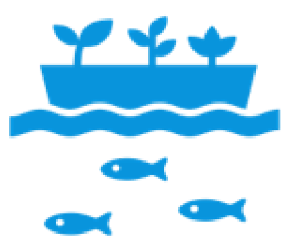
AquaSense

MENG-957-T: Smart Aquaponics System with Sensors and Web-Based Feedback



By Christy Koh and Joshua Koh

Project Abstract

AquaSense was developed to help farmers and hobbyists rear fish and grow plants hydroponically in a symbiotic system, and to easily and continuously monitor pH, water level, temperature, and light. An aquaponics system was built and integrated with a multi-sensor monitoring system with web-based feedback and alerts. An existing fish pond was expanded to grow plants through the design, construction, and refinement of growth beds, bell siphons, and greenhouses. The growth beds drained and filled cyclically, nourishing the plants with nutrient-rich fish waste water and acting as a biofilter to purify the water for the fish. A microcontroller-based monitoring system, using Raspberry Pi, collected data from multiple sensors: a pH probe, ultrasonic sensors, thermistors, temperature transducers, and a photocell. This data received from the sensors was streamed to a Node.js/Express web server and graphically displayed in real-time. Incorporating user-defined acceptable ranges of each measured parameter, the program checked if the readings exceeded this range and alerted the user via email or text. The sensor system was waterproofed so that it could operate even under rainy conditions. With a cost of $113.60, the sensor system is affordable for farmers and hobbyists. It can be implemented on top of existing aquaponic systems for continuous monitoring and enables rapid response to unfavorable conditions, thus helping to maximize food production. Since AquaSense requires minimal maintenance on the user's part, the complete system could be implemented nearly anywhere for individuals to become more self-sufficient and/or boost their diet and income.

Table of Contents

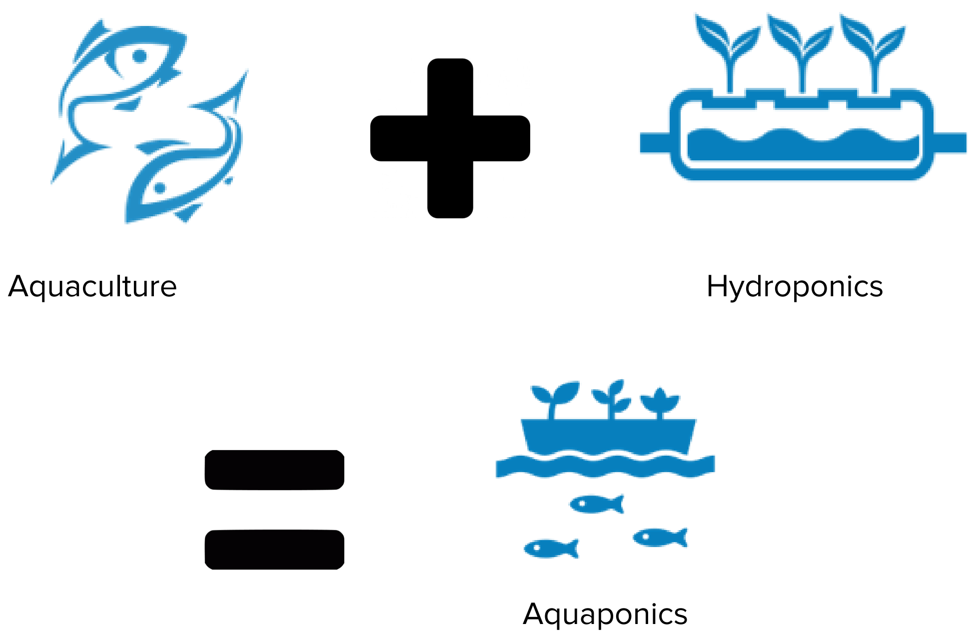
Planning

1. Background Research

Aquaponics combine aquaculture (raising fish) with hydroponics. Aquaponic systems are used by both hobbyists and farmers to remove harmful nitrates and nitrites in fish waste water while also supplying nutrients to the plants.

In a purely aquacultural system, the fish excrete ammonia as waste. Bacteria in the water convert the ammonia to nitrates or nitrites, which will eventually build up in the system to toxic levels. At this point, a water change is necessary to remove some of the nitrates and nitrites from the system. In a hydroponic system, nutrient-enriched water is used to cultivate plants, but it is necessary to supply those nutrients to the system by dissolving fertilizer in water.

Integrating an aquacultural system with a hydroponic system solves both problems faced by the individual systems; the plants act as a biofilter for the fish waste, which supplies the necessary nutrients for the plants to grow. The entire aquaponic system makes use of the mutually beneficial systems to yield both organic produce and fish.



Planning

2. Engineering Problem

An aquaponics system is very complex as conditions for the proper growth of both fish and plants must be monitored constantly and maintained for optimized growth. For example, the pH of the water should remain between 6.5 and 7.5, and plants require a certain temperature or must be protected from frost. In addition, it is also important to monitor the water level of the grow beds and the fish tank to make sure the aquaponic system is running smoothly.

As a result, monitoring aquaponic systems often involves measuring conditions such as pH, water level and temperature. However, these measurements are often taken manually, which is time- and effort-consuming. Many commercially available systems offer a solution to this problem, but they are not affordable for a homestead farmer.

In order to solve the problems of cost and convenience, we will design and build an aquaponic system that is not only affordable but also able to monitor conditions using a Raspberry Pi, and display the data on a web server.

3. Engineering Goal

The design and construction of an affordable and effective aquaponic system with integrated sensor feedback (water level, pH, and temperature) streamed onto a web server for hobbyists and fish farmers to easily and continuously monitor water conditions.

