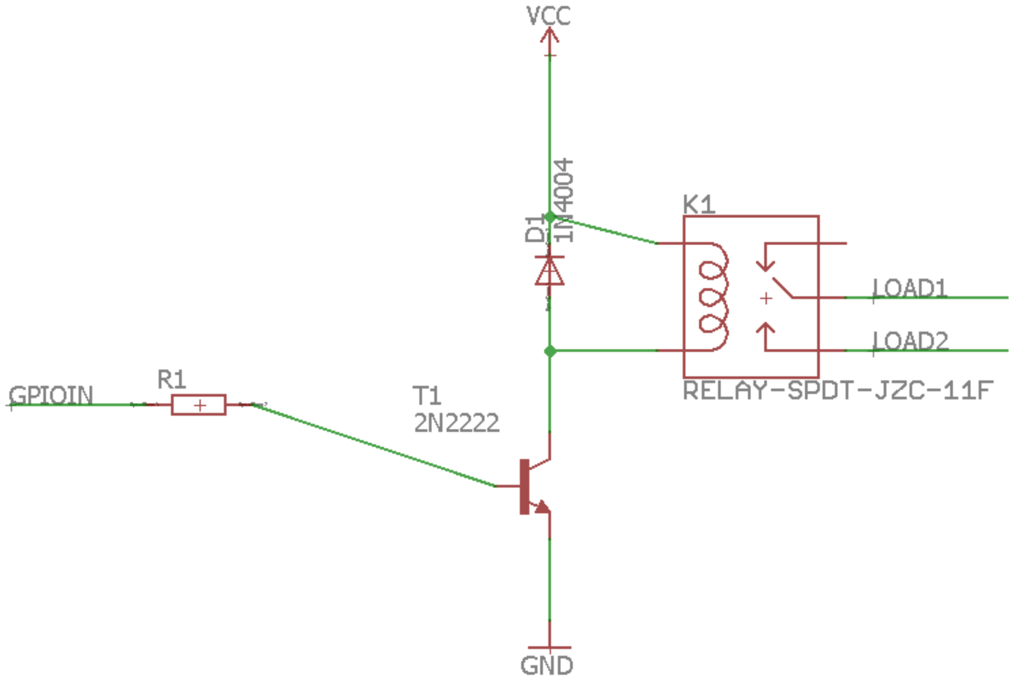


Figure 36: Relay schematic using 5V VCC + switching voltage from GPIOIN line

This is the circuit that is essential to our project. Without the relay switch working properly the design will fail. Proper testing of each component on this schematic is crucial and must be done thoroughly.

5.2.3 Microcontroller Schematic

The main brains of our design is the microcontroller unit. The ATmega328P didn't have a preinstalled library when Eagle was downloaded so, after searching the Eagle public forum, Sparkfun's schematic library for controllers was found and used. This setup is clear about power being added on the left side connections, while communication and GPIO ports are on the right.

Powering the MCU is the 3.3V regulated voltage from the power supply schematic mentioned above. The VCC must be connected to both the VCC and AVCC to the MCU, this allows us to use all the ports including the ADC ones through the power connections. These ADC ports are necessary for us to use because they allow us to capture the analog signals sent to us from our sensors and convert them to a digital signal that we can use and manipulate in the microcontroller and our application. The two ground ports are connected to the ground that is universal through the power supply schematic.

SV1 represents our temperature and humidity sensor. Port one has to be connected to VCC in order to turn the sensor on and have a resistor connected to the line that runs from port 2 to the MCU. Port 2 is connected to the PC1 of the MCU which represents one of the analog-to-digital converters. This helps convert the data from the real world sensor to the digital form the controller uses to pass information with. The resistor acts as a resistor divider to help realize what temperature it is and how humid it is. This works because the humidity and temperature are registered as resistance values and can be compared through the resistor divider in the ADC. Port 3 is a null port and therefore will not be connected to anything. Port 4 is a ground terminal that is connected to the universal ground port.

SV2 represents our barometric pressure sensor. Ports 1 and 4 are null ports and therefore are not connected to anything. Port 2 is our ground port and is connected to the universal ground port supplied from the power schematic. Port 3 is both our turn on and data pin, by using the voltage divider rule the analog to digital converter can register what the actual barometric pressure is. This works because the barometric pressure is registered as a resistance in the real world and has to be converted through a reference resistance.

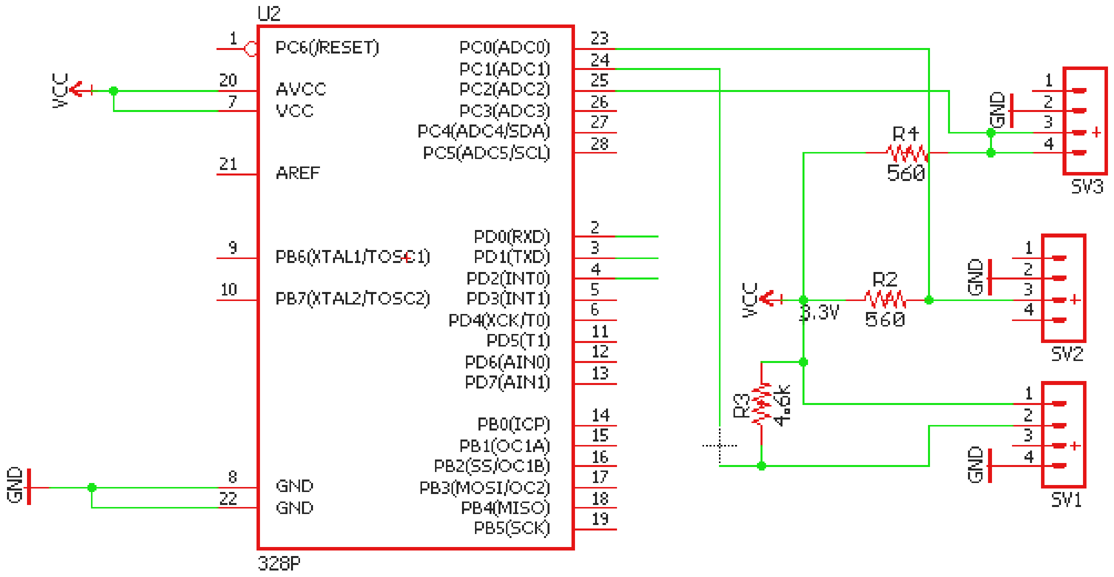


Figure 37: Microcontroller schematic

The Rx and Tx ports will also be used in a later schematic for communication through WiFi. These ports transmit the data and receive the data from WiFi in order to pass the information gathered by the sensors connected on board. These two terminals are the most important when it comes to interacting with our device via the application we designed along with it.

5.2.4 Wi-Fi Module Schematic

The microcontroller cannot communicate on its own, it requires a Wi-Fi module. The one provided with this schematic is the ESP module we used for developing our design. This will be used as a backup for the project if the team is unable to successfully design a full Wi-Fi module of their own.

The Wi-Fi module uses the RX and TX pins to communicate with the microcontroller and pass data between the two. The controller sends the sensor information to the Wi-Fi module that will then send the data to the user via the application. The Rx and Tx pins are working as a UART for the microcontroller.

The VCC is added onto the CH_PD to enable the module and VCC turns it on, these will be supplied from the 3.3 volt power given from the regulated source from the power schematic. The GND will be attached to the universal ground supplied from the power schematic.

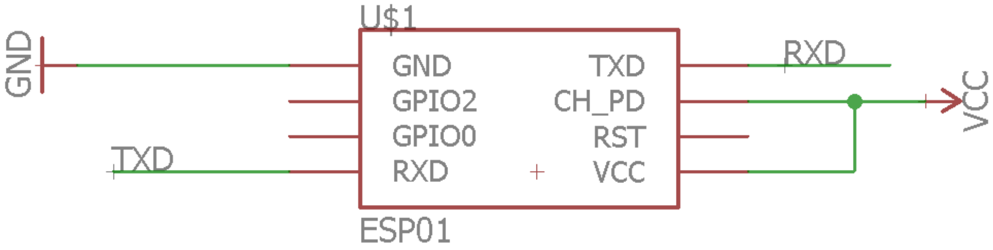


Figure 38: Wi-Fi module schematic

The four separate schematics combine together in order to work properly. These are the grounds of our breadboard design and the eventual PCB layout. Luckily the schematics were designed in Eagle and can be changed from schematics to PCB form easily. This narrows down the amount of work needed to be done later on in our design.

5.3 Case Design

The overall design of the case is extremely important. When working with electronics and water is imperative to use extreme caution so as the two are not able to come in contact with each other. If, for any reason, the two would happen to mix together than the entire system has the potential to be destroyed and become unrepairable.

Using the table below as a reference we will do our best to get our water protection to an IP65 rating. The IP rating, or Ingress Protection rating, is an internationally recognized waterproofing standard used in most modern devices and cases. The two digits in IP65 represent different things, the first digit means it's able to resist dust and small foreign objects while the second digit is the proofing from liquids. With the second digit being a 5 we are after protection from water spray in all directions, this would take care of waterproofing against rain and condensation as well as a backup spray from the water pump due to a fault.

IP Rating for Liquids	Description
1	No protection
2	Protection against condensation
3	Protection against water droplets deflected up to 15 degrees from vertical
4	Protected against spray up to 60 degrees from vertical
5	Protection against water spray from all directions
6	Protection against low pressure water jets in all directions
7	Protection against strong water jets and waves
8	Protected against temporary immersion
9	Protected against prolonged effects of immersion under pressure.

Table 32: IP ratings for liquid safety

Source: http://www.enclosurecompany.com/contact.php

This being the case we took careful consideration to have the electronics box waterproofed. By using molded plastic and water tight seals, we would incase the electronics in a water-tight box. Another important measure was to separate the two as much as possible while still keeping the design portable and compact.

This was a tough design to complete, but with today's technology people can literally print the designs they wish to use in their projects. By using a 3D printer, we are able to sketch out our box design on a computer program such as AutoCAD and export it to the 3D printer where the sketch could be made in less than a day. By utilizing this advanced technology are able to modify our design when needed at almost any time.

Another need of this design was to have the solar panel positioned in the most optimal way. When you mount it directly on the device it would be facing 90 degrees up. This is not the most ideal angle however. As seen in the figure below the optimum angle for solar panels varies depending on location.

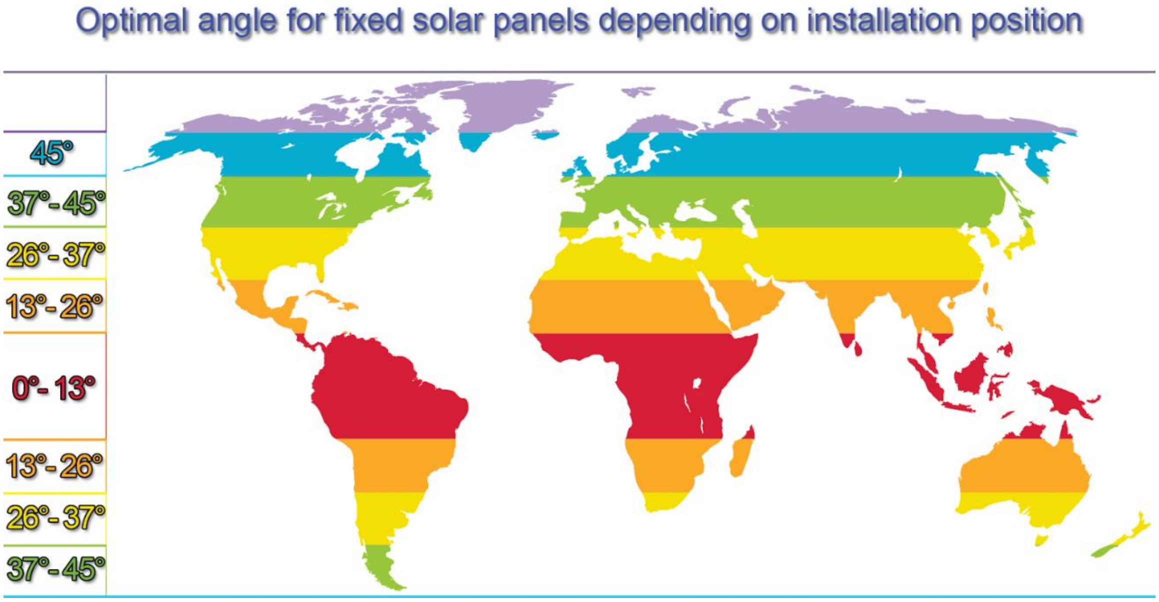

Optimal angle for fixed solar panels depending on installation position
45°
37°-45°
26°-37°
13°-26°
0°-13°
13°-26°
26°-37°
37°-45°

Figure 39: Best angle for solar panels around the world

Source: http://solarpanelsphotovoltaic.net/find-best-solar-panel-angle-tilt-angle/

To make sure that our design would get the best amount of sunlight, and since it is supposed to be a marketable product bought around the world, we needed to install a way for the solar panel to move around and adapt to its environment. By using a simple hinge, we can achieve this. All that is needed is a part that can be added onto the 3D printed electronics box which can easily be made with the 3D printer again.

Another necessity was the water tank we would use to draw from in order to water our plants. The main length has to come from the length of the water level sensor chosen for our design. By using a longer and thinner design we're able to achieve a compact and portable final project. The materials used have to be clear so that users can physically see how much water is left in the container. A piece also has to be designed so that the container can hang onto the mounting system used for the solar panel and electronics box, but have some distance between all three for safe measure. By using a minimum spacing of 1 inch between the components we can achieve a better and safer design for the electronics.

One of our goals was to be able to spray at least a 5-foot balcony in order to create a broad range in which we can keep our garden watered. Our minimum goal is to be able to water a single plant without a large spray as well. The best way to do this is to have a flexible point of spray, which includes the ability to change nozzles easily. We will find a proper fitting nozzle connection for our water pump that is essentially universal and able to function as a wide spray or as a regular spray with low power for a single plant.

The way that it draws water from the system is integrated in the design of the water tank, a hole is left open and fitted properly so the intake hose can be placed close to the bottom of the tank. The output hose will be fixed with a fastener to the 3D printed rig but will have the ability to be moved around and taken out of the fastening if the user chooses to do so.

At the end of the output, the hose will have the nozzle connection attached so the user can easily change which connection they would like to use.

5.4 PCB Design

The schematics we designed in section 5.2 were designed using the EAGLE software that was selected. The two main reasons EAGLE was selected was for its compatibility with fabrication companies and its ease of use. EAGLE comes preloaded with a numerous amount of functions that make the whole design process of a PCB significantly easier. By implementing the design schematics from the above section of 5.2, EAGLE is able to convert the schematics directly into a PCB. Each device schematic comes with its own PCB implementation and symbol. By using the simple convert button EAGLE is able to design a board automatically by using the PCB symbol and the connections created from the schematic.

The schematic is able to be netted, or routed, together using another function on EAGLE. The autoroute function is able to take those lines used from the schematics and finds the best way to connect them. The function knows not to have the lines cross in order to prevent shorting out the PCB entirely. You can also adjust the dimensions on the board, so we are able to achieve our small sized PCB easily.

The PCB ultimately uses a two-layer design that has a VCC and ground layer. This makes the connections easier to solder on and the layout easier as well. By using two layers we also save space on routing lines so we can make the PCB smaller and more compact for our electronics box. This ends up satisfying our goal of a small printed circuit board as mentioned in our objectives.

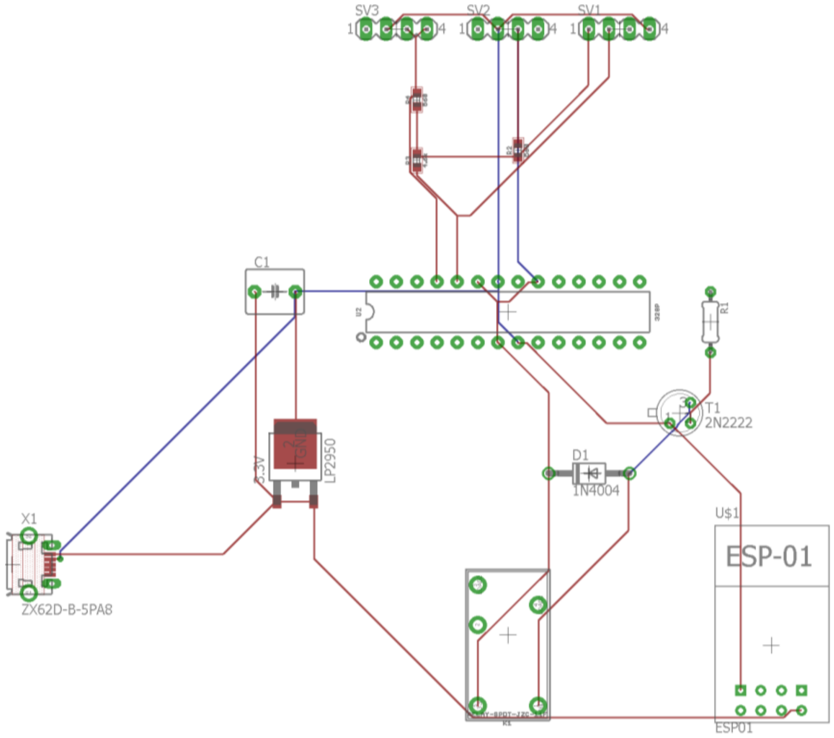


Figure 40: Representation of first PCB design in the EAGLE software

Originally, we had decided to go with PCBCart to make our devices, but after more research the group decided to go with OSH Park. OSH Park took a total of 22 days to return our PCBs to us. 5 boards came back with improper routing and drilling on the board. For example, the Microcontroller Unit on board had one of its pins completely missing from the drilling process done at the manufacturers. The 5 boards and shipping cost a total of $26.

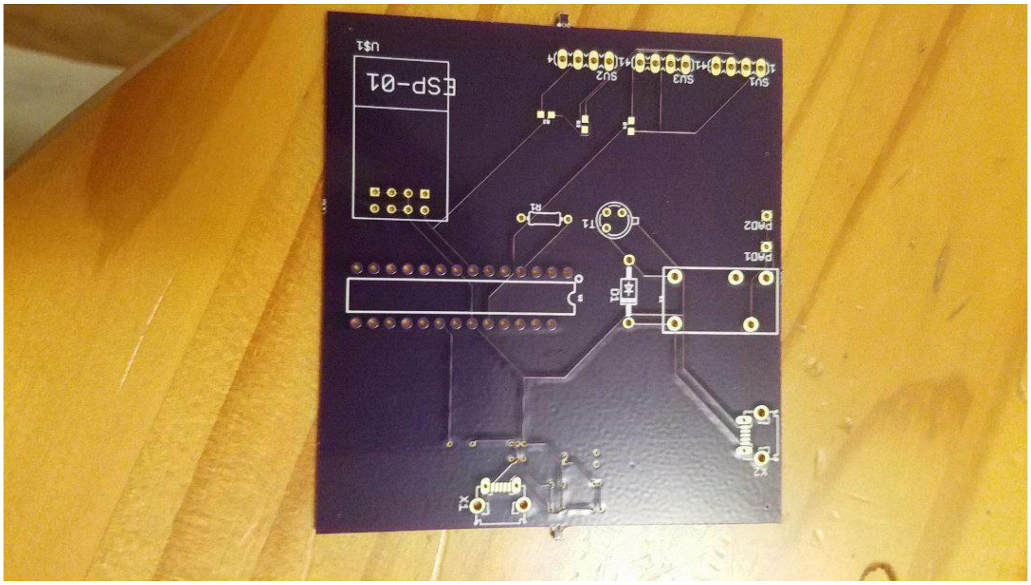


Figure 41: OSHPark PCB with defects

To correct this the design was then sent out to PCBWay which has been highly recommended for their fast design and shipment. The turn-around took a total of 2 days manufacturing and 3 days for shipping. The traces and board returned perfectly with a total of 5 boards for $26. This in turn raised our total amount of money spent on PCBs, but we did get a fully functioning and reliable circuit board.

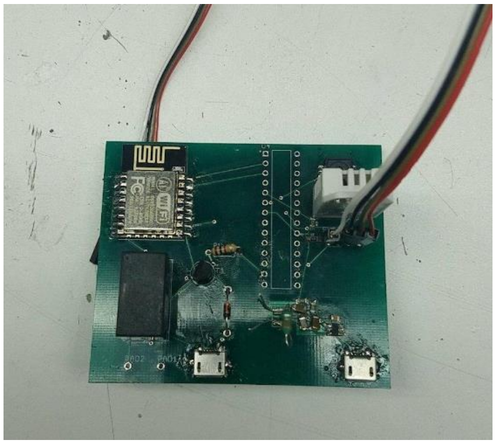


Figure 42: Final PCB design with some components added

From the OSHPark PCB we also learned a few mistakes that needed to be fixed in the design before we sent it out to PCBWay. We also received some feedback from the professor on new components that should be added to make the design a little more complex and interesting. One of those components was the voltage regulator. It was originally designed to be a standard switching regulator. After using some design tool resources, like T.I.'s Webench, the group settled on a Synchronous Step-Down Regulator. Texas Instruments TPS562200 will take in the 5V input and step it down to 3.3V using two resistors doing a simple voltage divider. Since this circuit is so efficient it is able to output the 1-amp input current from the battery. By doing this it will leave enough current for each device to work properly. The max amount of current draw from this system will range from 100 milliamps to 600 milliamps which means our circuit will be able to handle it completely. After initially testing, it was found that the input capacitors of 10 microfarads each would be blown due to a current flow back. By reducing these two capacitors it stabilized the circuit and made it possible to transfer the 5-volt, 1-amp input to a 3.3-volt 1-amp output. Below is figure of the final PCB design that went on the device.

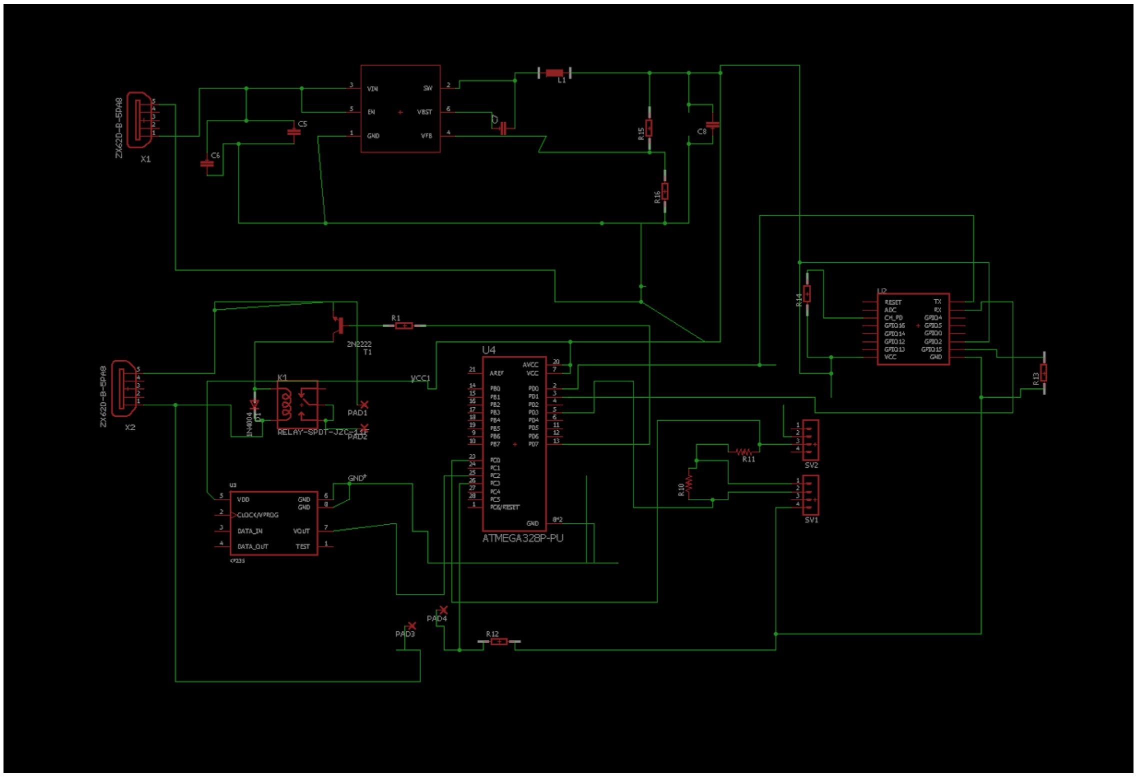


Figure 43: Final PCB schematic

6. Software Design

The software design of our project can be divided into four sections. The first section will discuss our design methodology and explain how we are focusing on developing the software. The next section will lay out the development tools we are using for the project. The third section consists of the software that is running within our microcontroller. This piece of code is in charge of continuously receiving data from the various sensors on the PCB and making decisions accordingly. The fourth section consists of the software that is running on the user's smartphone. From the smartphone application, users are able to read the data from the sensors and make changes to the device's settings.

6.1 Methodologies

While working on this project our team has utilized a few different design methodologies, using a few aspects of some of the more popular models instead of sticking strictly to specific model throughout our whole development phase. The models that we've borrowed the most from are the agile development model and the iterative and incremental build model.

The agile model has allowed us to adequately respond to change. Instead of planning out our entire system, requirements, and testing before beginning development, we were able to start with a rough idea and outline of requirements and use that to influence our research and early development. We were also able to easily incorporate suggestions given to us by our professor as we turned in drafts of our research, and we weren't hindered by changes to our requirements late in development. Using the agile framework also allowed us to break up the work for this project, with each person responsible for planning and executing most of the work for their section of the project. However, we were not cut off from each other completely. We regularly kept in contact using a group messaging service, so teammates always had up to date knowledge of what was being worked on by who. Any major design decisions were brought before the whole team and voted on, and we held bi-weekly meetings in person to talk about our overall progress on the project and discuss our plans for moving forward. The agile development model is also related to the iterative and incremental development model that we drew ideas from, they both prepared us for ever changing requirements and adaptive development.

The incremental model of development focuses on designing, implementing, and testing a product, all the while adding small features and components until the whole product overall is finished. We can tell that we are finished when the product satisfies all requirements set forth at the beginning of the project and any more that were added along the way. Although we are not following the incremental model exactly, we are not delivering any partially finished prototypes to customers, we are still adhering to the general principles, which are:

- Breaking the entire system down into smaller, more manageable chunks that will be pieced together one at a time.
- After building the base system, we prioritize the next components and features that will be added.
- Once we finish working on one section, we test it and add it to the overall system before moving on to the next section. This ensures that our project is working each step of the way before adding anything new to the equation.

We particularly liked implementing the iterative and incremental processes because each sensor and component was added individually and had testing done to make sure it worked before we moved on to the next component. This modularity made it relatively easy to identify and fix problems because it was usually the most recently added component causing issues, we did not have to dig through hundreds of lines of code to find some obscure compatibility issue.

6.2 Tools Used for Development

Our main means of coding and debugging our hardware will be through the Arduino programming language and the Arduino integrated development environment. Integrated development environments, or IDEs, are sort of all-in-one packages that theoretically contain everything someone needs to work on their code. At a basic level, an IDE offers the ability to organize and add code to your project. There are usually built in tools like compilers to help developers build their code, as well as debugging tools. IDEs try to provide their users with everything they need to accomplish their goals without having to install numerous programs. Many even offer advanced code completion or integration with version control systems like GitHub. The reason we chose to use the Arduino IDE is because it makes it easy to write code and upload it directly to the board. In addition to that, it also runs on Windows, Mac OS X, and Linux, meaning every single team member is able to have a version of the IDE on their computers and therefore be able to contribute to the programming of the microcontroller.

Since our smartphone application focuses on Android, we are using none other than the Android Studio IDE. This IDE provides one convenient location for us to code, build, and test our application. Although a few members of the development team have Android phones, Android Studio provides us all with a built in emulator, allowing us to choose a variety of phones to upload and run our code on, provided we have enough RAM, without having to risk damaging one of our own phones. Our decision to choose Android Studio is talked about more in section 6.4.

We used two other tools to stay up to date with each other and communicate. Google Docs made it possible to collaborate on our research and formal writing since it allows its users to see changes to a shared document as they happen. Facebook Messenger was used to set up meetings and suggest design ideas, as well as keep teammates in the loop on what was being developed and when.

6.3 Microcontroller Software

The code that is running within the microcontroller has various functionalities. The first step is to be able to properly read the incoming data from the various sensors. We first need to set up a serial connection with the sensors so that they are able to send data to the microcontroller unit for processing. After performing an initial setup, initializing the serial connection and any variables we need, we can jump into the main loop of the microcontroller, this is where we request data from the sensors on the board. Luckily, we are able to use some existing Arduino functions, such as analogRead(), to help us grab this data from the sensors, as opposed to us having to create our own communication protocol from scratch. The analog to digital converter also aids us greatly in helping to grab data from our analog components. We need to make sure the barometric pressure sensor and water level sensor are both connected to analog pins on the microcontroller and then they will have their analog signal converted to a digital one. From there, we are be able to take that digital data, manipulate it, and send it out to our application to display the results to the end user. The process is largely the same with the rest of our digital components, although there will be less conversion and overall processing to do before the data is ready. At this point, we are almost ready to let the rest of the code take over and deliver this data to our application, however, we want to make sure we are getting accurate readings. If we were to take just one sample to grab data from one of our sensors, it could be tainted by noise from the rest of the circuit or other uncontrollable factors. Since we want to get accurate readings, we will be taking five samples from our sensors and taking an average of all the readings. We arrived at five samples, with small delays between them, after reading through some of the datasheets for our sensors. We believe this will smooth over any random noise or signals that may be present during any one reading of the sensor, and the cost in terms of time and power is not very high.

Once we have the data from the sensors then it is time to put the data to use and let the code make decisions. One of the main parts of the microcontroller code is the algorithm that takes care of the “Automatic Watering Mode” of the device. Although we are still working on the fine details, the main idea is that the algorithm will be able to automatically determine when would be the best time to water the plants based on the data it is receiving from the sensors. We are currently doing research to determine what values from each sensor are the deciding factors when it comes time to water the plants or not.

Another main feature of our device is its ability to receive input from the user for the “Manual Watering Mode”. Through wireless communication in the form of Wi-Fi, the smartphone application is able to communicate with the microcontroller using the Wi-Fi module. The Wi-Fi module sends signals to the microcontroller based on the data it receives from the smartphone application. These signals are processed and interpreted in order to carry out the desired actions from the user. These actions include watering the plants based on temperature, time, humidity and many more things.

Lastly, our code keeps track of all the sensors and make sure they are working properly. This means that if any sensor stops sending data to the microcontroller, a signal is sent to the Wi-Fi module which will then notify the smartphone application. The battery and solar panel are periodically checked to make sure the battery is working as intended and that the solar panel is still charging it. The microcontroller is also be able to enter low-power mode during periods of inactivity to preserve battery.

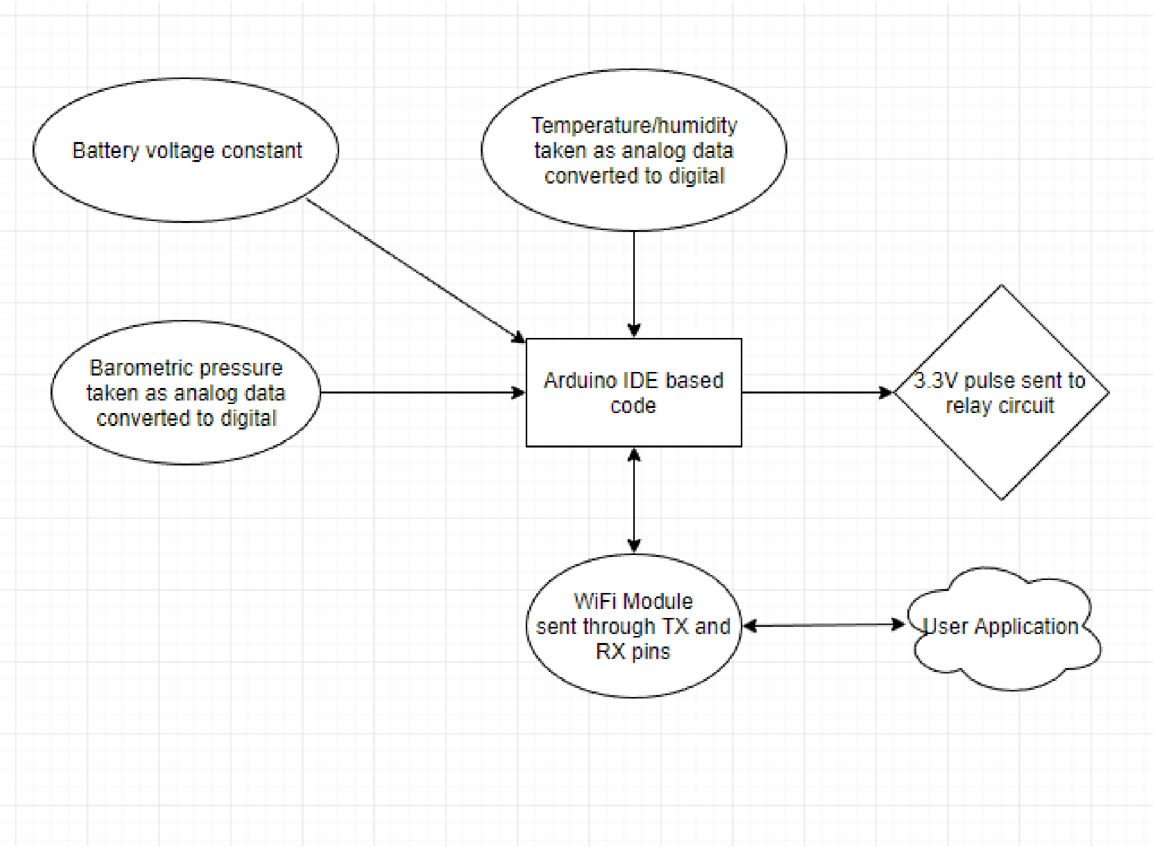

Battery voltage constant
Temperature/humidity taken as analog data converted to digital
Barometric pressure taken as analog data converted to digital
Arduino IDE based code
3.3V pulse sent to relay circuit
WiFi Module sent through TX and RX pins
User Application

Figure 44: Microcontroller sending/receiving of data

Unlike some of our other sensors, temp/humidity one does not need to use the ADC of our microcontroller to give us usable data.

6.4 Smartphone Software

The smartphone platform of choice for this project is Android, with the Integrated Development Environment (IDE) of choice being none other than the official IDE for Android App development, Android Studios (Figure 15). The reason for this choice is because Android is a very flexible and powerful platform with an immense amount of tools, which have everything we need to create our product. Apple's iOS platform is also very popular but it requires the use of an Apple computer and XCode in order to write any code for its platform, a device only one out of the four team members owns. This constraint is one of the main reasons why the smartphone application will first be developed and tested on the Android platform, and then may eventually have an iOS version if time permits.

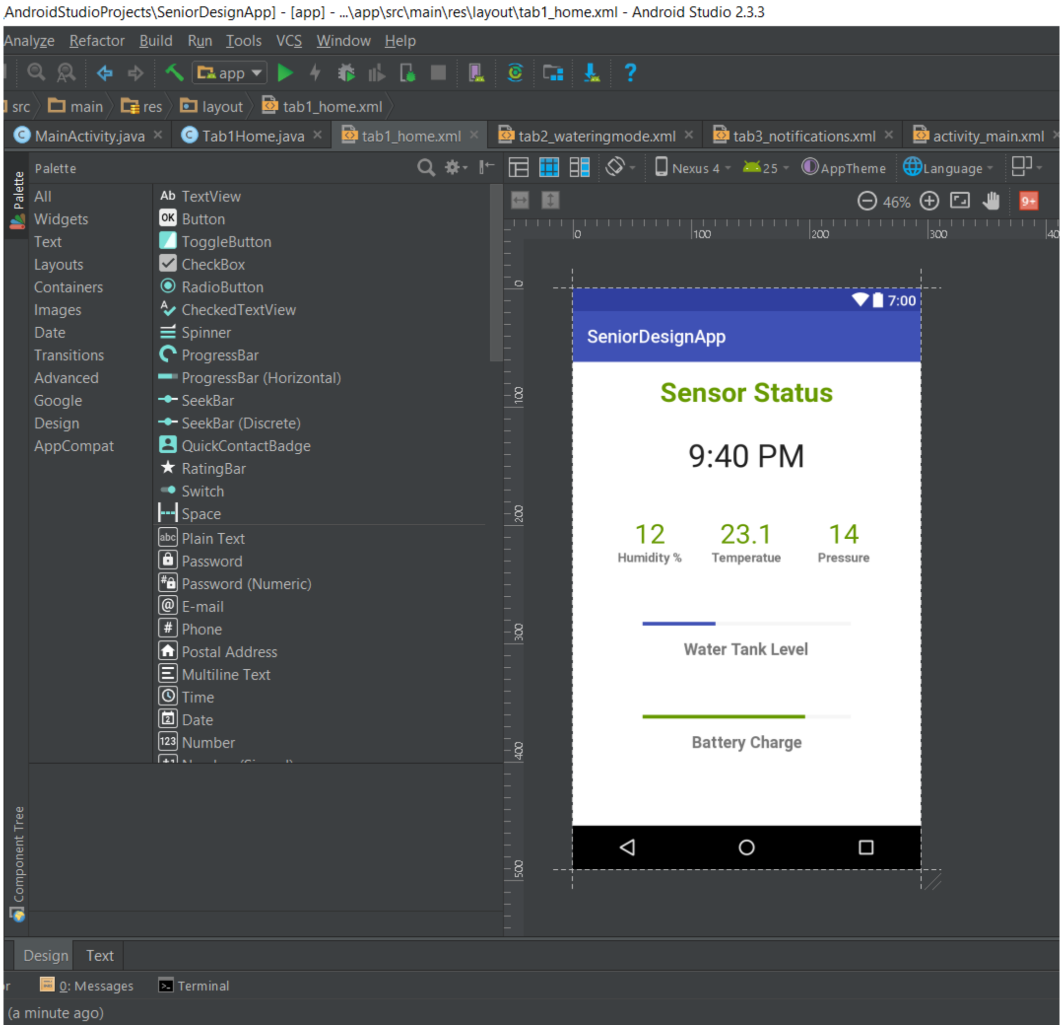


Figure 45: Android Studios IDE

6.4.1 Design and Functionality

The design of the user-interface (UI) for the Android application first followed a tab system with different views. The tabs were placed at the top of the screen and remained visible as long as the application was open so that users can switch between different screens easily. This layout is not only simplistic, but also widely used. Having a layout most users have already seen before makes the application very easy to use. The application revolves around three main tabs: Home, Watering Mode and Notifications. Each tab was represented by an icon/title at the top of the screen. Below you will find the two prototypes for the user interface. The first prototype follows the tab system mentioned above, while the second prototype came after testing when we realized it was much easier to test and develop using a single-screen layout. Accessing the different features this way made it really fast and simple, as well as editing the UI.

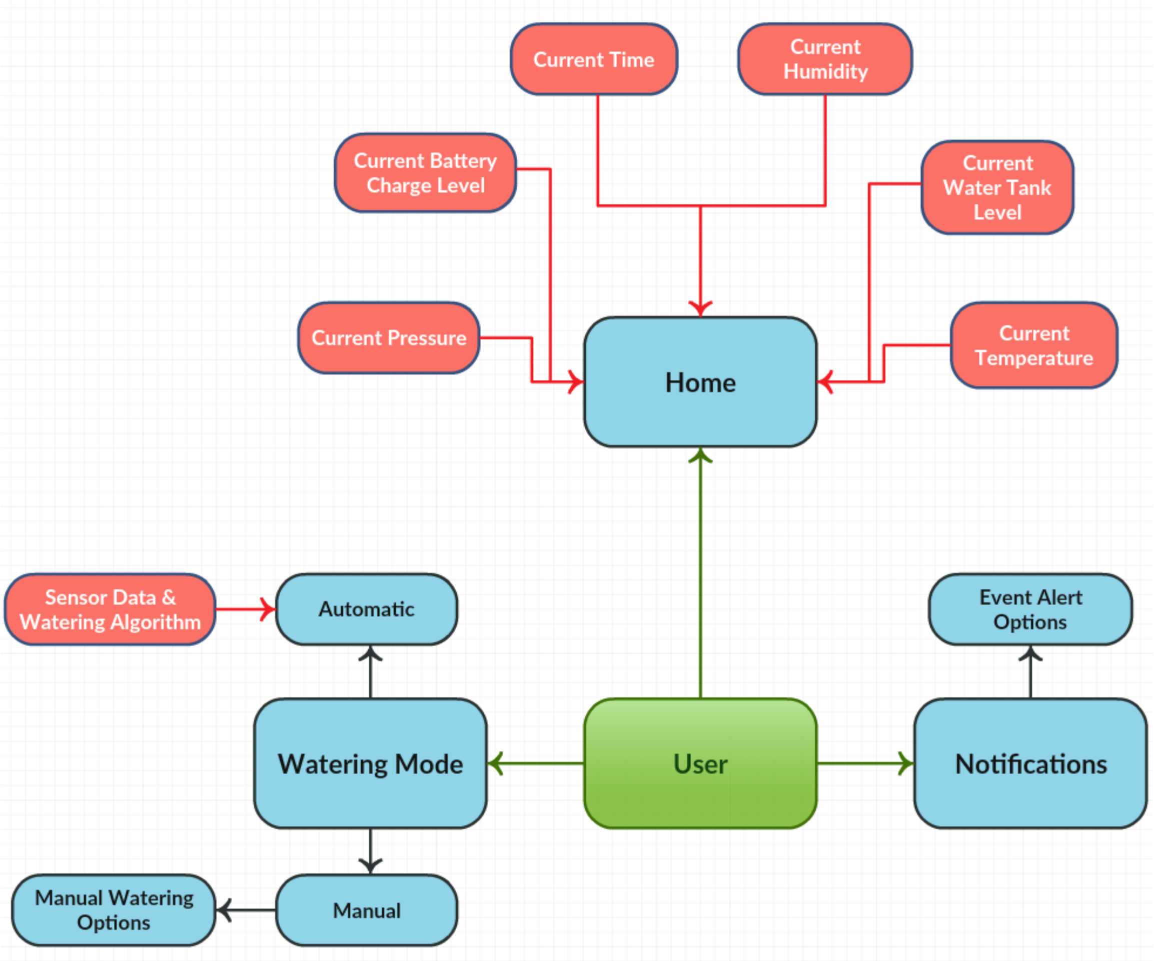


Figure 46: Smartphone Application UI Diagram

When the application is first opened, it sends the user directly to the “Home” tab/section, which is the default. From this screen, users are able to see all the relevant information from the sensors, which includes the temperature of where the device is located, the humidity, the amount of water left in the water tank, the charge level of the battery, among other things. This information gives the user a quick glance at the state of the system and its surroundings. Since our device communicates to the mobile application wirelessly, the user can be anywhere and still be able to check the state of the system without having to physically walk out to their balcony.

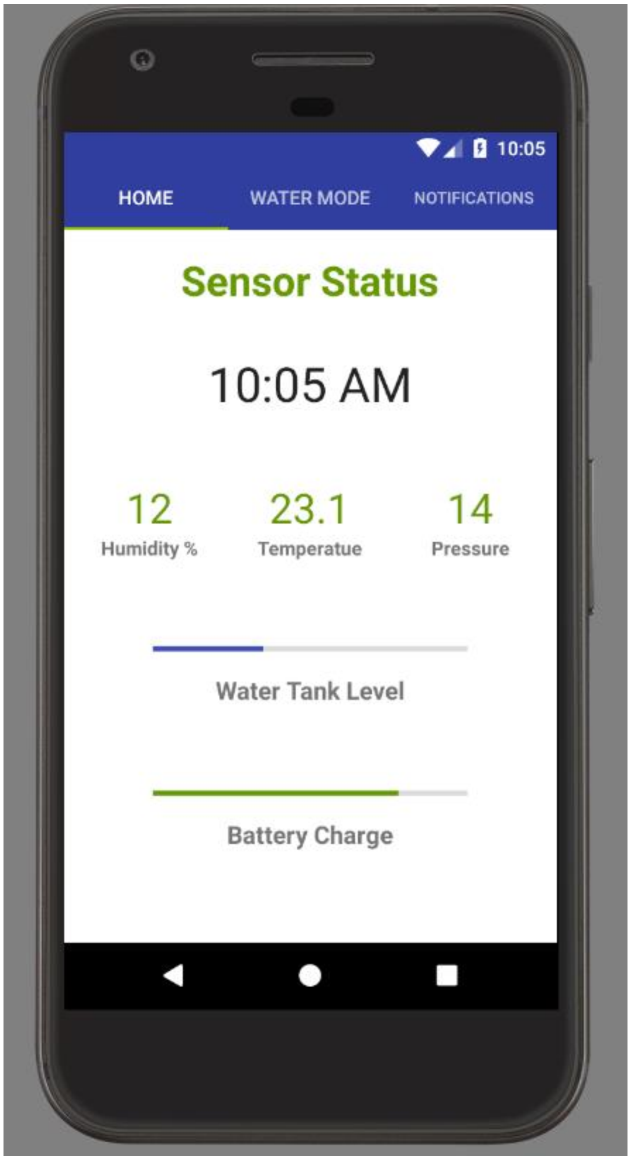
The second tab/section is the “Watering Mode” screen. From this tab/section, users have the ability to choose between the automatic watering mode or a manual watering mode based on their preferences. The automatic watering mode makes use of the sensors in the device to determine when is the best time to water the plants based on our algorithm. The “Watering Mode” screen consists of a button to enable/disable the automatic watering mode, followed by a list of options for the manual watering mode when the automatic one is disabled. Once a choice is made, the other option is grayed out (to indicate that it cannot be accessed) until the current option is turned off. If the automatic watering mode is disabled, the user is able to choose from a list what option they want to go with for the manual watering mode. For example, they will have the option to either water the plants at a set time every day, or once every certain number of hours, or maybe even just water the plants once the temperature drops to a certain level, just to name a few. This way, if the user feels like our automatic setting is not the most efficient, they will have other options.

Lastly, we have the “Notifications” screen. From this tab/section, users have the ability to setup their notification preferences for the device. This screen consists of a list of events that the user would like to get notified about. Each event has a toggle next to it which the user can use to enable or disable the notification for that event. The top of the screen has a title that says: “Notify me when…” and is followed by the list of events under it. Some of the events will be: “the water level drops below 10%”, “the outside temperature is above 95 degrees Fahrenheit”, “the battery is below 20%”, etc. With this system, if any of those events happen on the device, the user is notified on their smartphones through the app. For example, if the user has the water level event enabled, they will get a notification as soon as the water level reaches 10%, allowing them to go add more water before it completely runs out. Figure 13 above is a simple block diagram of the smartphone application which gives us a visual representation of the options within the application.

6.4.2 User-Interface Prototype v1

While the user-interface for the smartphone application went through many changes before its final release in December, it had to start somewhere. In the following pages, we will be showcase the very first prototype of the smartphone application which goes over the basics of the application and some key design components that made it all the way to the final release. All of the design and prototyping has been done on the Android Studios IDE using mostly HTML-style code to construct the user interface. An android device emulator was used to test the code and see how it would behave in an actual device. The emulator device is a Google Nexus Pixel running Android version 7.1.1.

First, we will go over the main screen which is the Home page that every user is welcomed with once the application is started. From here, users can take a quick glance at all the available sensor data and see the current state of the system. Secondly, we will go over the Watering Mode screen which will allow the user to choose between a fully automatic watering mode or a manual mode based on their preferences. Lastly, we will cover the Notifications screen which allows the user to set up notifications for individual events based on their preferences.



Sensor Status

10:05 AM

12	23.1	14
Humidity %	Temperatue	Pressure

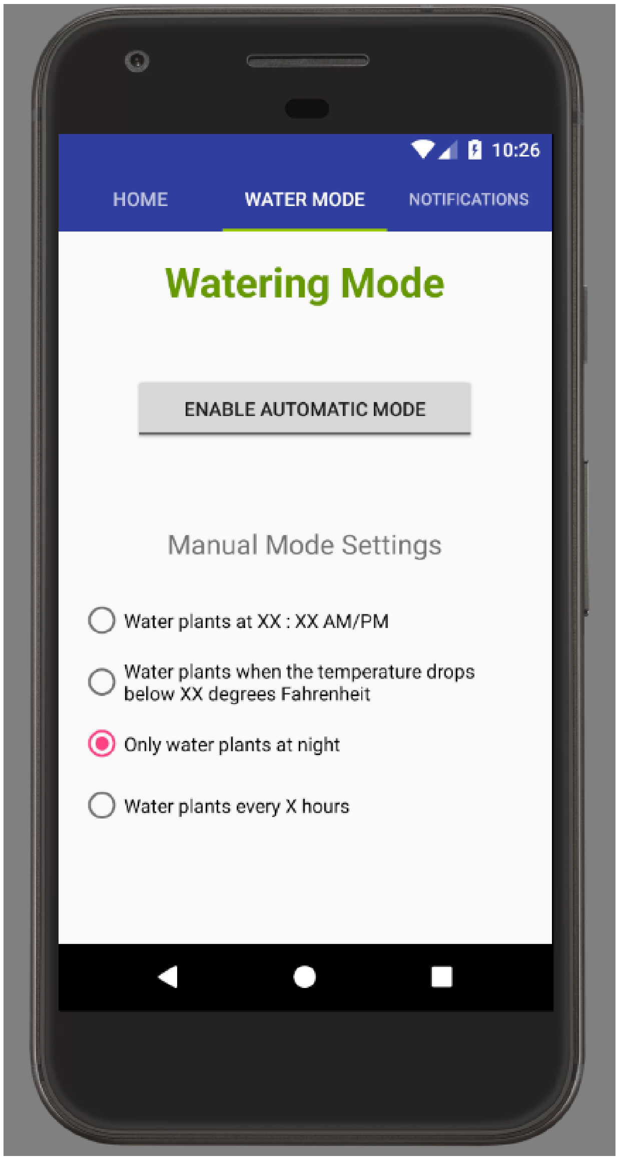
Water Tank Level

Battery Charge

Figure 47: Application Home screen v1

The Home screen is the very first screen the user is faced with once the application is launched from the application drawer. As seen in Figure 34, even on its initial stages, it already features a very clean and easy to read design. Starting at the top, we have the title of the "page" which tells the user what exactly they are looking at. Underneath the title, we have the current time in a nice large font which makes it more accessible than the small clock available in the notification bar up top. The time is followed by numerical data from three of the sensors. On the left, we have the data from the humidity sensor which tells us the percentage of humidity currently surrounding the system. In the middle, we have the current temperature (in degrees Fahrenheit) surrounding the system. On the right side, we have the final numerical sensor which is the pressure sensor. Like the other two, this sensor shows the current pressure surrounding the system.