Planning

4. Engineering Plan

We will construct an aquaponic system on top of our existing pond and filter system by building grow beds. These beds will fill and drain cyclically using a bell siphon. This setup will be used to develop and test the monitoring system in a realistic environment. We will program a Raspberry Pi to receive data from temperature, pH, and ultrasonic sensors and push it to an online web server, where it will be displayed graphically for the user's convenient viewing. We will program a function that allows the user to create presets that define each condition's "danger" levels.

5. Criteria

- Aquaponic system maintains suitable growing conditions for plants and fish (temperature, pH).
- Monitoring system senses temperature, pH, and water level.
- Sensor data is streamed onto a web server.
- User must be able to define the range of acceptable temperature, pH, and water level readings.
- System will send an alert to the user if readings fall outside the acceptable range.

6. Constraints

- Waterproofing - Raspberry Pi and all electrical components must be protected from rainfall and other forms of water damage
- Wiring - Wires must be short enough to minimize interference with signals
- Accessibility - Plants must be accessible for planting, pruning, and harvesting
- Time - Abstract for Science Fair due March 8
- Cost - System should be affordable for hobbyists and fish farmers ($100-$150)

Planning

7. Design Features

1. An ebb-and-flow aquaponic system consisting of two growing beds and one fish pond.
2. A cost-effective microcontroller-based aquaponic monitoring system which enables extensive real-time monitoring of pH, temperature, and water level.
3. A web-based solution that allows real-time viewing of environmental conditions over the Internet. It will send an alert to the user if conditions reach user-set ‘danger’ levels.
4. A greenhouse which maximizes plant growth throughout the seasons.

8. Materials List

Aquaponic System

- Fish waste water
- Pre-existing filter and pump
- ¾”, 2”, and 3” PVC pipes and connectors
- Hydroton Expanded Clay growing media
- 2 containers that can function as aquaponic beds
- 2-mil plastic sheet
- 8ft 2x4 Wood
- 4-Ply Fir Plywood
- Green Onion plants

Monitoring System

- Raspberry Pi
- MCP3008 Analog to Digital Converter
- OPA2376 Operational Amplifier
- Neptune pH Probe and calibration solutions
- NCP15XH103J03RC Thermistor
- AD592 Temperature Transducer
- Ultrasonic Sensor
- Silicone Sealant
- Ultrasound sensor
- Wires
- Breadboard
- Container
- Photocell Light Sensor

Tools

- Dremel
- Sandpaper
- Pipe cutter
- Drill
- Soldering Iron
- Wire Stripper
- Crimping Tool
- Computer/Laptop
- Gloves
- Safety Glasses

Building

9. Construction of the System (Growth Beds)

The most important part of an ebb-and flow system is the siphon. The siphon controls the filling and draining of the grow bed, and thus how often nutrients are delivered to the plant's roots. We chose to use a bell siphon because it was simple and effective. To build the drainpipe, we used a 1 inch PVC pipe connected to a 2 inch adapter, which is fitted into the newly cut hole. A 1.5 inch threaded adapter at the top of the pipe promotes faster flow due to size reduction. On the other end of the hole, two pieces of pipe are connected via a 90° elbow connector. This directs the outflow back into the fish tank. This will be covered by a capped 2 inch PVC pipe with slots cut into it. This cover, in turn, will be covered by a black 3 inch PVC pipe with holes in it. This outside pipe shields the siphon so that the growth media will not go through and clog the bell siphon. Each drainpipe was attached to the two 30.88” × 20.31” × 14.55” black totes (containers) using ¾” bulkheads.

We purchased a 42” × 66” board of 0.5 inch thick 4-ply fir plywood. Each growth bed was stationed on each side of the board. We traced the outlines of the two growth beds onto the pieces of wood and drilled 2.5 inch diameter holes into both the bottom of the growth beds and into the piece of wood.

Building

9. Construction of the system (Growth Beds)

3 lengths of 2x4 wood were nailed onto the board for support. The entire structure was mounted on a stack of concrete blocks so that the grow beds would be elevated higher than the pond.

Building

9. Construction of the system (Bell Siphon, Growth Medium)

From left to right: 3 inch cover, 2 inch cover, 1 inch drain pipe. Each will be stacked on top of the other.

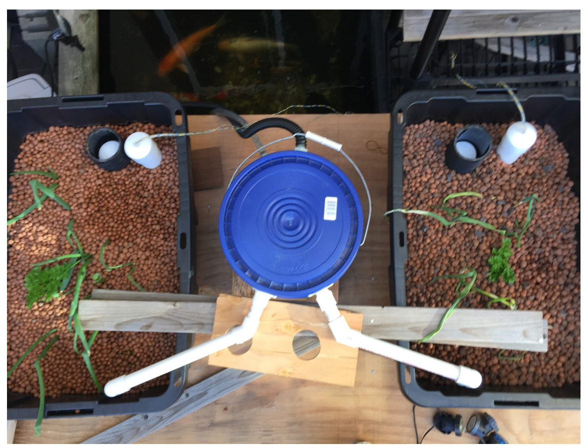
Using our pre existing filter, we directed the flow of filtered water into both growth beds by placing the filter in between both beds. We also directed the drainage back into the pond. In order to test the siphon, we allowed the system to run with empty grow beds for several days.

Building

9. Construction of the system (Growth Medium, Plants)

We selected Hydroton Expanded Clay as our growth medium. The round shape of the clay pebbles would allow plenty of space for the roots to grow while minimizing root damage when the medium is shifted around. The small imperfections on the surface of the clay also allowed for the retention of moisture. After filling both the growth beds with Hydroton, we planted green onions (*Allium Cepa*), wheatgrass, as well as cut lettuce and celery bases.

The plants that fared the best were the green onions and the celery.