Below the numerical data, we find the data for the water tank water level and the system's battery current charge. This data is showcased on a progress-bar style figure that starts off at 100% and gradually works its way down to 0% as the battery charge drains and the water in the tank starts running out. When each of the progress bars gets to a critical level, they change colors to red to symbolize a warning, letting the user know that it might be time to take action. In this case, a notification will also be sent to the user if they have decided that is a notification they want to receive.



Watering Mode

Manual Mode Settings

○ Water plants at XX : XX AM/PM

○ Water plants when the temperature drops below XX degrees Fahrenheit

● Only water plants at night

○ Water plants every X hours

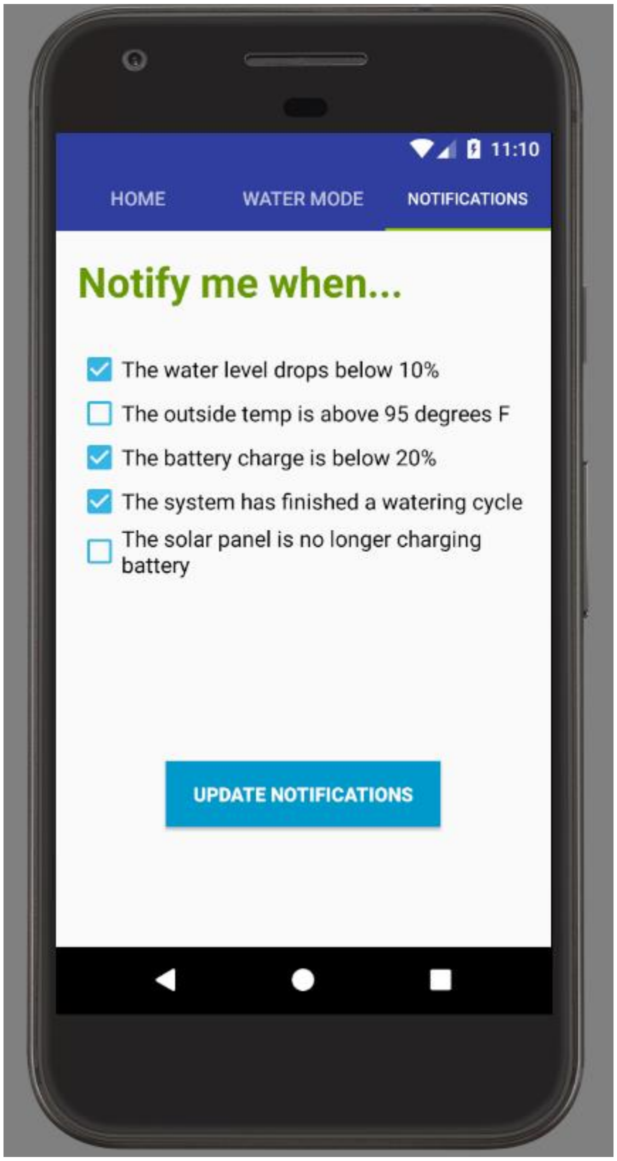
Figure 48: Application Watering Mode screen v1

Navigating through the various tabs or screens in the application is very easy and can be done in one of the two ways. The first way of changing screens is by tapping on the title of the tab/screen you wish to go to and it will take you there. This way you could jump from the Home screen all the way to the Notifications screen in one click. The current screen is noted by a brighter text color and a subtle green line underneath the tab title, as seen in

Figure 35. The second way of navigating through the screens is by simply scrolling from left-to-right or right-to-left depending on which screen you are currently on. This makes it easier for people with bigger phones in which reaching the top tab-bar can be a hassle. However, this way you can only go over one screen at a time. For example, by tapping the titles on the tab-bar, you are able to jump from one screen to another without any problems. This means that going from Home to Notifications is only one click away. With the scrolling mode, you would have to scroll twice in order to go from the Home screen to the Notifications screen because it would first go to the Watering Mode screen on the first scroll and then end up at the Notifications screen after the second scroll.

The Watering Mode screen is probably the one that is going to change the most as time goes on, at least in terms of looks. However, for right now, even though it is a little plain, it still has all the features we need. At the top, we have the page title, followed by a button. This button is the one that enables and disables the automatic watering system which uses our algorithm to predict when will be the best time to water the plants throughout the day. When the automatic watering system is disabled, the button says, “Enable Automatic Mode”. Once pressed, the bottom of the button changes color to signify that it is in the ON position and the text changes to “Disable Automatic Mode”. When the automatic mode is enabled, the manual mode settings are disabled and cannot be changed.

In the figure above, the automatic watering mode is disabled, which means the manual mode is enabled. When in the manual watering mode, the user is faced with a list of options from which they can choose one. Their selection will determine when the system waters the plants. Right now, the only options are to either water the plants at a certain time each day, at a certain time interval (every X number of hours), only at night time, or when the outside temperature drops below a certain number. As seen in Figure 35, right now every option has a place holder in the form of XX. Later on, this will be changed to a text field in which the user will be able to enter their desired input, as long as it is within the available range.



Notify me when...

☑ The water level drops below 10%
□ The outside temp is above 95 degrees F
☑ The battery charge is below 20%
☑ The system has finished a watering cycle
□ The solar panel is no longer charging battery

Figure 49: Application Notifications screen v1

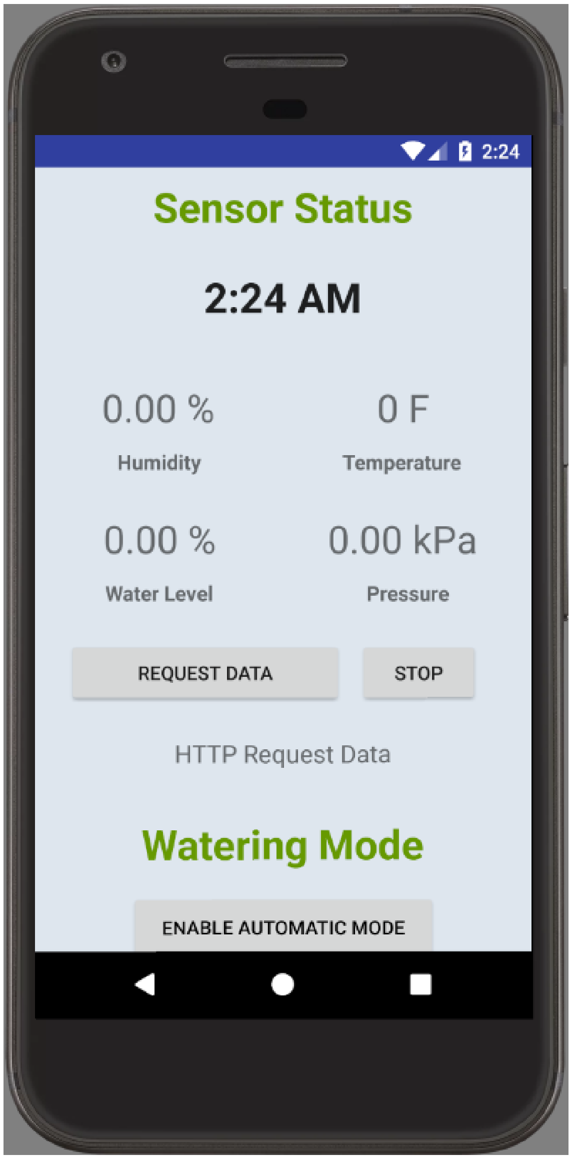
The Notifications screen begins with the title "Notify me when…" at the top, just like the other screens. Underneath the title we have a series of checkboxes which allow the users to make multiple selections. At the bottom of the screen we have a button that says, "Update Notifications", which will update the notification settings upon being clicked. This screen follows a very straight forward design scheme that should be easy to pick up by anyone. Users will first select all the events they are interested in and then hit the "Update Notifications" buttons at the bottom of the screen so that the system can save the new settings and only notify the user when the newly selected events occur.

Much like the Watering Mode screen, the Notifications screen will probably also be going through a lot of changes before its final version. Currently, we only have 5 events being displayed to give an idea of what the application will look like but later down the road more events will be added. Also, as seen in Figure 36, for this first prototype we hard-coded some pre-set values for every event. The end goal is to have the users be able to select the values in each of the events, much like the manual settings in the Watering Mode screen. For example, instead of only being notified when the water level in the water tank drops below 10%, the user will have the option to change that value to whatever they desire and then be notified based on their newly selected value. This allows the user to personalize the system to meet their individual needs.

6.4.3 User-Interface Prototype v2

While the app looked great in Senior Design 1, during testing in Senior Design 2 we realized that having multiple tabs/screens was an issue. Having to switch between tabs while checking for sensor data and testing the different buttons made us waste a lot of time at the beginning. We decided to switch the tab layout to a simpler single-page scrolling layout with different sections. This way, all the data was in a single screen and testing was made a breeze since we had everything just a simple scroll away. Debugging and developing was also made a lot simpler since we only needed one java file and one xml instead of three of each.

In the figure below, we can see the new Home screen. All of the relevant data from prototype v1 is still available, however we have also added two new buttons and another Textview. The first button is “Request Data” and its purpose is to allow us to receive data on demand for debugging and testing purposes. It also acts as an activation switch for the automatic timed data retrieval were the app automatically updates the data on the screen every 2 seconds to make sure everything is working properly. The “Stop” button next to it does exactly that, stop the transmission of automatic sensor data. It also acts a manual water pump power button from which we can turn the water pump on and off for testing. All of the feedback data from the two buttons gets returned to the user through the “HTTP Request Data” Textview right under the two buttons.



Sensor Status

2:24 AM

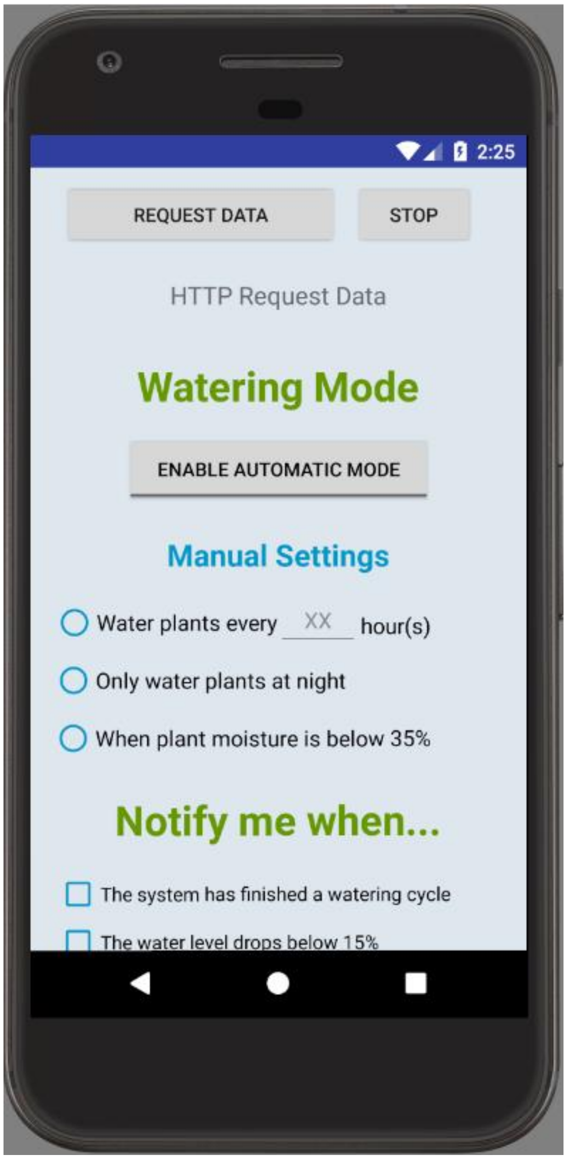
0.00 %	0 F
Humidity	Temperature
0.00 %	0.00 kPa
Water Level	Pressure

HTTP Request Data

Watering Mode

Figure 50: Application Home Section v2

Next is the Watering Mode section. This section did not change too much from the previous prototype. Manual setting options disappear when the automatic mode is enabled and then come back when it is disabled. As seen in the figure below, for the manual settings, users can not only select what watering mode they prefer, but they can also change the values of some of the modes. For example, on the first manual setting which waters the plants on timer when it is enabled, users have the option to enter their preferred number of hours they want the system to wait in between plant watering. The application recognizes user input and adjusts the timer accordingly. For testing, as well as demo purposes, the value entered gets interpreted at the number of seconds you want the system to wait instead of hours. We did this for obvious reasons. With only 10 minutes to demo, we can’t sit there and wait for the system to water every 1 hour. By changing it to seconds we are able to test it and showcase it much easier.



HTTP Request Data

Watering Mode

Manual Settings

○ Water plants every XX hour(s)

○ Only water plants at night

○ When plant moisture is below 35%

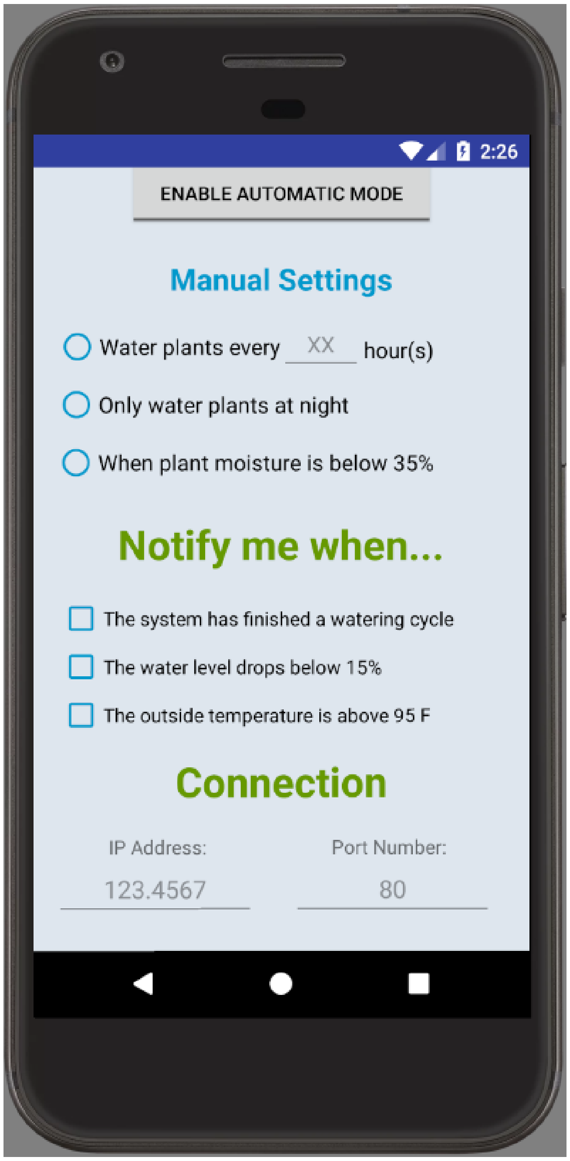
Notify me when...

□ The system has finished a watering cycle

□ The water level drops below 15%

Figure 51: Application Watering Mode Section v2

Lastly, we have the Notifications section. This section, much like the Watering Mode section, also did not change very much from the previous prototype. Only thing that was added was a new section under it in which we take care of the application communication with the WiFi module. This new section is called “Connection” and it contains two text boxes for the user to input the IP address and port number of the WiFi module once it connects to their home network. This is showcased in the figure below.



Manual Settings

○ Water plants every XX hour(s)

○ Only water plants at night

○ When plant moisture is below 35%

Notify me when...

□ The system has finished a watering cycle

□ The water level drops below 15%

□ The outside temperature is above 95 F

Connection

IP Address: 123.4567

Port Number: 80

Figure 52: Application Notification / Connection Section v2

Later on down the road, more conditions can be added under the manual watering mode settings and under the notifications. However, for the purpose of this project, we believe the current conditions are enough to showcase the capabilities of the system and how it works.

7. Testing & Demonstration Plan

The testing of the design is important to confirm the product is working properly and in the way the designers want it to operate. Without testing the design could fail and the team would not have any indication on where the faults may be in the system.

This section goes over:

- The stages and hardware that was tested in order to assure each part works properly.
- The software testing and debugging to insure the interface works with the hardware.
- How our team plans to demonstrate the entire system working together.

7.1 Hardware Testing

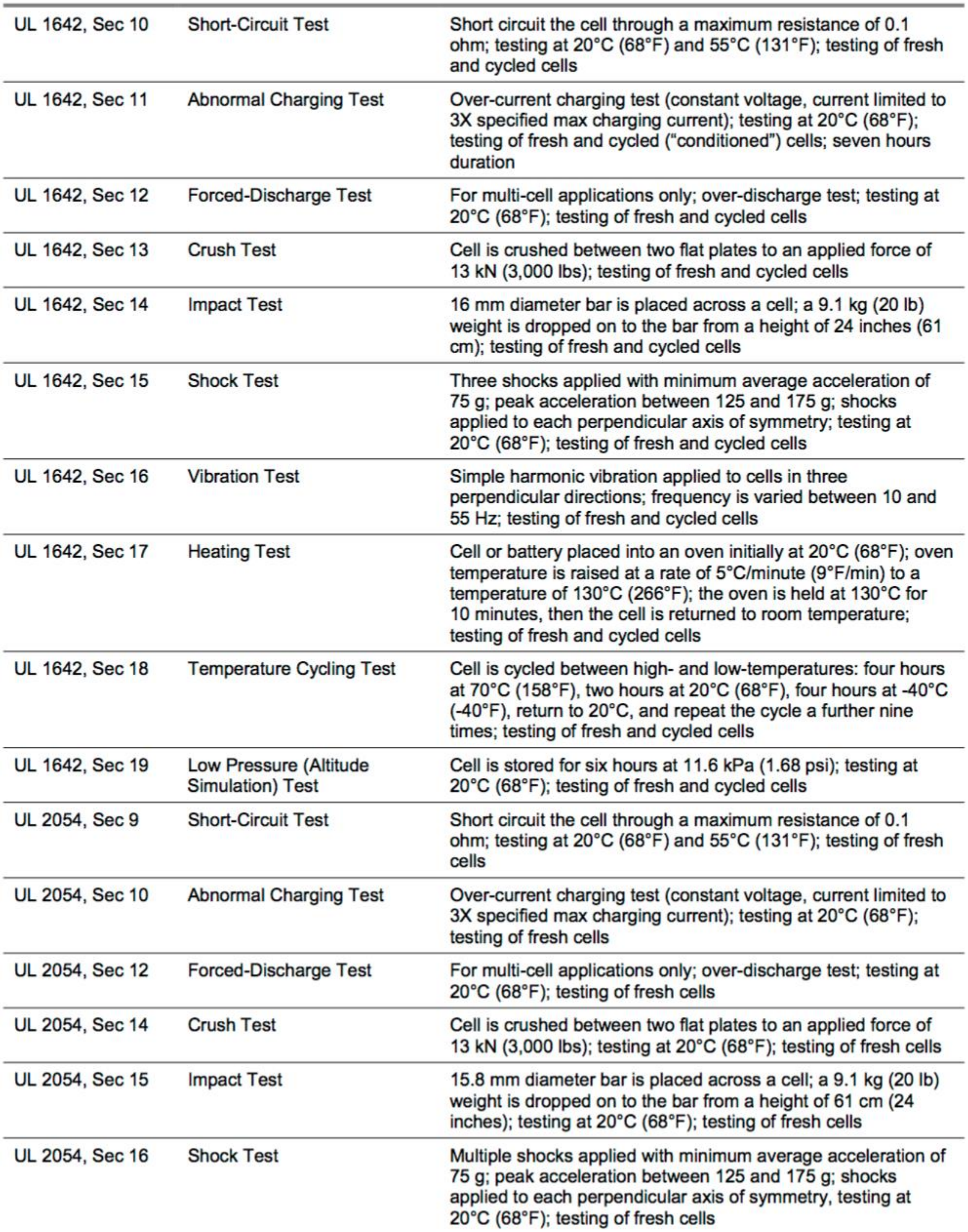
7.1.1 Solar Panel & Battery

The solar panel's voltage output can be tested by using a voltmeter. We can use the probes of the voltmeter and attach it to the wire coming out of the panel to detect the voltage produced by the solar panel by placing it in sunlight. The voltmeter should display approximately 6V for the solar panel to pass the test since we initially chose a panel that would generate at least 6V of peak voltage. If approximately 6V are displayed on the voltmeter, then the solar panel has passed the test.

In order to test the solar panel and V44 battery's charging capabilities, we can plug the solar panel DC output to the battery's DC input and check whether the battery is charging or not. If the battery is charging, the battery has passed the test.

To test whether the V44 battery's USB ports are functional and that they are capable of charging other devices, we can simply use a cable that connects to the battery's USB port and the phone, and check whether the phone is charging. If the phone is charging, then the battery has passed the test.

To test the charge indicator of the battery, after the battery has been charging other device(s) for a while, the charge indicator should change its color. If the color of the indicator changes, the charge indicator has passed the test.



UL 1642, Sec 10	Short-Circuit Test	Short circuit the cell through a maximum resistance of 0.1 ohm; testing at 20°C (68°F) and 55°C (131°F); testing of fresh and cycled cells
UL 1642, Sec 11	Abnormal Charging Test	Over-current charging test (constant voltage, current limited to 3X specified max charging current); testing at 20°C (68°F); testing of fresh and cycled ("conditioned") cells; seven hours duration
UL 1642, Sec 12	Forced-Discharge Test	For multi-cell applications only; over-discharge test; testing at 20°C (68°F); testing of fresh and cycled cells
UL 1642, Sec 13	Crush Test	Cell is crushed between two flat plates to an applied force of 13 kN (3,000 lbs); testing of fresh and cycled cells
UL 1642, Sec 14	Impact Test	16 mm diameter bar is placed across a cell; a 9.1 kg (20 lb) weight is dropped on to the bar from a height of 24 inches (61 cm); testing of fresh and cycled cells
UL 1642, Sec 15	Shock Test	Three shocks applied with minimum average acceleration of 75 g; peak acceleration between 125 and 175 g; shocks applied to each perpendicular axis of symmetry; testing at 20°C (68°F); testing of fresh and cycled cells
UL 1642, Sec 16	Vibration Test	Simple harmonic vibration applied to cells in three perpendicular directions; frequency is varied between 10 and 55 Hz; testing of fresh and cycled cells
UL 1642, Sec 17	Heating Test	Cell or battery placed into an oven initially at 20°C (68°F); oven temperature is raised at a rate of 5°C/minute (9°F/min) to a temperature of 130°C (266°F); the oven is held at 130°C for 10 minutes, then the cell is returned to room temperature; testing of fresh and cycled cells
UL 1642, Sec 18	Temperature Cycling Test	Cell is cycled between high- and low-temperatures: four hours at 70°C (158°F), two hours at 20°C (68°F), four hours at -40°C (-40°F), return to 20°C, and repeat the cycle a further nine times; testing of fresh and cycled cells
UL 1642, Sec 19	Low Pressure (Altitude Simulation) Test	Cell is stored for six hours at 11.6 kPa (1.68 psi); testing at 20°C (68°F); testing of fresh and cycled cells
UL 2054, Sec 9	Short-Circuit Test	Short circuit the cell through a maximum resistance of 0.1 ohm; testing at 20°C (68°F) and 55°C (131°F); testing of fresh cells
UL 2054, Sec 10	Abnormal Charging Test	Over-current charging test (constant voltage, current limited to 3X specified max charging current); testing at 20°C (68°F); testing of fresh cells
UL 2054, Sec 12	Forced-Discharge Test	For multi-cell applications only; over-discharge test; testing at 20°C (68°F); testing of fresh cells
UL 2054, Sec 14	Crush Test	Cell is crushed between two flat plates to an applied force of 13 kN (3,000 lbs); testing at 20°C (68°F); testing of fresh cells
UL 2054, Sec 15	Impact Test	15.8 mm diameter bar is placed across a cell; a 9.1 kg (20 lb) weight is dropped on to the bar from a height of 61 cm (24 inches); testing at 20°C (68°F); testing of fresh cells
UL 2054, Sec 16	Shock Test	Multiple shocks applied with minimum average acceleration of 75 g; peak acceleration between 125 and 175 g; shocks applied to each perpendicular axis of symmetry, testing at 20°C (68°F); testing of fresh cells

Figure 53: Series of UL Test

Source: http://www.prba.org/wp-content/uploads/Exponent_Report_for_NFPA_-_20111.pdf

Arduino Battery Tester

As the water sprinkler system is used overtime, it becomes necessary for the user to get an idea of how much longer their system can last without a charge. The goal here is to let the user check on their mobile application about the battery status, and also notify them if battery is critically low. In order to accomplish this, it is necessary to understand and implement the right electronics that goes behind this task.

To achieve this goal, we used the following components:

Component	Purpose
5.1V Zener Diode	5.1V Zener diode will allow the current to flow only in one direction until it reaches the breakdown voltage of 5.1V; limit of the diode. Once this limit is reached, the voltage flows in reverse direction. This will help protect the parts such as the Arduino Uno board and other modules as it will never cross the maximum voltage that each of these parts can handle without any damage.
2.2kΩ Resistor	The 2.2kΩ resistor's role is to match the current coming from the battery to the value that is compatible with the Arduino board; this again helps eliminating potential damage to the board.
100Ω Resistor (3x)	Each of the three 100Ω resistors will be connected to its own LED; on the positive sides. Again, to control the current flow coming to the LEDs so that they don't blow up.
Red LED	Represents critically low battery.
Yellow LED	Represents approximately half battery level.
Green LED	Represents fully charged battery.
Battery	Source to power the system and the component being tested to show the user charge status.
Arduino Uno	The board's job is to read the voltage difference of a battery and recognize how far it is from being charged and recognize the status of the battery; determine which LED should turn on.

Table 33: Arduino battery tester components

Below is a diagram describing the layout of this battery tester:

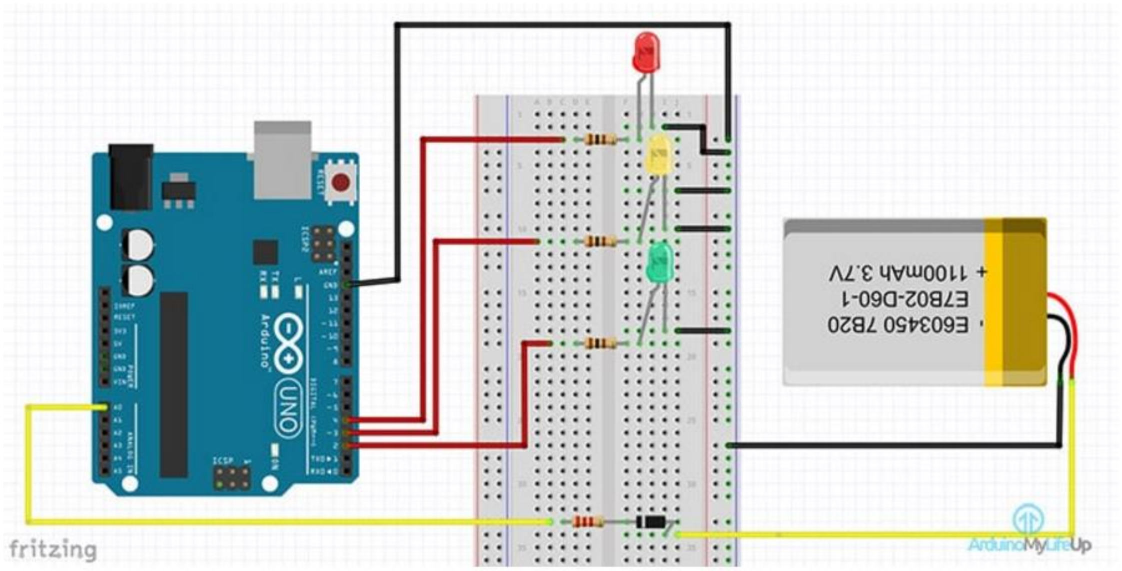


Figure 54: Battery tester layout

Source: https://arduinomylifeup.com/arduino-battery-tester/

The battery tester circuit was implemented like the one shown in the diagram above, then the Arduino Uno board was programmed using the Arduino IDE to recognize the connected pins and make the LEDs react based on the voltage difference between the calculated voltage values and original voltage values.

This battery tester circuit was tested by using different batteries each having different charge levels. A fully charged battery was used to make sure that the green LED turned on when it was connected to the output probe, a medium or a half-way charged battery was used to make sure that the yellow LED turned on, and finally a battery without any charge was connected to make sure that the red LED turned on when connected.

7.1.2 Microcontroller Testing

To ensure that our microcontroller is working properly the MCU was tested in a few different ways. The atmega328P is an Arduino board compatible controller. This was used to our advantage and test the initial chip with the board to drive an LED by having the user press a button attached to it. This ensures the chip arrives working properly and is able to handle input (from the button) and deliver an output (driving the LED).

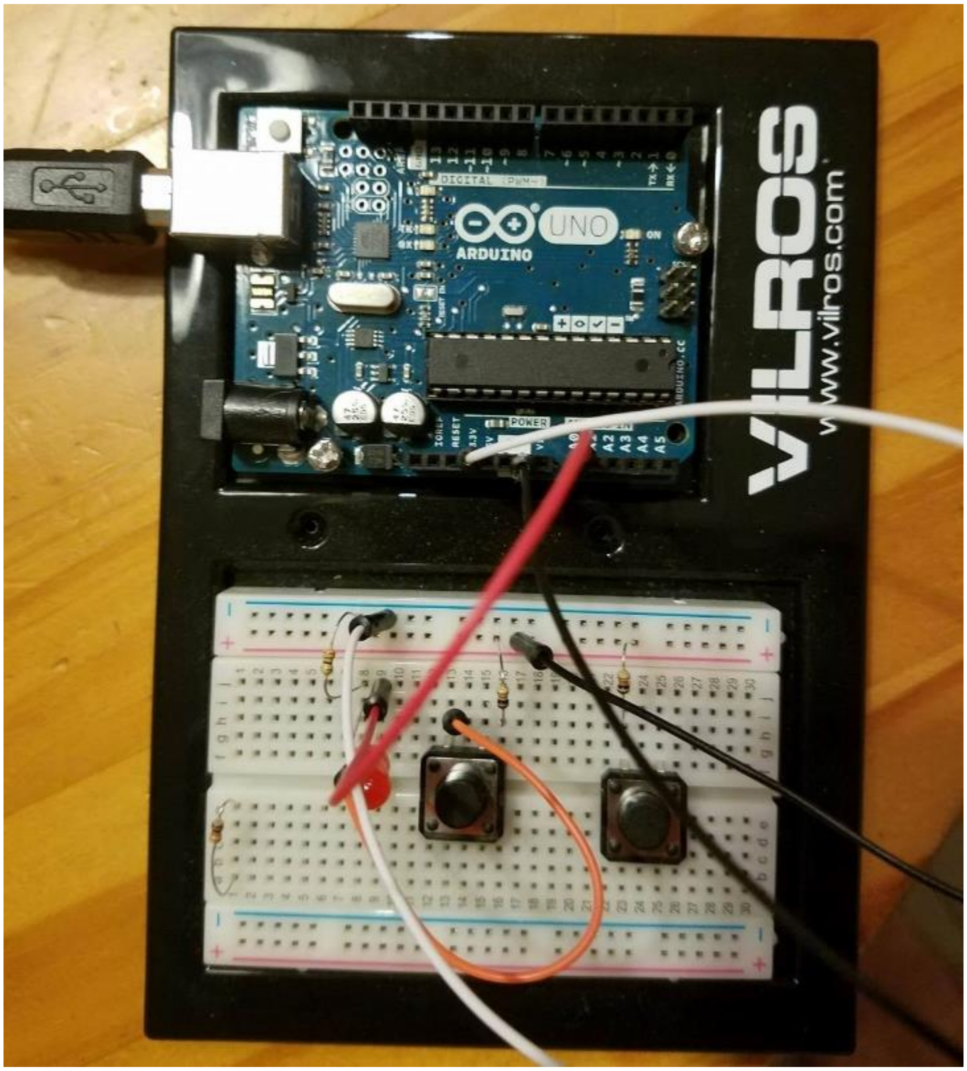


Figure 55: Microcontroller testing on dev-board with LED and push button

We then deliver the same test by driving the LED with a button but this time on a breadboard. This tells us if the microcontroller is able to operate off a pre-built board and can deliver the same tasks on standalone boards. By doing this it allows to make the next step and integrate the microcontroller onto the final PCB.

Analog to Digital Conversion

The analog to digital converter (ADC) on the ATmega is very important to a few of the sensors for this project since they output their signals in an analog format. In order for us to use these signals they need to pass through the ADC, so we want to make sure that it is converting values properly for us. The ADC features a 10-bit resolution and conversions to digital signals take about 13 - 260µs. This is well below the power up and response times for our sensors so we shouldn’t expect this to be a bottleneck during testing. In the figure below, it can be seen that we select our analog signal from the input multiplexer. The ADC then generates a 10-bit value based on the analog input voltage and send that to the ADC Data Register. We are able to initiate tests and read the approximated digital value of the sensors by sending just one instruction to the board: analogRead(). This returns a digital value for us between 0 and 1023.

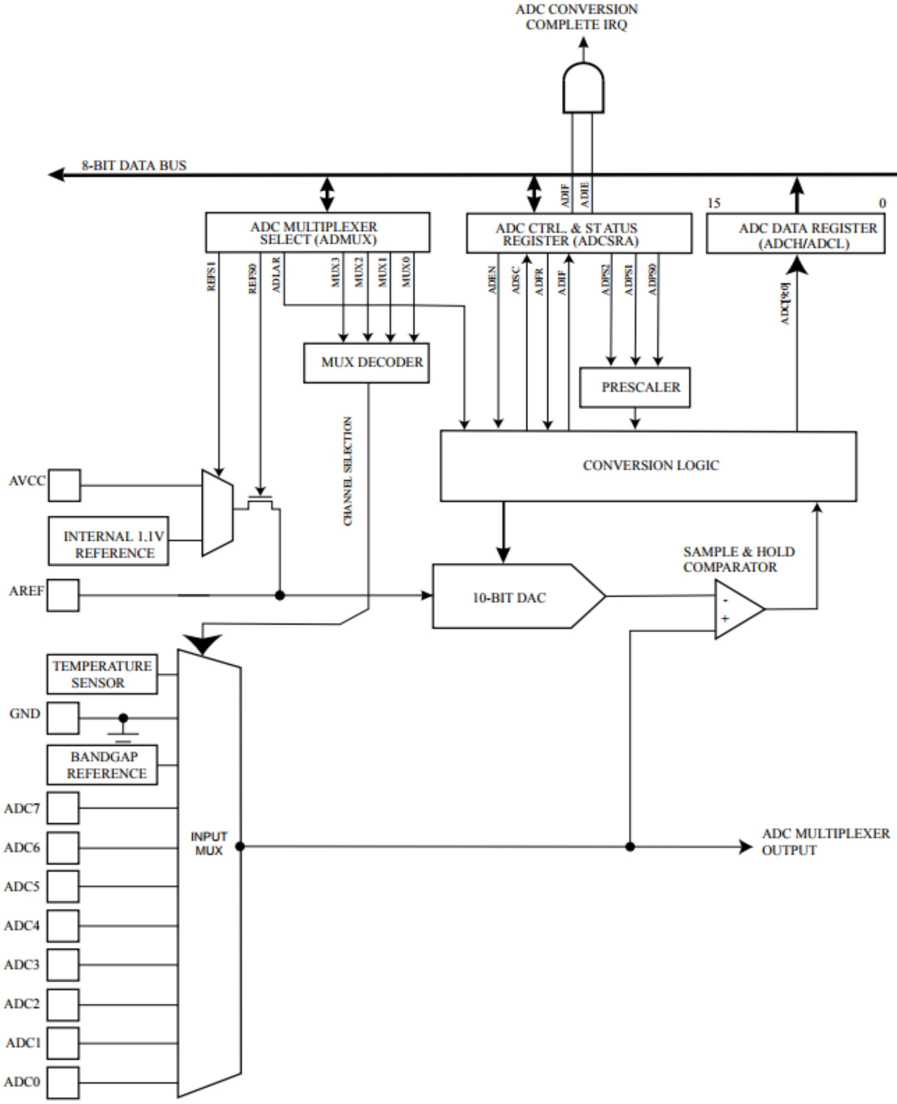


Figure 56: ADC on the Atmega

Source: http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

Testing Power Characteristics

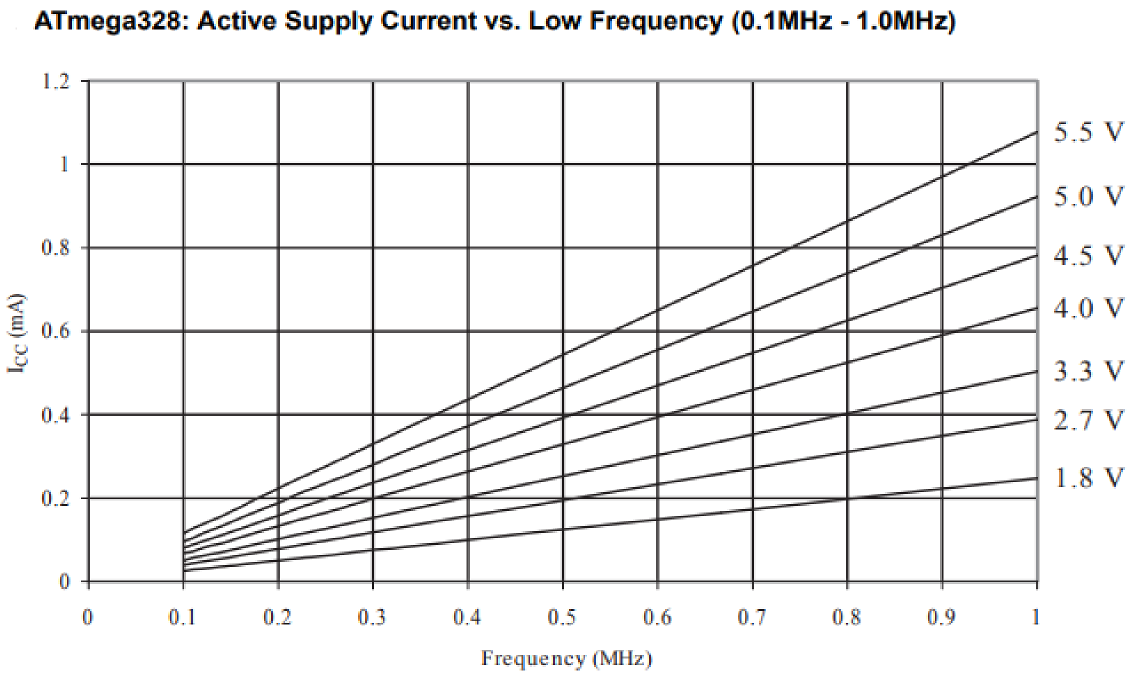


Figure 57: Atmega electrical characteristics

Source: http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

The figure above describes the typical operating characteristics of our microcontroller at low frequencies. We can see that the relationship between frequency and power drawn is approximately linear, and the datasheet shows that this trend continues on through the higher frequency ranges of 1MHz - 20MHz. It is important to know these characteristics because it paints a picture of how our microcontroller should behave during normal use and while we are stress testing our components. We won't be operating the microcontroller near its limits, so we shouldn't have any fear of damaging the microcontroller itself when we are running stress tests on our components. We are also able to verify using a multimeter how much current we should be drawing at given frequencies and voltages, so before we hook up any components to the microcontroller we tested its electrical characteristics. This usually isn't a problem with mass-produced devices like this, but sometimes faulty products can slip through the cracks, and this is an easy test to complete and we don't want to risk damage to any of the components we ordered.

Performance

We chose a few different metrics with which we measured performance of our system. Since power efficiency can be a very broad term, we try to break our performance measurements down into number of components used, accuracy, CPU efficiency, which itself can be broken down into processing time, code efficiency, and memory used. We want to take advantage of the accurate readings from our sensors so that we can provide the user with proper weather data while simultaneously making sure their plants stay watered. If we can achieve this goal without the battery discharging rapidly then it is a success. Now, just because we can achieve success by operating on battery for an extended portion of the day does not mean we can stop there. After writing our preliminary code to obtain data from the sensors and communicate to our application via the Wi-Fi module and microcontroller we study it and look for ways to optimize further. For example, all of our sensors are using serial communication and we can optimize the code to read from them in an efficient order, reducing clock cycles and reducing power consumption. Another way to improve our battery performance is to disable the ADC when it is not in use. This can be accomplished in two ways, the first being writing to registers directly to enable and disable the ADC, or by using freely available low-power software libraries. For this project, we are comfortable writing to the registers to save on code complexity and performance. The ATmega also sports a low power mode that draws just 0.75µA compared to 0.2mA in active mode.

7.1.3 Water Pump & Relay Circuit Testing

The pump we chose was selected because it was said that it would deliver the water flow we needed at a 5-volt input. To test this, we drive the motor with a 5-volt DC power supply and make sure the water is flowing at a desirable rate. Next, we integrate the relay circuit that drives this load if the test passes.

Each part in the relay circuit needs to be tested to ensure that it is functioning properly. The diode is connected to a small power supply to ground and checked for breakout voltages and the current flowing through it. The transistor has the same done to it by having a common collector setup with the diode attached to its connector to ground. We check that when a voltage is applied to the base current flows through it and can drive the current through the diode. Finally, the relay terminal is checked by having a higher voltage applied to its switch terminal to make sure it turns on the connect load of the water pump.

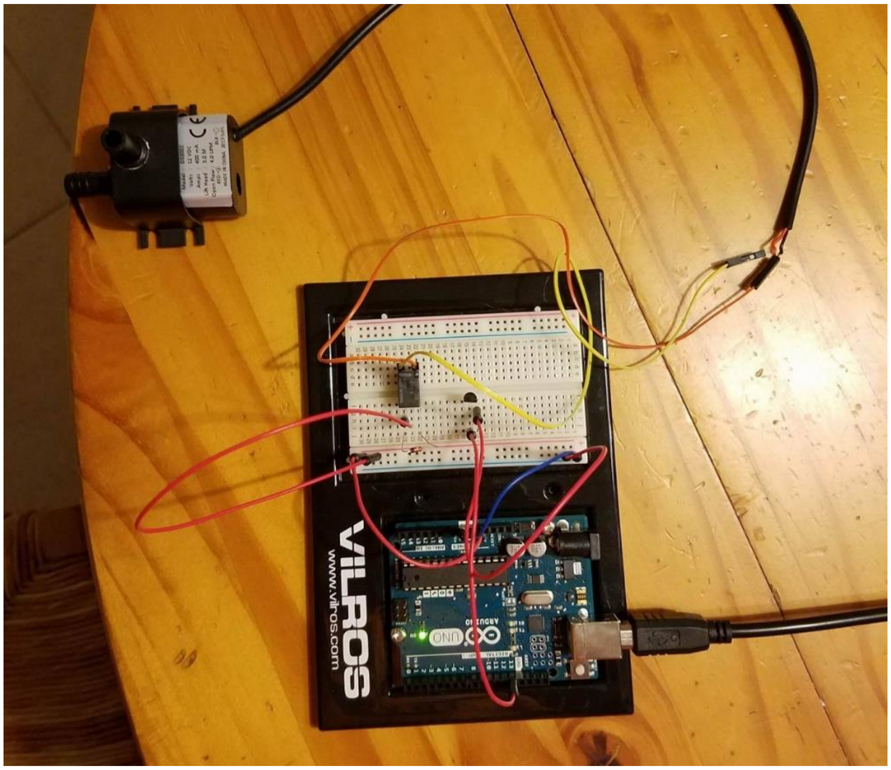


Figure 58: Water pump and relay test on Arduino dev board

After all the parts have been tested properly they are joined together and then tested off of the microcontroller. This is done with a 5-volt DC power supply being attached to the top node between the diode and relay terminal and a 3.3 volt DC power being sent through the base of the transistor, if the water pump turns on then the test was a success and it can be attached to the microcontroller's GPIO pin.

7.1.4 Sensor Testing

Sensors are tested individually on an Arduino development board initially, then, after we verify each sensor works independently of one another we start combining all the sensors onto one breadboard and test each one of them again. This is one of the easiest and most effective ways to go about testing our components because the Arduino board comes with easy plugins to GPIO and power pins. The microcontroller is programmed to read and output the sensor's data to a computer attached to the Arduino's USB port. It is important that we look for accurate values here because we need the sensors working properly together before we finalize them on our PCB. Some individual tests for each sensor are outlined below.

The DHT22 temperature and humidity sensor uses four pins and outputs a calibrated digital signal from the factory, so we shouldn't have to mess around with code to get proper values from the sensor after setting up serial communication. Initially, we test the sensor by connecting one pin to our power supply, one pin to ground, and another pin will be used as our data signal. According to its datasheet, one of the four pins is null and not used for anything. After power is supplied to the sensor, we need to wait at least one second before issuing any instructions to it since it will still be in its unstable power up phase. After we pass this phase, testing can begin. To test the sensor, it is important to know that a single cycle of communication between it and the microcontroller takes five milliseconds, so it is to be expected that any instructions issued in the middle of a request for data will probably cause some errors and with this knowledge we built in a small delay to our test code. Once properly connected, a serial channel will be set up and we will use the Arduino's digitalRead() function to grab data from the sensor. The DHT22 should send us 40 bits of data, with higher value bits being transmitted first. The data is divided into five octets: eight bits represent the integer portion of relative humidity, the next eight represent the decimal portion of relative humidity. The third and fourth octet are for the integer and decimal values of temperature, respectively. The last octet is an eight-bit checksum that can be used to verify our readings, it should be equal to the last eight bits of the four preceding octets added together.

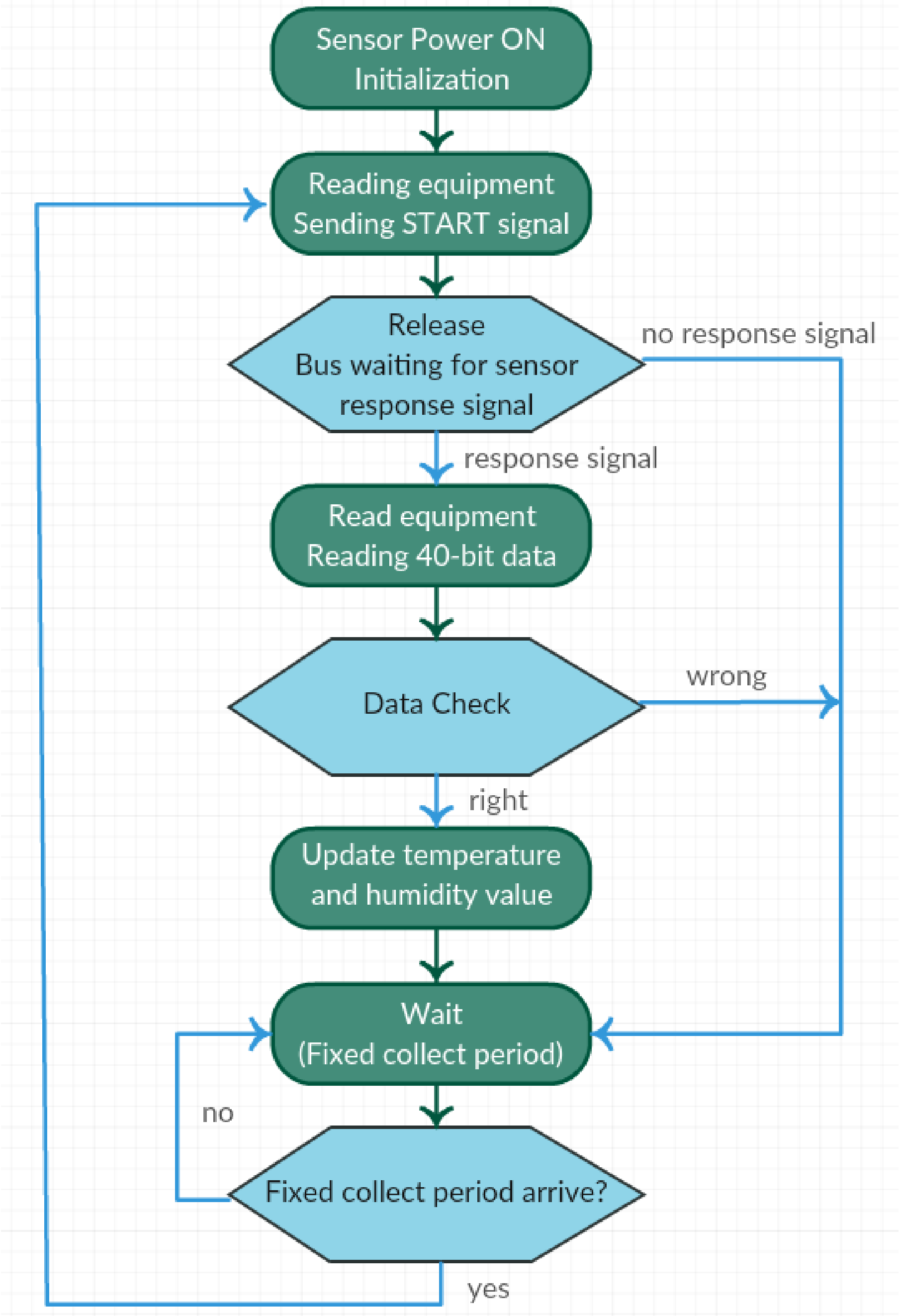


Figure 59: DHT22 Data Collection

The next sensor in line to be tested is the KP235 barometric pressure sensor. It has eight pins and features an analog output. It has data in and data out pins that are digital, however it is stated these are used only during calibration and testing in the factory, therefore we connect pin seven, VOUT, to an analog pin and pass the signal through an analog to digital converter in order to get something usable for our application. Because of its similar power up and response times, we need to wait at least five milliseconds between requests to this sensor, so even our stress tests grabbing near constant sensor data should adhere to this as the smallest possible delay. We are able to turn the output voltage into a pressure by following the given transfer function for the sensor, and we can further compare that to freely available local data. If there are any major discrepancies we can try compensating in our code, but if problems are intermittent or incorrect readings do not change linearly we have to seek other methods of getting our barometric pressure.

The KP235 sensor also supports broken wire detection, and we want to test that this is functioning correctly in case any wires come loose during normal use of our device. This shouldn't be the case by any means of normal use, but we need to verify that a broken connection in this sensor won't short out or damage the rest of our circuit. In the event that the ground connection or supply voltage are interrupted the output of the sensor we see its voltage levels drop to lower levels determined by the internal circuitry. We can monitor the sensor and if we ever see it reporting these values we can initiate a shutdown protocol so that we can safely attempt to fix the problem. If this problem happens to an end user then we can inform them through the application and attempt to bypass the KP235 sensor in our code until such a time that the issue can be fixed.

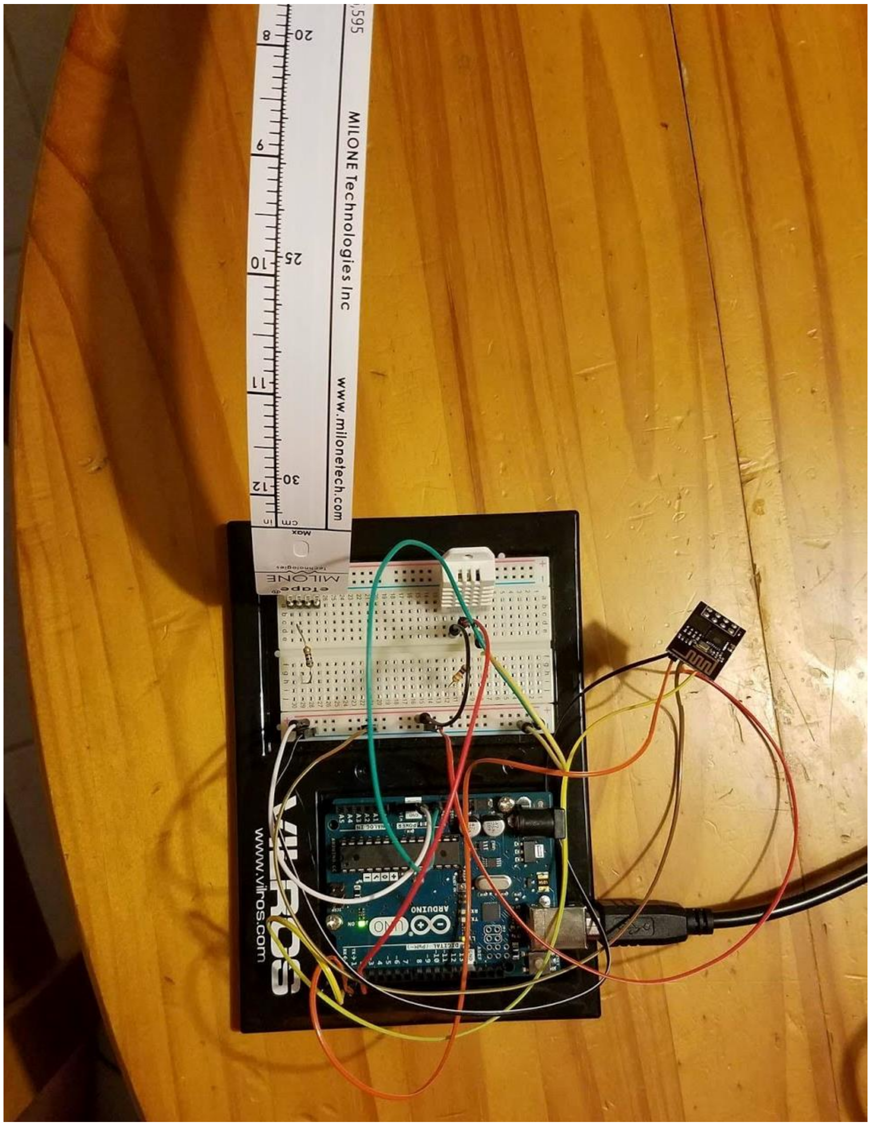


Figure 60: Sensor testing on Arduino dev board

Finally, we hooked up our eTape water level sensor. This is another sensor with analog output and it can be modeled as a variable resistor to represent how much water is left in the tank. It is hooked up to the board with a voltage divider. In order to test that the strip is working properly we are using our tank for the project. Because it is a uniform shape, we are able to mark the maximum and minimum heights for the tank, along with a few values in between. It is easy enough to pair readings from the strip sensor with what we are seeing with our eyes, it will be clear to spot a reading of 50% full from the sensor when the actual level is more or less than that, for example. Because this sensor is being modeled as a resistance that maps to certain level values, we shouldn't have to worry about oversampling on the hardware end unless we go overboard. We tested the limits of how quickly we can sample this sensor by itself and when it's connected to other components, but since we are only measuring the level of our tank with this sensor its measurements during actual use will be far apart.

7.1.5 Power Circuit Testing

The power circuit testing consists of two major tests, the battery and the voltage regulator. The battery can be tested using the tests listed above. To test the voltage regulator a DC voltage is applied to the input side of the device. The input from the power supply will be greater than 4.5 volts, to represent the battery. This should provide enough voltage to have the proper output voltage of 3.3 volts. If the 3.3 volts is not appearing, add a voltage divider to the output to ground and take a measurement from that. After the desired voltage has been reached move on to the full circuit test.

After the voltage regulator has passed, as well as the battery has been confirmed to be working properly, it can be attached together to form the basis of our power circuit. Using the 5 volts from the battery to the input of the regulator, measure the output of the regulator and make sure that you are receiving the desired 3.3-volt output, using the voltage divider resistors if necessary. Once the desired voltage has been reached the circuit passes the test.

7.1.6 Wi-Fi Module Testing

The Wi-Fi module will be tested when the microcontroller is on the Arduino development board. The power is supplied via a DC power supply so there are no fluctuations in power or voltage. During the first few test, we were having issues with the Arduino 3.3v pin not providing enough current to the ESP8266. We were able to flash the module with no problem but as soon as the program started, the ESP would simply top working. The reason for this we later found out was because the Arduino Dev Board only outputs about 50-60 mA of current through the 3.3v pin. The ESP needs over 100 mA of current when it is at its peak performance. The module itself is mounted on a breadboard and then pinned into the correct slots for each corresponding TX, RX and ground node.

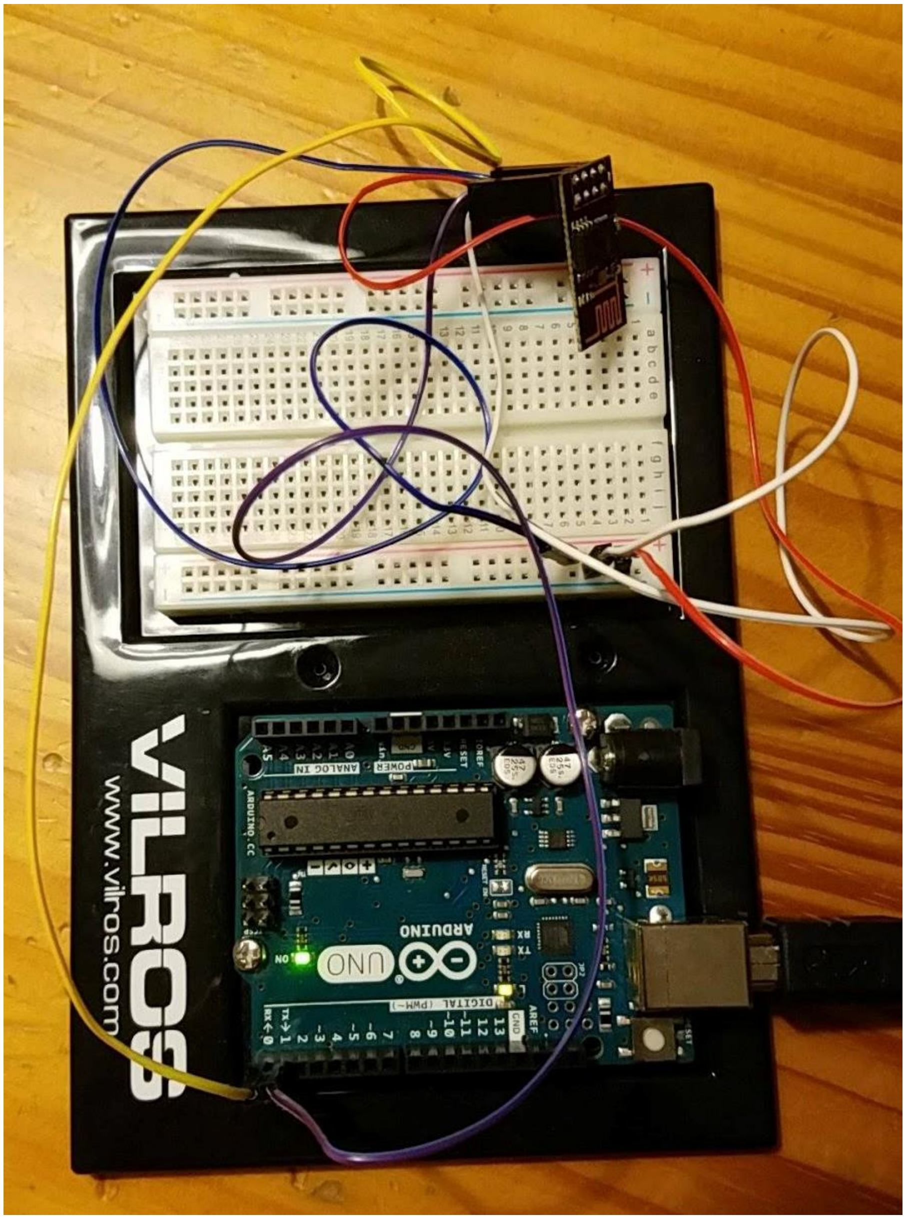


Figure 61: Wi-Fi module testing on Arduino dev board

The Arduino then runs a simple script to fetch data from a weather website. If the data is received properly it posts the data to the user via a terminal on the computer the Arduino dev-board is connected to. This proves the module is in working condition. If the Arduino fails to do so, another script will be ran to see if data is being transferred between the two modules or not. If there is data transferring, then some form of communication error might have occurred. If it is not transmitting data than the device failed and another device must be purchased to replace the faulty one.

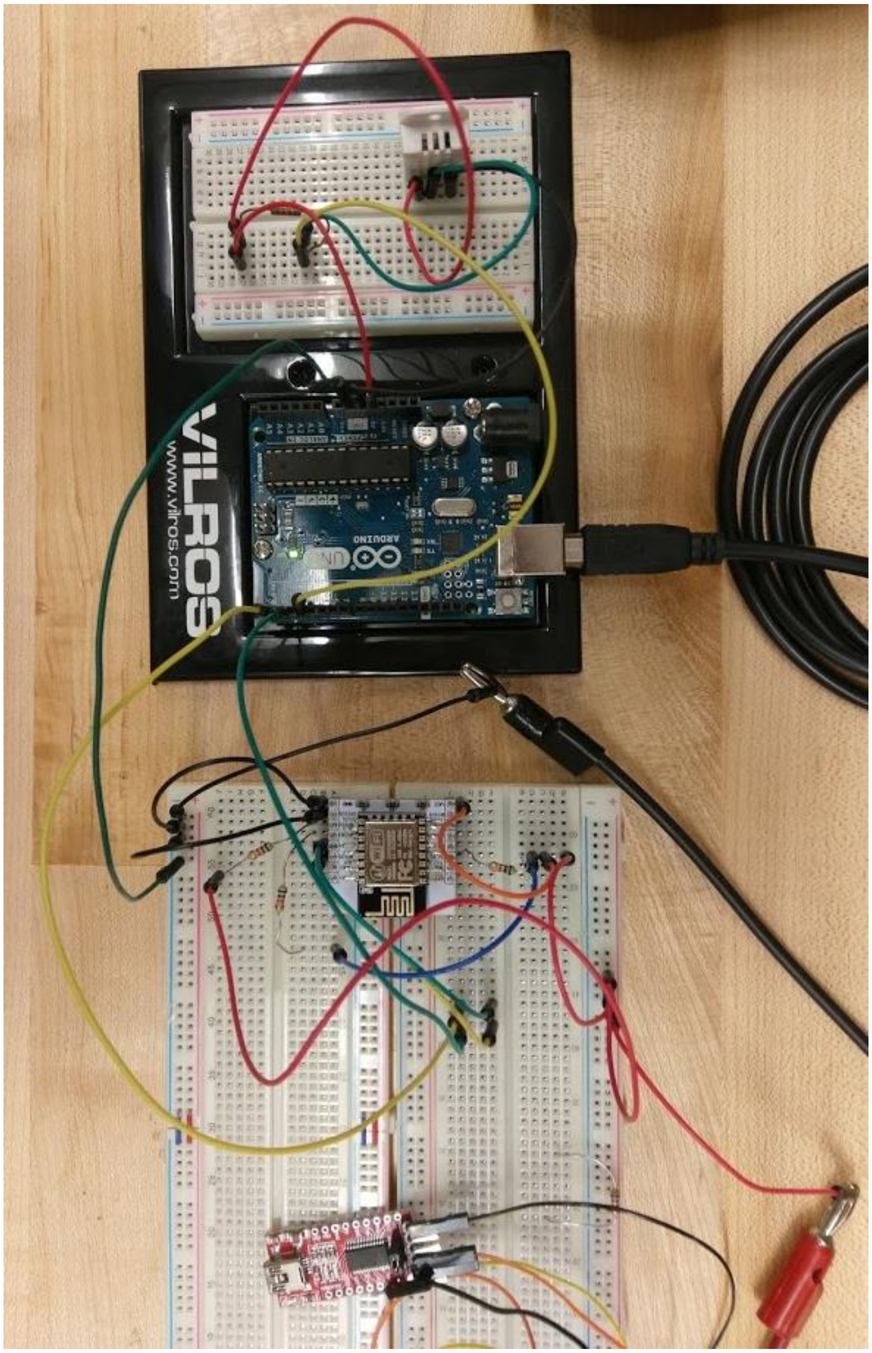


Figure 62: Flashing newer ESP8266-12e WiFi module

During testing, the ESP8266 module kept crashing and having issues with power. After some research, we realized that there was a newer version of the ESP8266 available (version 12e) which took care of some of those issues found in earlier models. This newer module was purchased and used for testing and development from there on out. The figure above showcases the newer module, the ESP8266-12e being flashed using a USB to Serial converter.

7.1.7 Final Testing Table

The table below represents how each component can pass or fail. If one of the components fails its test than the team must immediately order a new part in order to say on time with their project. Once all the items have passed their tests it is time to implement them together on a basic prototype board or breadboard. After all the systems and components are working together, then can the final PCB design be assembled, soldered and completed.

	Pass	Fail
Solar Panel	Able to charge the battery to full capacity and hold it stable with no burn outs	Unable to provide the voltage at a proper rate to the battery or at all
Battery	Supplies a 5V output at 2 amps holding its charge for a significant amount of time	Delivers a voltage less than 5V or a current less than 5V
Microcontroller	Able to receive input and push output, power on, display to a user information from sensors, operate at a low power	Doesn't do one of the following: receive input or push output power on display info.
Relay Circuit	Circuit gets switched by input voltage and can drive the water pump through a relay terminal	Circuit is not turned on by the input voltage or the pump cannot be driven by the circuit
Power Circuit	Able to convert from 5V to 3.3V and keep constant current out	Unable to convert volts down or keep a constant current
Wi-Fi Module	Transmits data from website to Arduino for printing out	Doesn't transmit any form of data between the two devices

Table 34: Hardware components testing

7.2 Software Testing

7.2.1 Microcontroller Software

Software testing for the microcontroller is a little different than for the smartphone application. Microcontroller software testing begins with the arrival of sensor data to the microcontroller. The analog sensors we are using required some research to find out how they communicate with our development board and microcontroller. Once we figured out how the sensors should be connected to the board we researched how the analog to digital conversions would be done to the signals. It just so happens that the equipment we are using can handle the analog to digital conversions for us, all we need to do is call a specific function when we want to grab data. Before we begin the transfer of data, we'll want to set up a reliable serial connection that sends us an average of a few samples with appropriate delays in between requests. Once we have setup a reliable connection between the sensors and the microcontroller, it is time to stress test every sensor individually to make sure that they are not going to randomly fail later down the road.

The way the sensors were tested was by gradually decreasing the interval in which data is pulled from them until we are pulling data from the sensors constantly or at least a lot more frequently than we would ever need to, just to make sure the sensor can handle it. Testing the sensors in this manner allows us to see how small the interval can be between requests for data from the sensors. Although an end user should not be able to force the sensors to report so frequently we want to be prepared with the knowledge of when our sensors start to falter so that we can adjust our default readings accordingly.

Once we have tested all the sensors, then it is time to test the microcontroller itself. This begins by having all the sensors send data to it at once as frequently as the sensors can to see if the microcontroller can handle it. Once it is validated that the microcontroller can handle input from every sensor at the same time, it is time to move on to testing within the microcontroller itself. At this point we play around with the different clock speeds, the low-power mode and other built-in functions to make sure the microcontroller behaves as expected and we don't have a faulty unit.

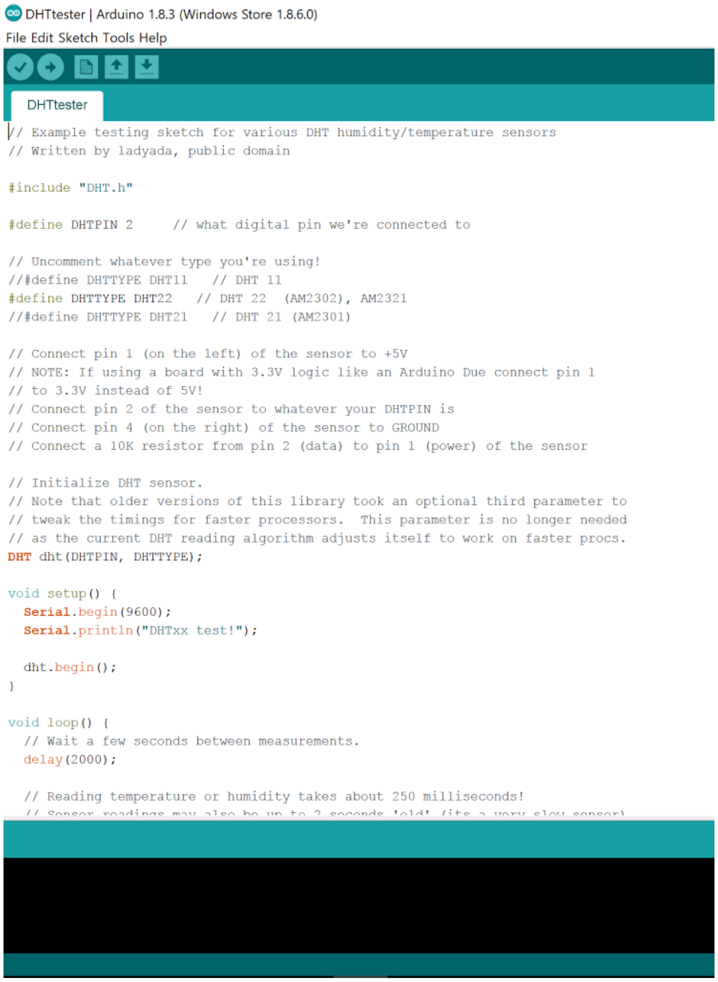
The next step of software testing for the microcontroller is having it execute the various functions and commands we write to read data from the sensors and react accordingly. This is also when we put our "Automatic Watering Mode" to the test using a controlled environment. Once everything was working as expected and most of the bugs had been figured out for this phase of testing, it was time to start communicating with the Wi-Fi Module.

Communication with the Wi-Fi Module was its own stage of testing since it was what took us the most time to figure out and get working correctly. The first step is to establish a solid connection between the microcontroller and the Wi-Fi Module. Once they can talk to each other, we start broadcasting a signal to see if the smartphone can pick it up.

The last step is to verify that the microcontroller and the smartphone are indeed connected wirelessly and can communicate with each other. Once they are communicating, only thing left to do is stress test both systems together to make sure that any little bug that appears can be fixed before the actual release.

After this round of testing under ideal conditions, we plan to take our project through two more, similar, phases of testing. This time, though, we are going to run through the same previously discussed tests under more extreme temperatures. While we still are within the ideal operating ranges of all of our components, we wanted to see how our system fares under the higher heat and humidity of a Florida summer and colder temperatures of a walk-in refrigerator. Once we verified that our system works in all of these conditions, we felt more confident presenting our system.

Luckily enough the Atmega328P is a microcontroller that is able to have the Arduino IDE bootloader onto it. This allows for lower level programming, done in a higher-level language like C. By using the Arduino Uno dev board and the Arduino IDE we are able to continuously program and tweak our code until the final programming goal is achieved. With the vast libraries and helpful community using the Arduino IDE helps this project out immensely in terms of programming time and efficient code.


DHTtester | Arduino 1.8.3 (Windows Store 1.8.6.0)
File Edit Sketch Tools Help
DHTtester

```
// Example testing sketch for various DHT humidity/temperature sensors
// Written by ladyada, public domain

#include "DHT.h"

#define DHTPIN 2     // what digital pin we're connected to

// Uncomment whatever type you're using!
//#define DHTTYPE DHT11   // DHT 11
#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321
//#define DHTTYPE DHT21   // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V
// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1
// to 3.3V instead of 5V!
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 4 (on the right) of the sensor to GROUND
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor.
// Note that older versions of this library took an optional third parameter to
// tweak the timings for faster processors.  This parameter is no longer needed
// as the current DHT reading algorithm adjusts itself to work on faster procs.
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");

  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
```

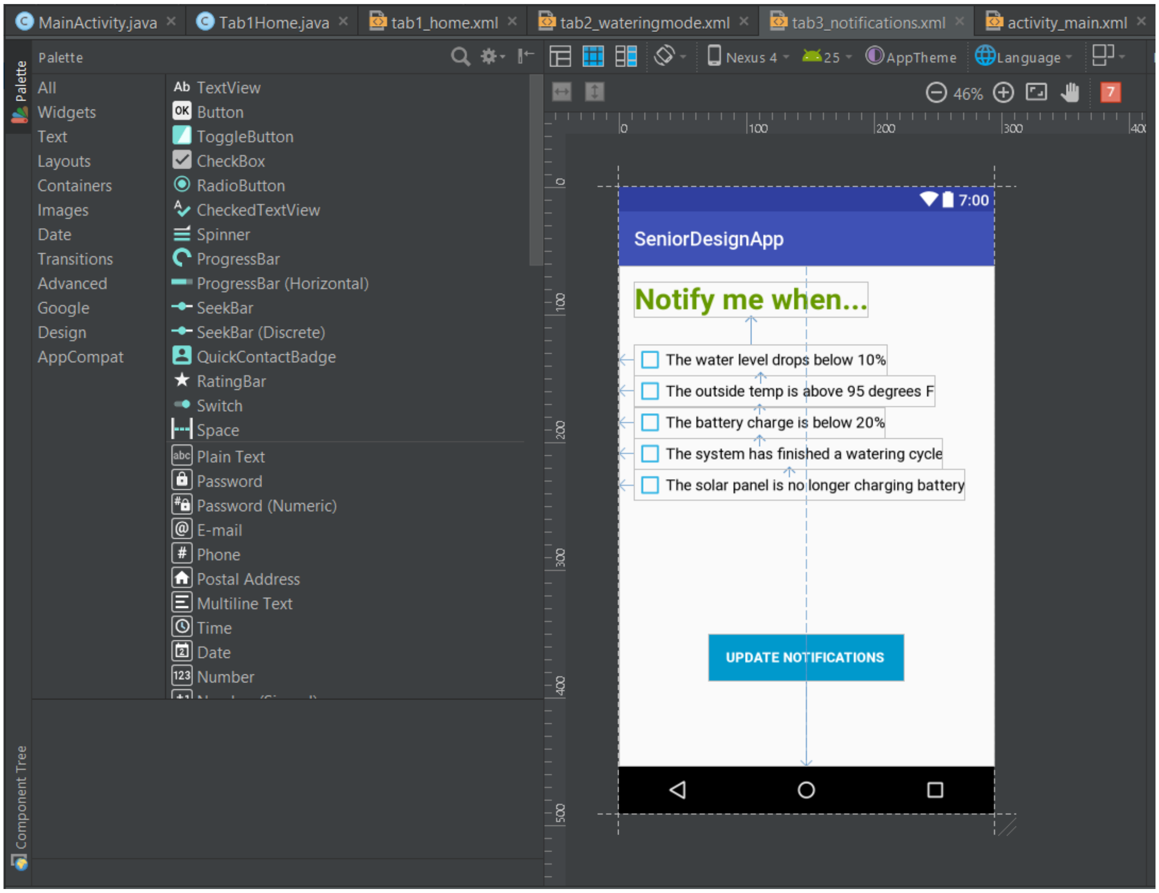
Figure 63: Arduino IDE running temp sensor test

7.2.2 Smartphone Application

Testing of the smartphone application was conducted in stages. The first stage of testing took place on Android Studios as the application was being developed. Throughout development, every individual feature was tested independently. Once multiple parts had been developed and tested independently, the second stage began.

During the second stage of testing, different groups of features were tested as units to see how the program behaves as a whole and how the different features interact with each other. Testing for this stage was done using an Android phone (Google Pixel) emulator on the computer. This is the stage where many bugs started appearing and development slowed down because a fault in one of the features had an effect on the whole system, which took some time to completely debug.

The final stage of testing is when the whole system is ready for its first stress test. During stress testing, the system is used in every single way possible (even in ways it was not meant to be used) to see if you can cause the application to crash. At this point is when we loaded the application to an actual Android device and left the Android Studios simulator behind. This is usually where you find the little bugs that you have to counteract with “if-statements” and error messages to the user. Once the developer is doing with his stress testing, it is time to let a select group of outside users interact with the application to see if they can break something or at least provide some feedback. At this point is when we were able to finally test the communication between the WiFi module and the android application using HTTP requests. Many issues were encountered here and is one of the reasons why the application evolved from prototype v1 into prototype v2 as showcased in a previous section.



Notify me when...

□ The water level drops below 10%

□ The outside temp is above 95 degrees F

□ The battery charge is below 20%

□ The system has finished a watering cycle

□ The solar panel is no longer charging battery

Figure 64: Android Studios Testing

7.3 Demonstration Plan

This section details how our team plans to present our system at the end of Senior Design II. We give a quick summary of our project that includes why we decided to take on this project, our goals, and a short run-down of components. These components include:

- ATmega328P microcontroller, the brains of our project

- Water level, temperature, humidity, and pressure sensors, how they interact serially with the microcontroller, and how we use that data in our smartphone application
- Wi-Fi module, which takes care of communications between our microcontroller and application
- Water pump
- Solar panel and battery, which we chose as a compromise between battery life and portability

We then plan to demonstrate the physical system itself, how small it's final size is, as well as demonstrate its portability and how easy it is for someone to set up. Once we have talked about our components and setup our system we began talking about how we pull data from sensors, the basic ideas being:

- Initialize the microcontroller and any variables that we'll be using throughout our demonstration and normal use
- Setup a serial communication channel between the microcontroller and sensors
- Explain the order and frequency of how we poll the sensors, taking care to mention that we use an average of five samples to avoid circuit noise and errant data
- After attaining values from the sensors, we stored those values and transport them through our Wi-Fi module to our smartphone application

At this point we take a break from showing off the hardware and switch gears into showing off our software application. We then run through the home screen of our application, which shows the current time, battery level, water level, temperature, humidity, and pressure values that have been sent over from the microcontroller. From here, we'll trigger the manual watering mode from inside the smartphone application until the water tank runs to a low enough that an event notification is sent to the user's phone.

We then explained how we combine data from the temperature, humidity, and pressure sensors to automatically determine when to water plants. This may be triggered automatically depending on the weather and length of our presentation, however, if it's not we can just explain the logic behind it. This ends the software side of our presentation, and we can return to the hardware to explain the quirks of our water level sensor, mentioning:

- How the water level sensor was calibrated for our tank
- Alternatives such as the ultrasonic sensor, mentioning its advantages and disadvantages

Finally, at the end of our presentation we talked about the features we wanted to add but weren't able to due to time or monetary constraints. We'll also field any questions from the panel of professors.

8. Administrative Content

In order to be organized and keep our deadlines in check it is important to schedule meetings through the group and create a list of when certain assignments are due. Being on top of our work and making sure we are ahead of the due dates instead of on time or behind is crucial to the success of this project. The team should work together in order to organize the best plan for everyone to work around.

This section goes over:

- The due dates for assignments, their status, and who is heading up the subject at hand.
- The financial division for each group mate given the costs of components.

8.1 Milestone Discussion

Senior Design I				
Task	**Start**	**End**	**Status**	**Lead**
Develop & Select Idea	5/24/2017	5/31/2017	Complete	Group 9
Assign Project Responsibilities	5/30/2017	6/2/2017	Complete	Group 9
Report Documentation				
Initial Documentation - Divide and Conquer	5/30/2017	6/2/2017	Complete	Group 9
Table of Contents	6/1/2017	7/7/2017	Complete	Group 9
First Draft	6/3/2017	7/7/2017	Complete	Group 9
Second Draft	7/8/2017	7/21/2017	Complete	Group 9
Final Draft	7/22/2017	8/1/2017	Complete	Group 9
Research and Design				
Component Research & Comparison	6/2/2017	7/7/2017	Complete	Chris / Ronak
Circuit Design	6/2/2017	7/13/2017	Complete	Peter
Application Research & Design	6/2/2017	7/7/2017	Complete	Joan
Order Components	7/14/2017	7/21/2017	Complete	Group 9
Order PCB	7/14/2017	7/21/2017	Complete	Group 9

Table 35: Senior Design I Milestone Table

Senior Design II				
Task	**Start**	**End**	**Status**	**Lead**
Test Phase I				
Build Prototype	11/4/2017	11/28/2017	Complete	Group 9
Test Prototype	11/7/2017	11/28/2017	Complete	Group 9
Redesign PCB Layout & Reorder if Necessary	11/7/2017	11/28/2017	Complete	Group 9
Test Phase II				
Build & Test Prototype	11/16/2017	11/28/2017	Complete	Group 9
Research & Acquire Final Component Enclosure	11/16/2017	11/28/2017	Complete	Group 9
Final Presentation	11/28/2017	11/28/2017	Complete	Group 9

Table 36: Senior Design II Milestone Table

8.2 Budget and Finance Discussion

Component	Price	Quantity
Water Pump	$12	1
Solar Panel & Battery	$159	1
PCB	$35	2
Micro-controller	$2.18	3
Temperature Sensor	$9.95	2
Humidity Sensor		
Barometric Sensor	$8.43	3
Wi-Fi Module	$6.95	5
Water Level Sensor	$39.99	1
Water Tank	$2	1
Miscellaneous Parts	$20	variable

Table 37: Project Budget Table

8.2.1 Estimated Cost of Device

A variable cost for small parts such as transistors, diodes and resistors should be taken into account here. While this may not affect the budget too much, it is important to stay under the $20 allocated funds. If we were to go past this then the project may be burning out parts and there would be a serious underlying issue.

Taking the minimum of each part of our equipment we can estimate the least we spend to be around $300 while using a $35 PCB. This is the price for one “unit” at cost. However, multiple parts have been bought and replaced with new ones because of soldering and burning out issues. The final cost of the project for the team was closer to the $430 mark, including all the extra parts needed for testing.

8.2.2 Funding

This project was entirely funded by the team. It was split evenly amongst the four teammates and did not have a sponsor to take over the cost of the project. This means each member was responsible for one-fourth of the cost which was around $430 for the entire project. Each member of the group agreed to this stipulation and believed it was fair. Receipts and other forms of proof of purchase were saved by each team member and presented at a later date to form a final total for the project, which gave us an idea of who spent how much, and how each member should be reimbursed.

9. Conclusion

The research and development of the self-sustaining plant watering system was at first a large task to handle with only four group members. The research needed for the solar designs, microcontroller, sensors and pumps were vast seeing as though the market was flooded with numerous designs and we had to obtain the best ones for our project.

Our creation, a self-sustaining plant watering system, was met with many complex and compact designs already marketed today. With such a vast number of designs to compare to, we were able to shape our design into the best possible way for the constraints we obtained ourselves. By combining the power of the sun, with the low-power-mode based microcontroller, we were able to supply enough power to have each sensor function properly as well as obtain a desirable amount of force from the pump to push out water for our plants.

Our goal was to design a simple and easy to take care of system that could operate on its own or manually. The integration of the software application with the hardware allows our device to complete these tasks in an efficient way, leaving little for the user to do. The only interference needed is to set a time or state to water the plants and adding water to the tank when it is low. This design is ideal for those busy workers or students who don’t have time to take care of their vegetation. As a product that could be marketed to a broad range of users, our goal of creating a widely used device was met.

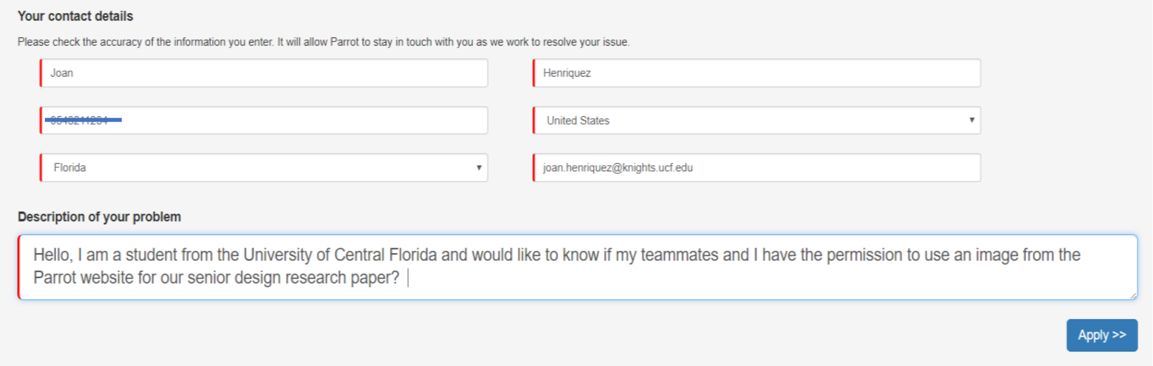
Although we ran into multiple issues throughout Senior Design 2, it was all just a giant learning experience. OSHPark taught us that even when you send them a good PCB design, they can still send you a PCB that will not work even after waiting multiple weeks. The ESP8266 WiFi module taught us that just because the module works in one setting, it does not mean that it will work everywhere and always the same. Every tutorial we watched and every article we read always mentioned that it needed to be powered by 3.3v and that any more voltage could damage it. However, they never mentioned that it needed around 100 mA of current to function properly, something we learned in the lab with fellow colleagues after trying multiple 3.3v power sources. Lastly, even when we thought we were completely done and ready for our demo, disaster struck and our only working PCB that was left broke while assembling the device. These events, along with many others, were the ones that brought the team together and made us come back better than ever every single time.

In conclusion, Senior Design is more than just the final class before graduation. It is also more than just another group project. Senior Design has not only taught us how to design PCBs, create mobile applications, write long papers and communicate using serial and HTTP request, but it has also taught us how to treat team members as family, how to solve problems when things don’t go as expected, and how to manage time effectively in order to succeed in this class while taking other classes, working park-time and having a social life. Overall, Senior Design is a learning experience that applies more than just the engineering field and that will definitely not be forgotten.

Appendix A – Copyright Permissions

In this section, we cite the source of all our images that were taken from the internet and request permission from their author. While we understand that not everyone will reach back to us either granting or denying our request for permission, we will still send an email out to everyone we have used an image or figure from.

Figure 4: Commercially sold self-watering pot
https://www.parrot.com/us/connected-garden/parrot-pot#parrot-pot-details
Permission Status: Requested



Your contact details

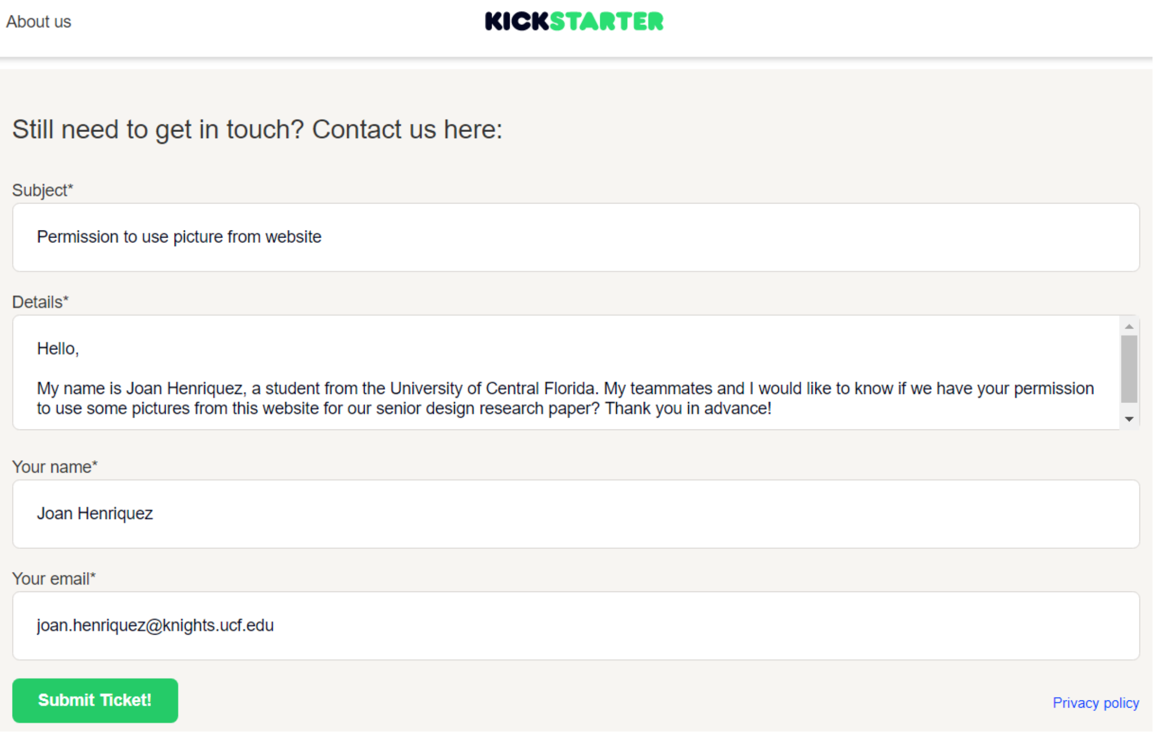
Please check the accuracy of the information you enter. It will allow Parrot to stay in touch with you as we work to resolve your issue.

Joan	Henriquez
[redacted]	United States ▾
Florida ▾	joan.henriquez@knights.ucf.edu

Description of your problem

Hello, I am a student from the University of Central Florida and would like to know if my teammates and I have the permission to use an image from the Parrot website for our senior design research paper?

Figure 5: Homemade IoT-based plant watering system
https://www.kickstarter.com/projects/sunair/smartplant-pi-smart-garden-with-your-raspberry-pi
Permission Status: Requested



Still need to get in touch? Contact us here:

Subject*
Permission to use picture from website

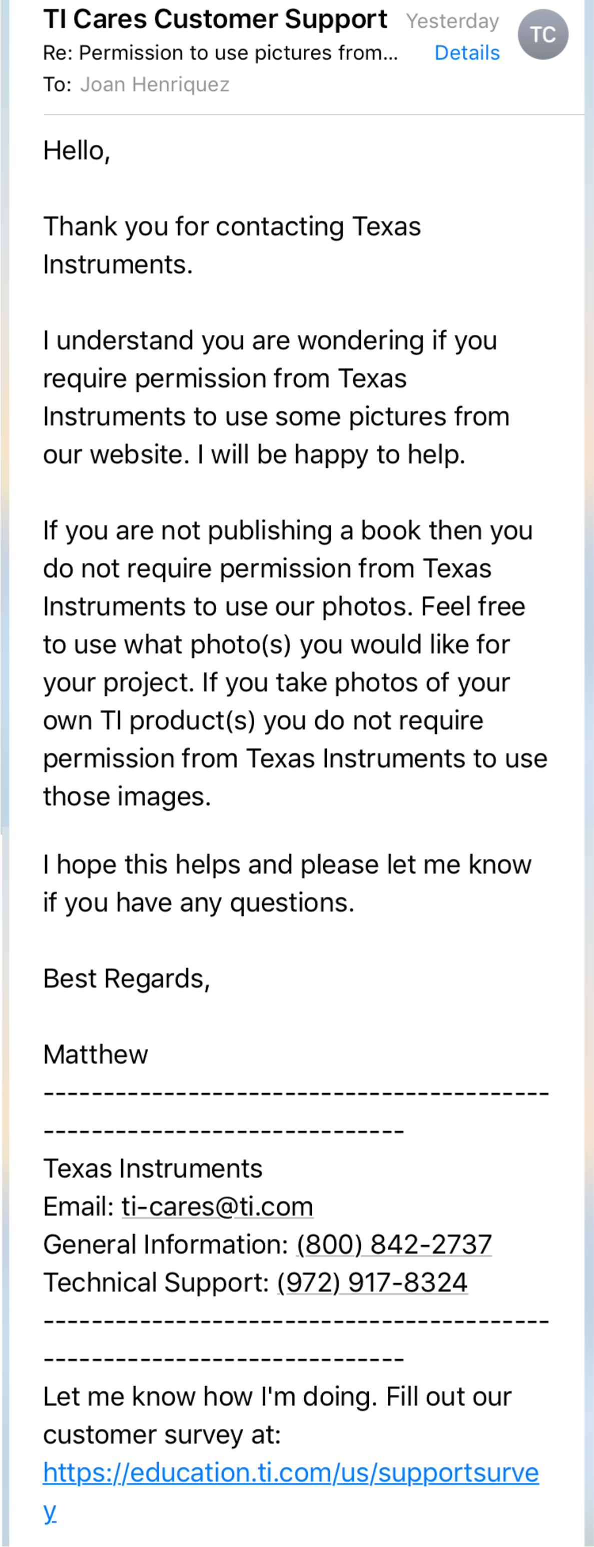
Details*
Hello,

My name is Joan Henriquez, a student from the University of Central Florida. My teammates and I would like to know if we have your permission to use some pictures from this website for our senior design research paper? Thank you in advance!

Your name*
Joan Henriquez

Your email*
joan.henriquez@knights.ucf.edu

Figure 6: Current draw compared to frequency of clock
http://www.ti.com/ods/images/SLVSAF6A/g_iam_fdco_las694.gif
Permission Status: **Granted**



TI Cares Customer Support
Re: Permission to use pictures from...
To: Joan Henriquez

Hello,

Thank you for contacting Texas Instruments.

I understand you are wondering if you require permission from Texas Instruments to use some pictures from our website. I will be happy to help.

If you are not publishing a book then you do not require permission from Texas Instruments to use our photos. Feel free to use what photo(s) you would like for your project. If you take photos of your own TI product(s) you do not require permission from Texas Instruments to use those images.

I hope this helps and please let me know if you have any questions.

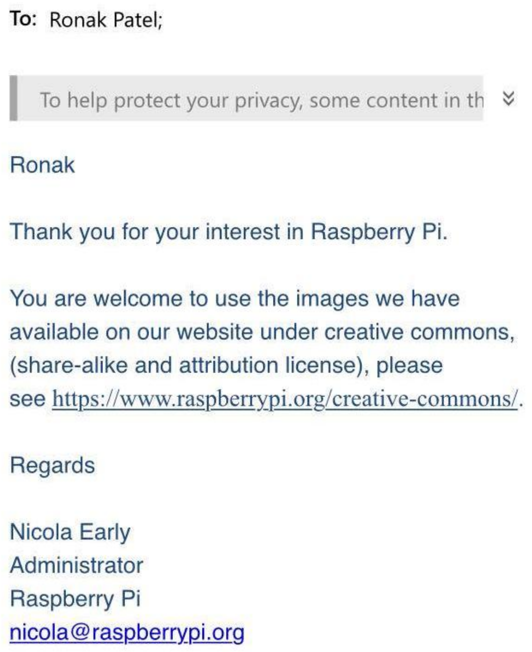
Best Regards,

Matthew

Texas Instruments
Email: ti-cares@ti.com
General Information: (800) 842-2737
Technical Support: (972) 917-8324

Let me know how I'm doing. Fill out our customer survey at: https://education.ti.com/us/supportsurvey

Figure 7: Standard PCB pin headers for GPIO pins
https://www.raspberrypi.org/documentation/usage/gpio/
Permission Status: **Granted**



To: Ronak Patel;

To help protect your privacy, some content in th

Ronak

Thank you for your interest in Raspberry Pi.

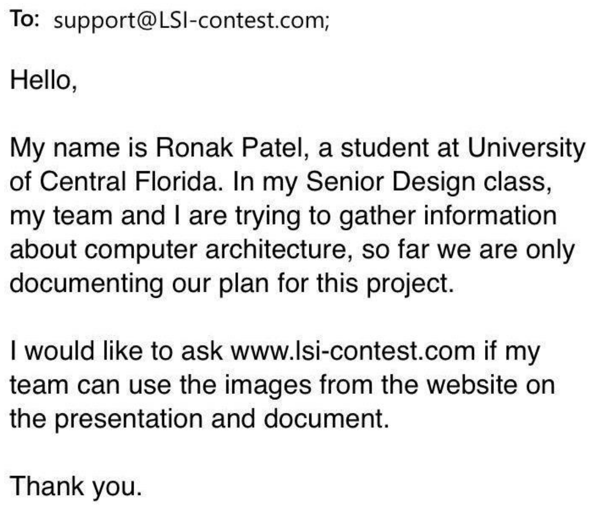
You are welcome to use the images we have available on our website under creative commons, (share-alike and attribution license), please see https://www.raspberrypi.org/creative-commons/.

Regards

Nicola Early
Administrator
Raspberry Pi
nicola@raspberrypi.org

Figure 9: instruction set architecture comparison
http://archive.cnx.org/contents/818b2af4-c1a1-422e-9c47-b5c2fb0f8d9a@3/what-is-high-performance-computing-fundamental-of-risc
Permission Status: **No Contact Information Found**

Figure 10: Basic layout architecture for RISC Processor
http://www.lsi-contest.com/2009/siyou-1e.html
Permission Status: Requested



To: support@LSI-contest.com;

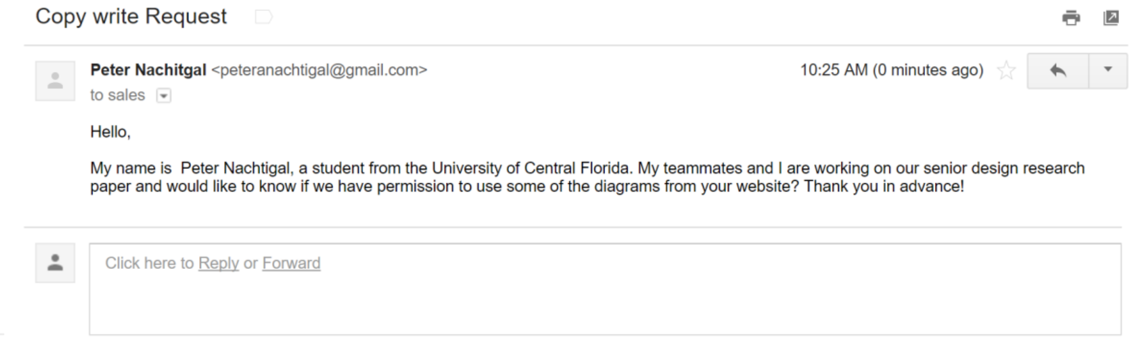
Hello,

My name is Ronak Patel, a student at University of Central Florida. In my Senior Design class, my team and I are trying to gather information about computer architecture, so far we are only documenting our plan for this project.

I would like to ask www.lsi-contest.com if my team can use the images from the website on the presentation and document.

Thank you.

Figure 11: Composition of solar panel
https://etap.com/renewable-energy/photovoltaic-array-fundamentals/
Permission Status: Requested



Copy write Request

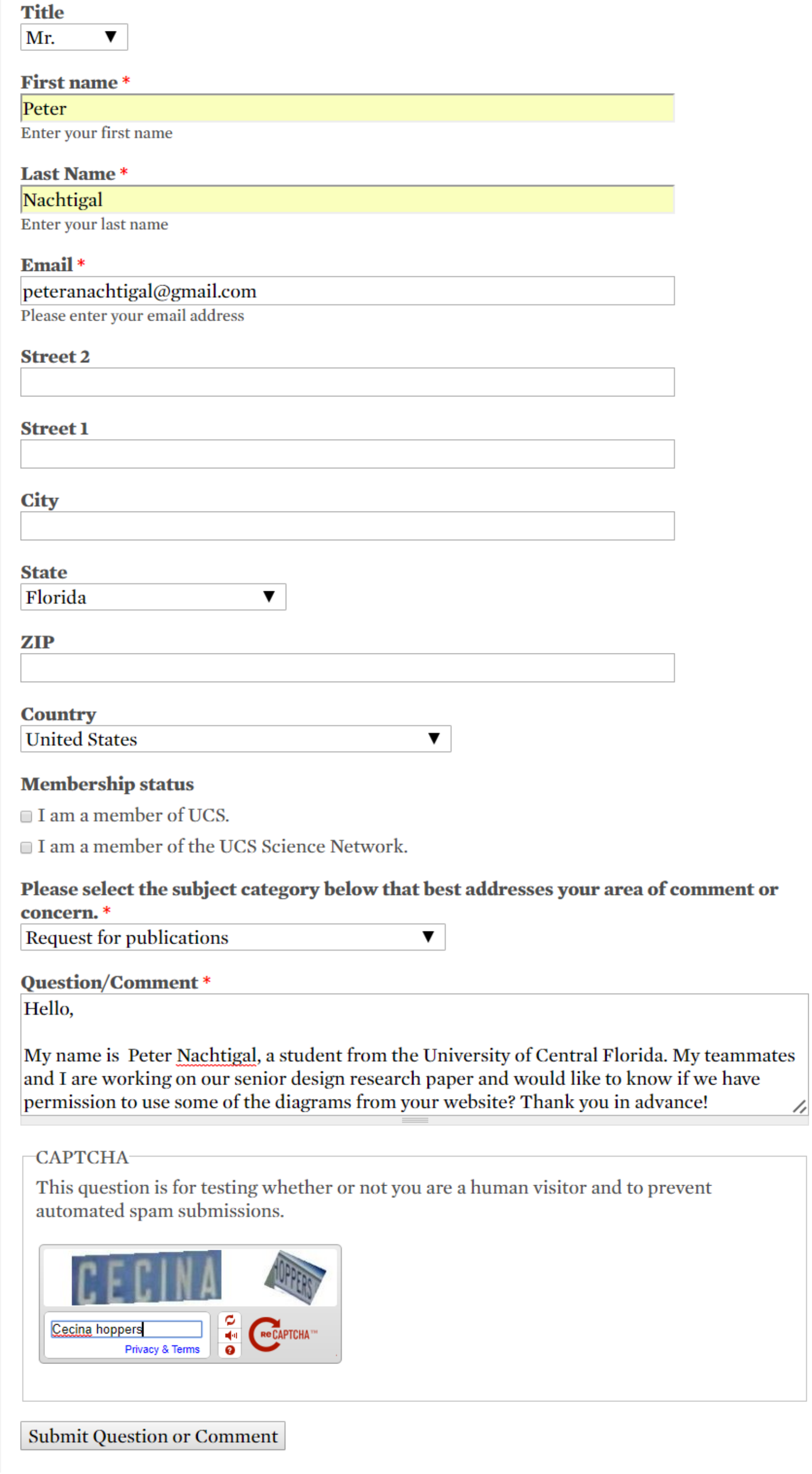
Peter Nachitgal <peteranachtigal@gmail.com>
to sales

10:25 AM (0 minutes ago)

Hello,

My name is Peter Nachtigal, a student from the University of Central Florida. My teammates and I are working on our senior design research paper and would like to know if we have permission to use some of the diagrams from your website? Thank you in advance!

Figure 12: Role of Photovoltaic Cell
http://www.ucsusa.org/clean-energy/renewable-energy/how-solar-panels-work#.WV4h5jOZORt
Permission Status: Requested



Title
Mr. ▼

First name *
Peter
Enter your first name

Last Name *
Nachtigal
Enter your last name

Email *
peteranachtigal@gmail.com
Please enter your email address

Street 2

Street 1

City

State
Florida ▼

ZIP

Country
United States ▼

Membership status

□ I am a member of UCS.

□ I am a member of the UCS Science Network.

Please select the subject category below that best addresses your area of comment or concern. *
Request for publications ▼

Question/Comment *

Hello,

My name is Peter Nachtigal, a student from the University of Central Florida. My teammates and I are working on our senior design research paper and would like to know if we have permission to use some of the diagrams from your website? Thank you in advance!

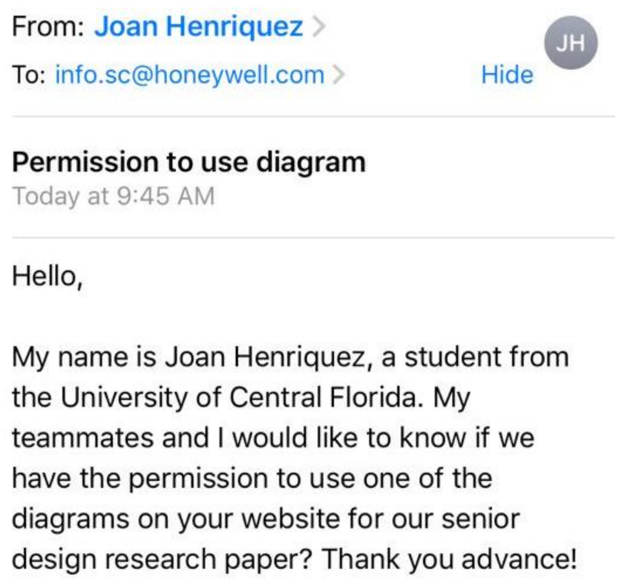
CAPTCHA

This question is for testing whether or not you are a human visitor and to prevent automated spam submissions.

Cecina hoppers

Submit Question or Comment

Figure 13: Sensor coatings
https://sensing.honeywell.com/index.php?ci_id=49925&la_id=1
Permission Status: Requested



From: Joan Henriquez

To: info.sc@honeywell.com

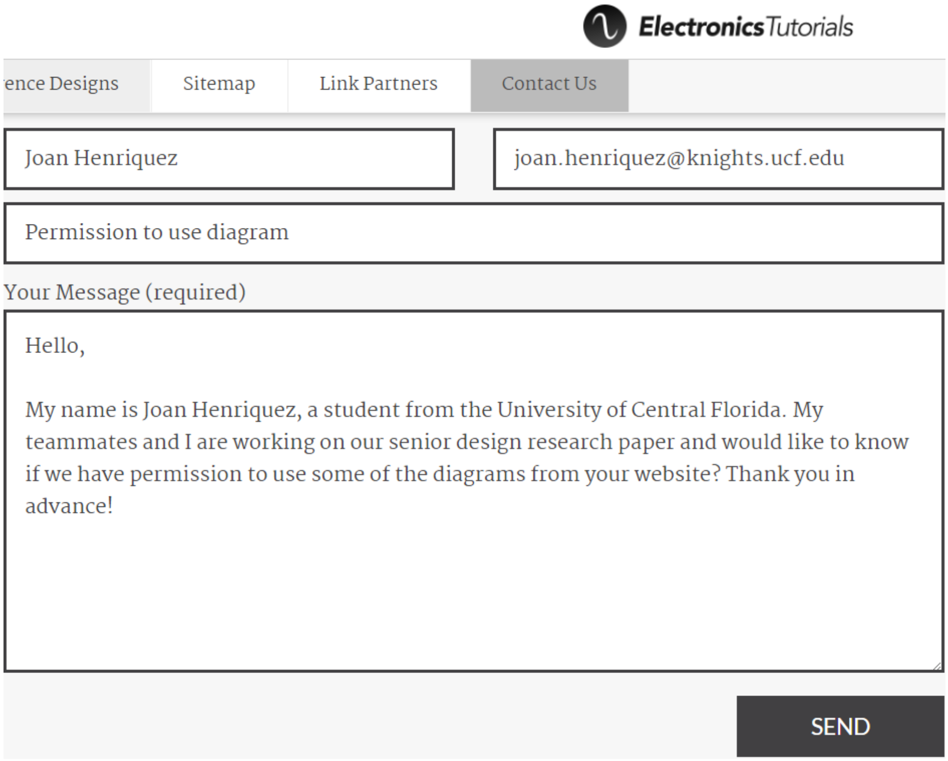
Permission to use diagram

Today at 9:45 AM

Hello,

My name is Joan Henriquez, a student from the University of Central Florida. My teammates and I would like to know if we have the permission to use one of the diagrams on your website for our senior design research paper? Thank you advance!

Figure 14: Negative temperature coefficient design
http://www.electronics-tutorials.ws/io/io_3.html
Permission Status: Requested



Joan Henriquez

joan.henriquez@knights.ucf.edu

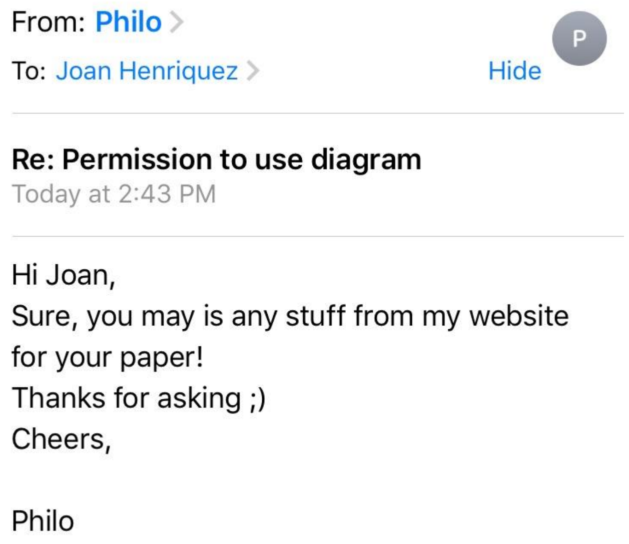
Permission to use diagram

Your Message (required)

Hello,

My name is Joan Henriquez, a student from the University of Central Florida. My teammates and I are working on our senior design research paper and would like to know if we have permission to use some of the diagrams from your website? Thank you in advance!

Figure 15: Diagram of basic laser sensor
http://www.philohome.com/sensors/lasersensor.htm
Permission Status: **Granted**



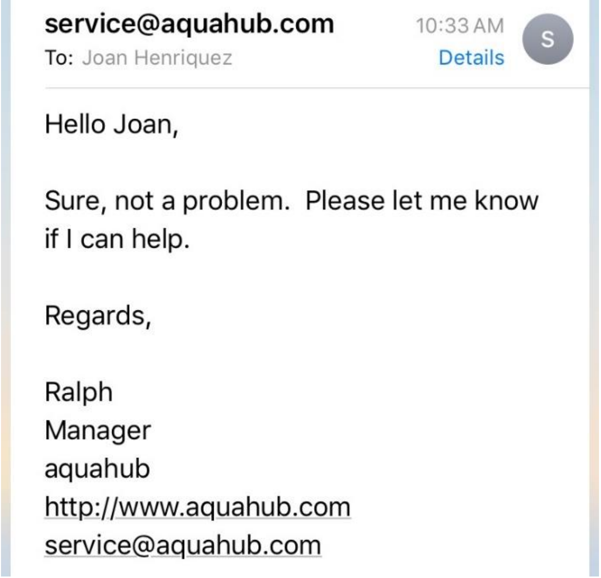
From: Philo >
To: Joan Henriquez >
Hide

Re: Permission to use diagram
Today at 2:43 PM

Hi Joan,
Sure, you may is any stuff from my website for your paper!
Thanks for asking ;)
Cheers,

Philo

Figure 16: Float sensor diagram
http://www.aquahub.com/store/media/floatwork-2.jpg
Permission Status: **Granted**



service@aquahub.com
10:33 AM
To: Joan Henriquez
Details

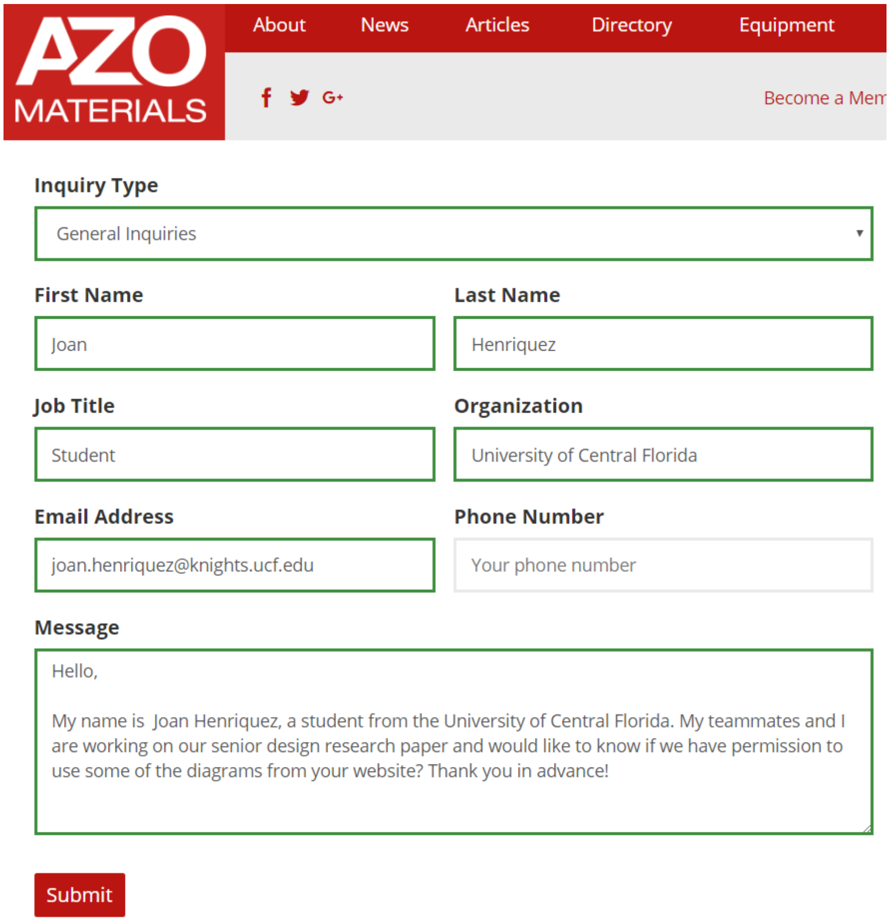
Hello Joan,

Sure, not a problem. Please let me know if I can help.

Regards,

Ralph
Manager
aquahub
http://www.aquahub.com
service@aquahub.com

Figure 17: Ultrasonic sensor
http://www.azom.com/article.aspx?ArticleID=12457
Permission Status: Requested



Inquiry Type

General Inquiries

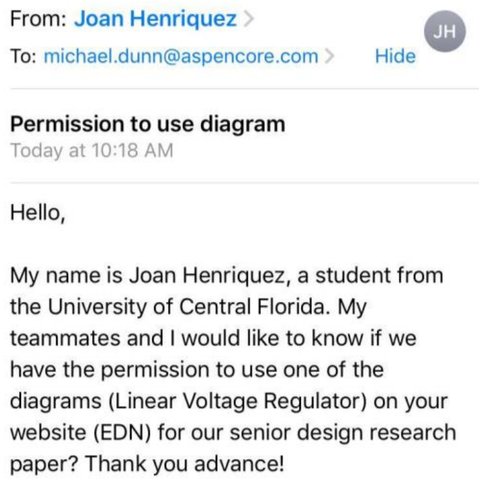
First Name	**Last Name**
Joan	Henriquez
Job Title	**Organization**
Student	University of Central Florida
Email Address	**Phone Number**
joan.henriquez@knights.ucf.edu	Your phone number

Message

Hello,

My name is Joan Henriquez, a student from the University of Central Florida. My teammates and I are working on our senior design research paper and would like to know if we have permission to use some of the diagrams from your website? Thank you in advance!

Figure 19: Linear voltage regulator
http://www.edn.com/Home/PrintView?contentItemId=4420313
Permission Status: Requested



From: Joan Henriquez

To: michael.dunn@aspencore.com

Permission to use diagram
Today at 10:18 AM

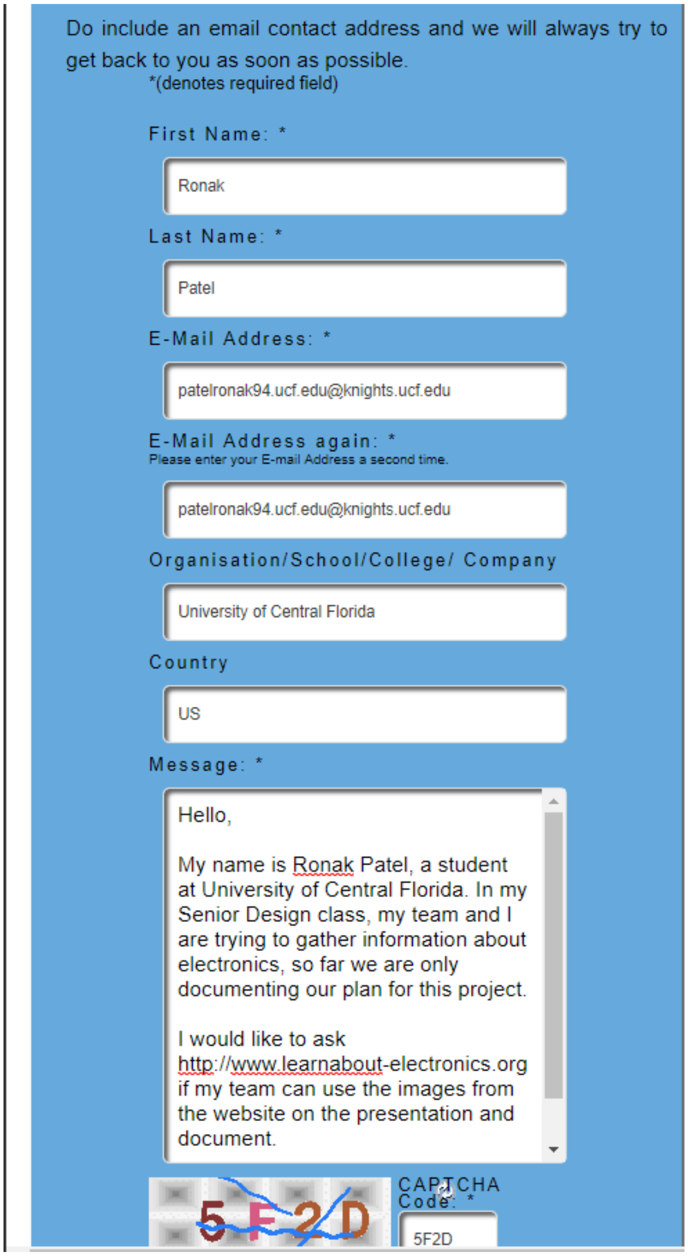
Hello,

My name is Joan Henriquez, a student from the University of Central Florida. My teammates and I would like to know if we have the permission to use one of the diagrams (Linear Voltage Regulator) on your website (EDN) for our senior design research paper? Thank you advance!

Figure 20: Buck converter circuit using PWM and BJT
http://www.ti.com/lit/an/snva559a/snva559a.pdf
Permission Status: **Granted**

SAME EMAIL AS FIGURE 5

Figure 21: Basic switching boost converter
http://www.learnabout-electronics.org/PSU/psu32.php
Permission Status: Requested



Do include an email contact address and we will always try to get back to you as soon as possible.
*(denotes required field)

First Name: *

Ronak

Last Name: *

Patel

E-Mail Address: *

patelronak94.ucf.edu@knights.ucf.edu

E-Mail Address again: *
Please enter your E-mail Address a second time.

patelronak94.ucf.edu@knights.ucf.edu

Organisation/School/College/ Company

University of Central Florida

Country

US

Message: *

Hello,

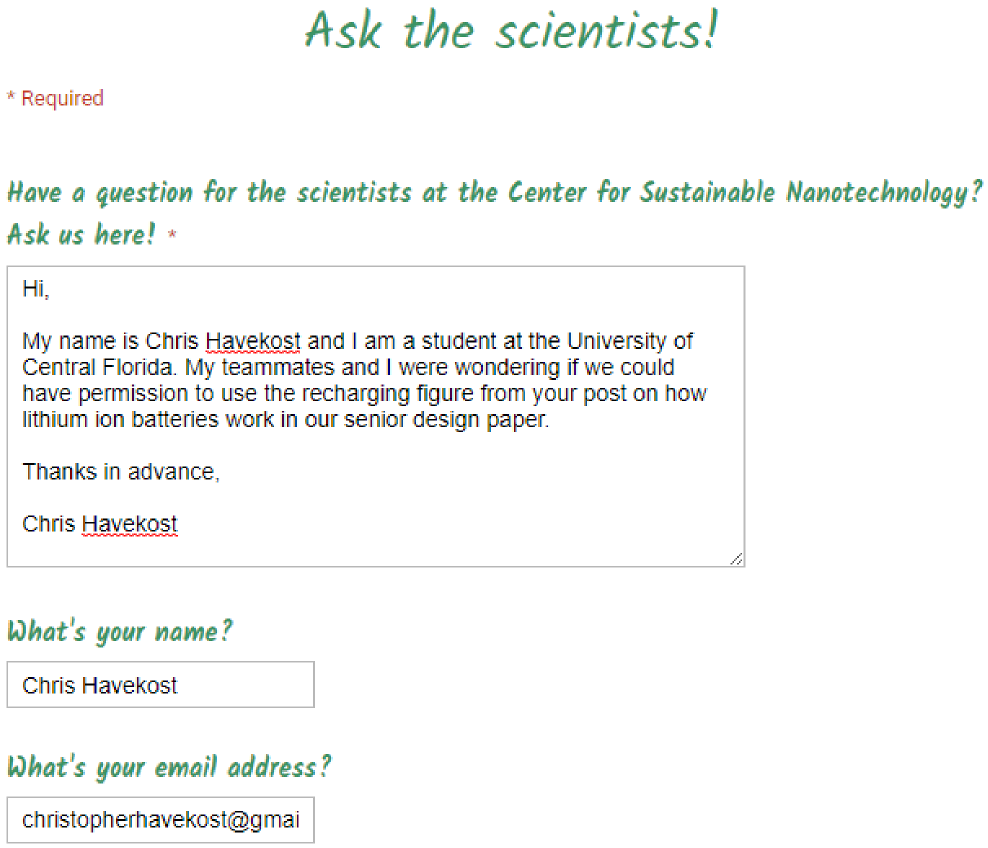
My name is Ronak Patel, a student at University of Central Florida. In my Senior Design class, my team and I are trying to gather information about electronics, so far we are only documenting our plan for this project.

I would like to ask http://www.learnabout-electronics.org if my team can use the images from the website on the presentation and document.

CAPTCHA Code: *

5F2D

Figure 23: Activity in Lithium-ion battery while powering device
http://sustainable-nano.com/2013/10/15/how-do-lithium-ion-batteries-work/
Permission Status: Requested



Ask the scientists!

* Required

Have a question for the scientists at the Center for Sustainable Nanotechnology? Ask us here! *

Hi,

My name is Chris Havekost and I am a student at the University of Central Florida. My teammates and I were wondering if we could have permission to use the recharging figure from your post on how lithium ion batteries work in our senior design paper.

Thanks in advance,

Chris Havekost

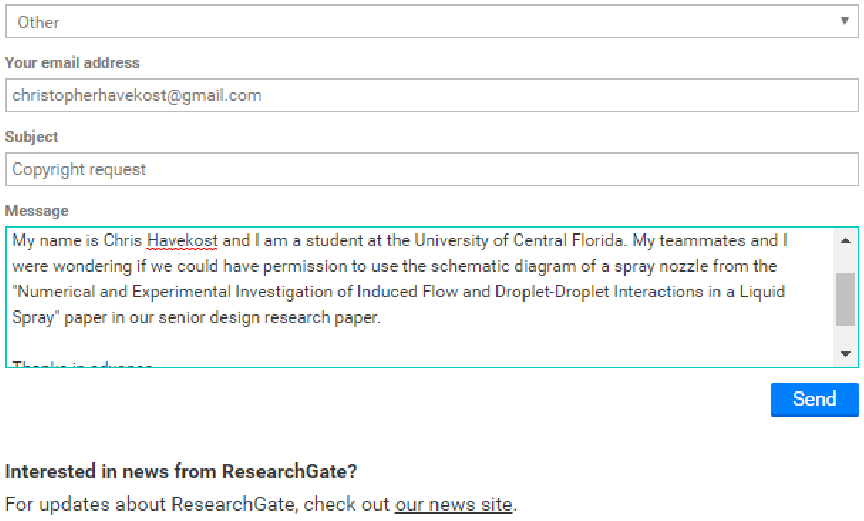
What's your name?

Chris Havekost

What's your email address?

christopherhavekost@gmai

Figure 25: Demonstration of direction of spray and angle
https://www.researchgate.net/profile/Johan_Padding/publication/281436185/figure/fig1/AS:284638355443714@1444874480149/Fig-1-Schematic-diagram-of-the-system-Left-a-pressure-swirl-nozzle-atomizer-produces.png
Permission Status: Requested



Other ▾

Your email address

christopherhavekost@gmail.com

Subject

Copyright request

Message

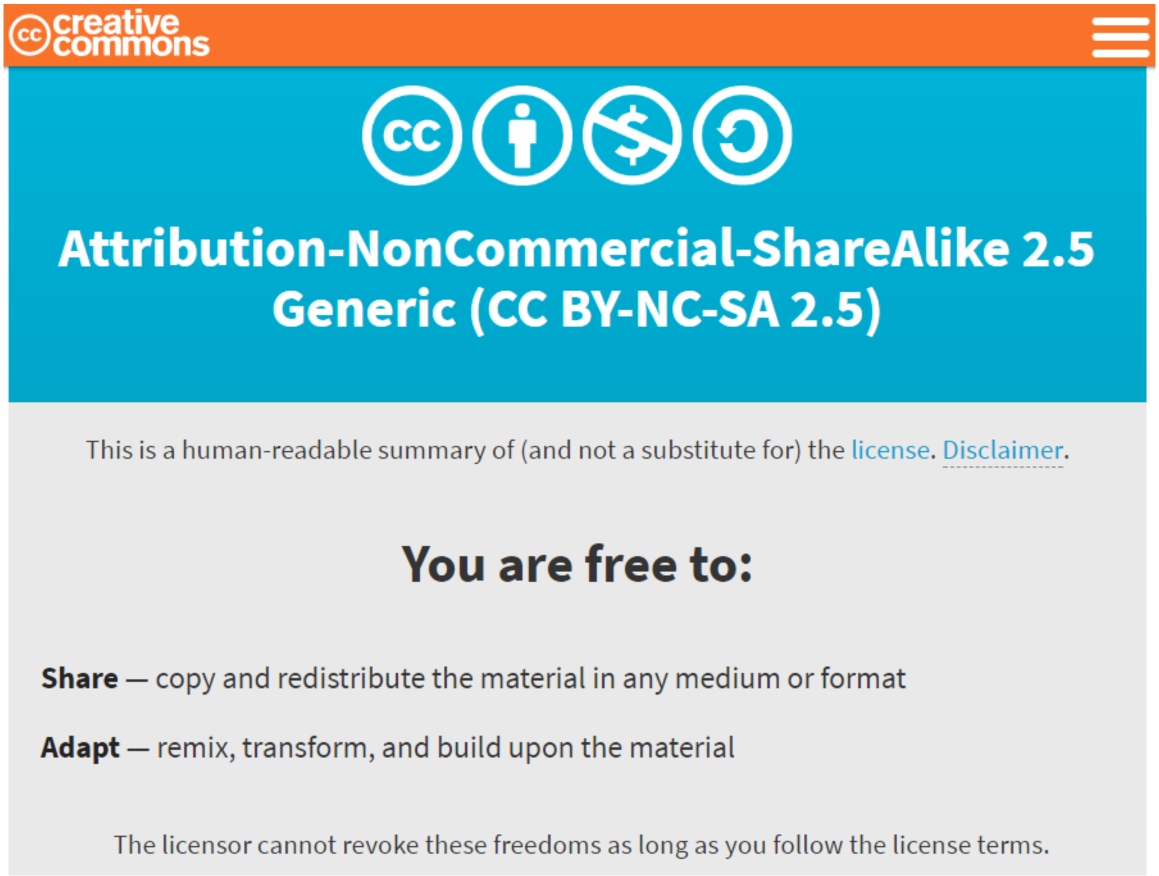
My name is Chris Havekost and I am a student at the University of Central Florida. My teammates and I were wondering if we could have permission to use the schematic diagram of a spray nozzle from the "Numerical and Experimental Investigation of Induced Flow and Droplet-Droplet Interactions in a Liquid Spray" paper in our senior design research paper.

Thanks in advance

Interested in news from ResearchGate?

For updates about ResearchGate, check out our news site.

Figure 26: Reciprocating pump
http://www.instructables.com/id/Simple-reciprocating-pump/
Permission Status: **Granted**



Attribution-NonCommercial-ShareAlike 2.5 Generic (CC BY-NC-SA 2.5)

This is a human-readable summary of (and not a substitute for) the license. Disclaimer.

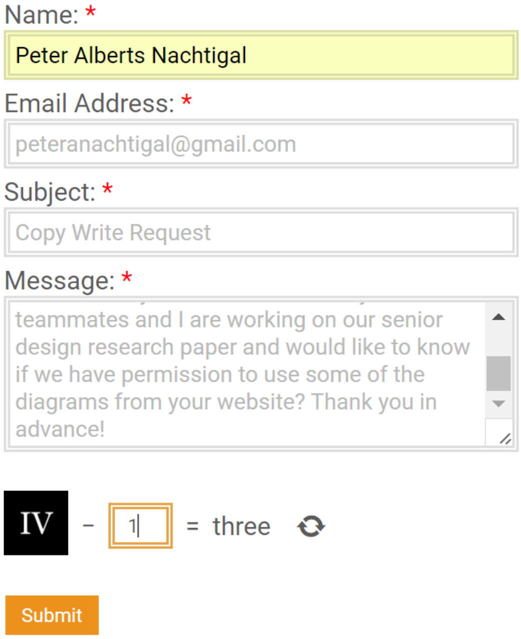
You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

Figure 27: Basic vane pump
http://camblab.info/wp/wp-content/uploads/2015/01/Rotary-Vane-Pump-Operation.jpg
Permission Status: Requested



Name: *
Peter Alberts Nachtigal

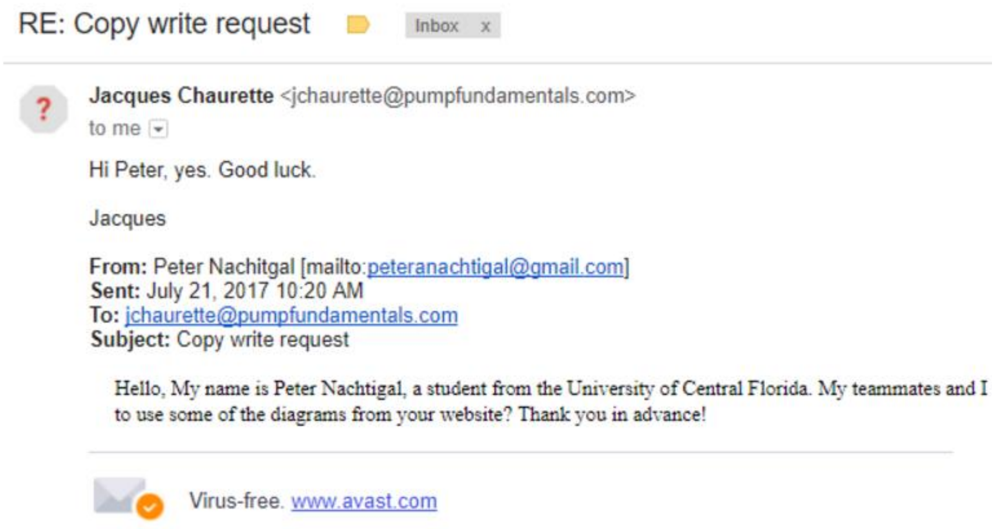
Email Address: *
peteranachtigal@gmail.com

Subject: *
Copy Write Request

Message: *
teammates and I are working on our senior design research paper and would like to know if we have permission to use some of the diagrams from your website? Thank you in advance!

IV − 1 = three

Figure 28: Impeller motor diagram
http://www.pumpfundamentals.com/pump_glossary.htm
Permission Status: **Granted**



RE: Copy write request

Jacques Chaurette <jchaurette@pumpfundamentals.com>
to me

Hi Peter, yes. Good luck.

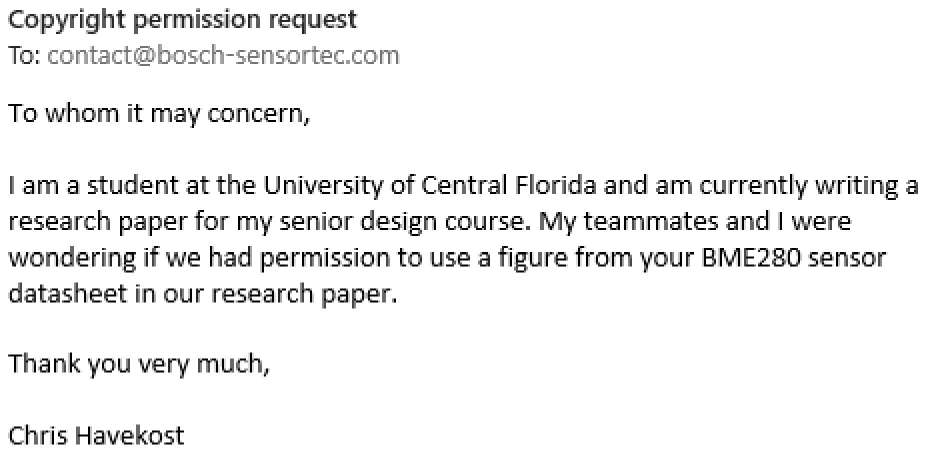
Jacques

From: Peter Nachtigal [mailto:peteranachtigal@gmail.com]
Sent: July 21, 2017 10:20 AM
To: jchaurette@pumpfundamentals.com
Subject: Copy write request

Hello, My name is Peter Nachtigal, a student from the University of Central Florida. My teammates and I
to use some of the diagrams from your website? Thank you in advance!

Virus-free. www.avast.com

Figure 29: Soldering sections of BME 280
https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280_DS001-11.pdf
Permission Status: Requested



Copyright permission request
To: contact@bosch-sensortec.com

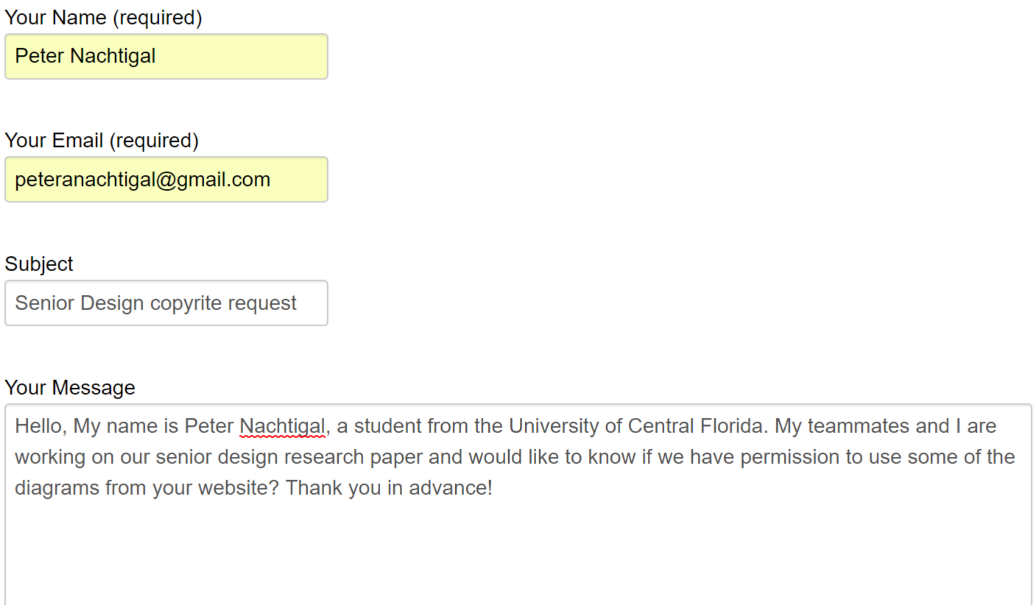
To whom it may concern,

I am a student at the University of Central Florida and am currently writing a research paper for my senior design course. My teammates and I were wondering if we had permission to use a figure from your BME280 sensor datasheet in our research paper.

Thank you very much,

Chris Havekost

Figure 39: Best angle for solar panels around the world
http://solarpanelsphotovoltaic.net/find-best-solar-panel-angle-tilt-angle/
Permission Status: Requested



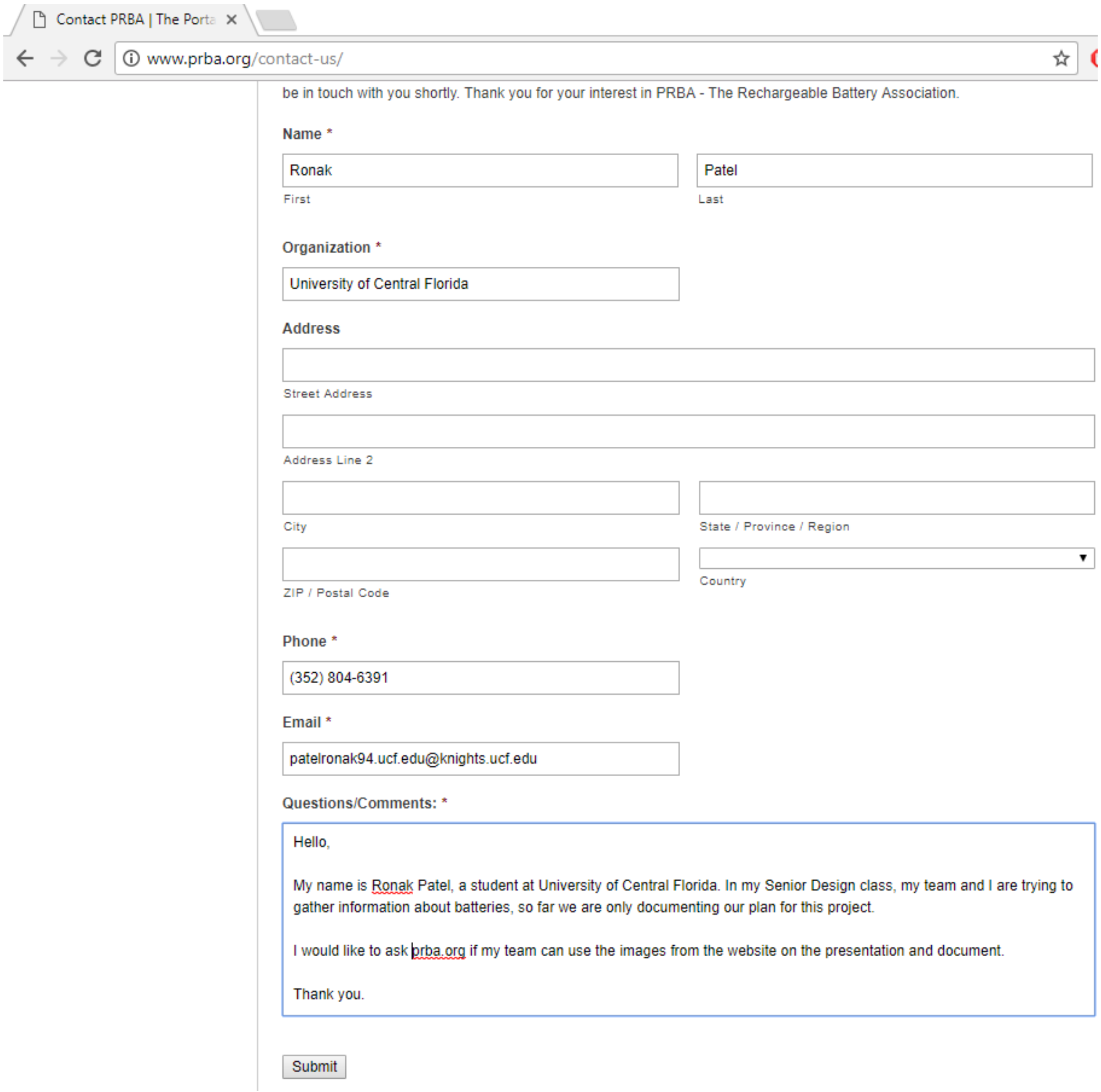
Your Name (required)
Peter Nachtigal

Your Email (required)
peteranachtigal@gmail.com

Subject
Senior Design copyrite request

Your Message
Hello, My name is Peter Nachtigal, a student from the University of Central Florida. My teammates and I are working on our senior design research paper and would like to know if we have permission to use some of the diagrams from your website? Thank you in advance!

Figure 47: Series of UL Test
http://www.prba.org/wp-content/uploads/Exponent_Report_for_NFPA_-_20111.pdf
Permission Status: Requested



be in touch with you shortly. Thank you for your interest in PRBA - The Rechargeable Battery Association.

Name *

Ronak
First

Patel
Last

Organization *

University of Central Florida

Address

Street Address

Address Line 2

City

State / Province / Region

ZIP / Postal Code

Country

Phone *

(352) 804-6391

Email *

patelronak94.ucf.edu@knights.ucf.edu

Questions/Comments: *

Hello,

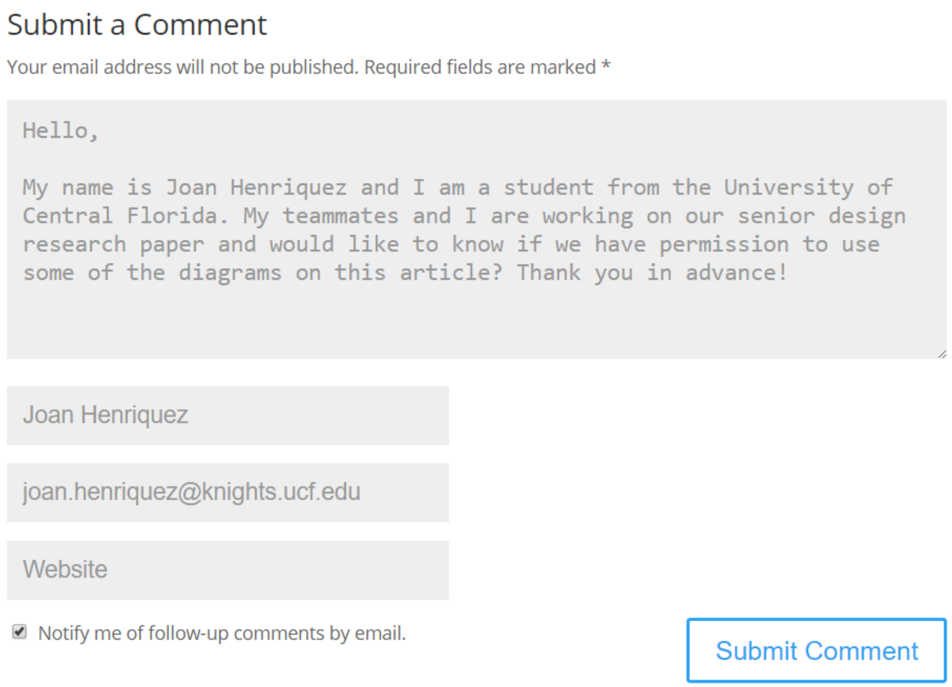
My name is Ronak Patel, a student at University of Central Florida. In my Senior Design class, my team and I are trying to gather information about batteries, so far we are only documenting our plan for this project.

I would like to ask prba.org if my team can use the images from the website on the presentation and document.

Thank you.

Submit

Figure 48: Battery tester layout
https://arduinomylifeup.com/arduino-battery-tester/
Permission Status: Requested


Submit a Comment

Your email address will not be published. Required fields are marked *

Hello,

My name is Joan Henriquez and I am a student from the University of Central Florida. My teammates and I are working on our senior design research paper and would like to know if we have permission to use some of the diagrams on this article? Thank you in advance!

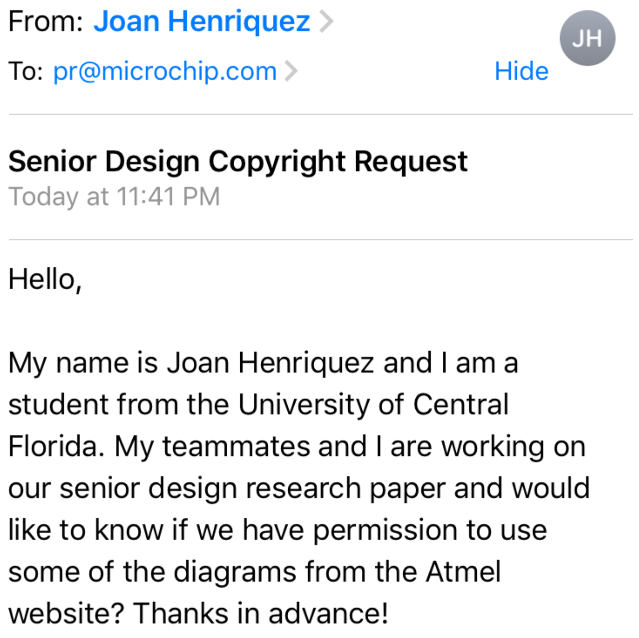
Joan Henriquez

joan.henriquez@knights.ucf.edu

Website

☑ Notify me of follow-up comments by email.

Figure 50: ADC on the Atmega
Figure 51: Atmega electrical characteristics
http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf
Permission Status: Requested


From: Joan Henriquez

To: pr@microchip.com

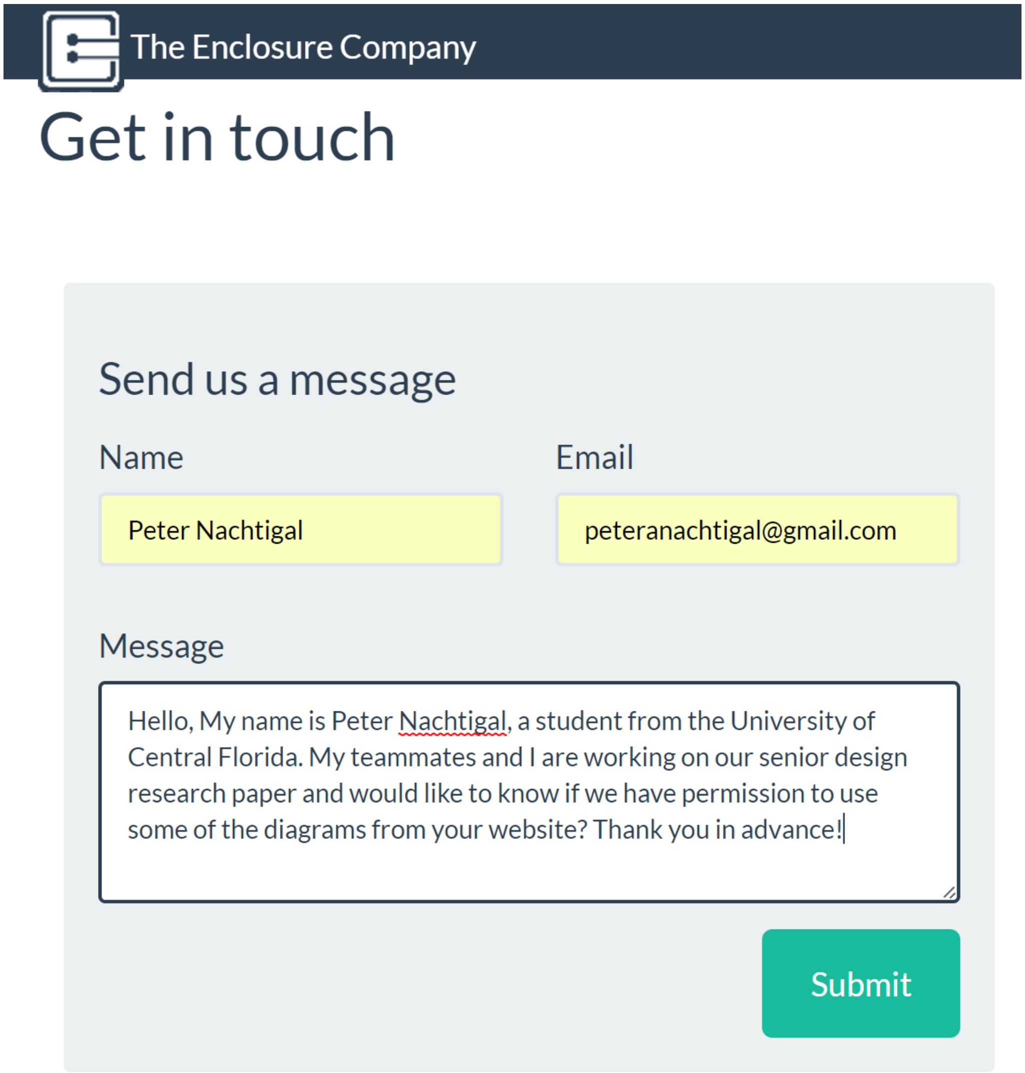
Senior Design Copyright Request

Today at 11:41 PM

Hello,

My name is Joan Henriquez and I am a student from the University of Central Florida. My teammates and I are working on our senior design research paper and would like to know if we have permission to use some of the diagrams from the Atmel website? Thanks in advance!

Table 32: IP ratings for liquid safety
http://www.enclosurecompany.com/contact.php
Permission Status: Requested



Get in touch

Send us a message

Name: Peter Nachtigal

Email: peteranachtigal@gmail.com

Message: Hello, My name is Peter Nachtigal, a student from the University of Central Florida. My teammates and I are working on our senior design research paper and would like to know if we have permission to use some of the diagrams from your website? Thank you in advance!

Appendix B – Citations

[1] http://www.atmel.com/technologies/lowpower/default.aspx

[2] https://learn.sparkfun.com/tutorials/serial-communication

[3] https://www.renesas.com/en-us/support/technical-resources/engineer-school/mcu-programming-peripherals-01-gpio.html

[4] https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/risccisc/

[5] https://sensing.honeywell.com/honeywell-sensing-humidity-sensors-line-guide-009034-7-EN.pdf

[6] http://ccm.net/faq/298-what-is-wifi-and-how-does-it-work

[7] https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works

[8] http://www.zigbee.org/what-is-zigbee/

[9] https://www.lifewire.com/types-of-voltage-regulators-818851

[10] http://www.ti.com/lit/an/snva559a/snva559a.pdf

[11] https://www.digikey.com/en/articles/techzone/2013/apr/generating-high-dc-output-voltage-from-low-input-supply

[12] http://www.explainthatstuff.com/howrelayswork.html

[13] http://www.explainthatstuff.com/pumpcompressor.html

[14] http://www.pumpsandsystems.com/topics/pumps/reciprocating-pumps/reciprocating-pumps-vs-multi-stage-centrifugal-pumps

[15] http://gccegov.org/documents/10179/276020/Java_standards_v1.0_%28NEAF%29.pdf/acd4a9dd-1ff4-4ba6-8b48-f6f5e953edd

[16] https://nepp.nasa.gov/docuploads/06AA01BA-FC7E-4094-AE829CE371A7B05D/NASA-STD-8739.3.pdf

[17] http://safety.eng.cam.ac.uk/procedures/Soldering/soldering-safety

[18] http://www.softwaretestingstandard.org/

[19] http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf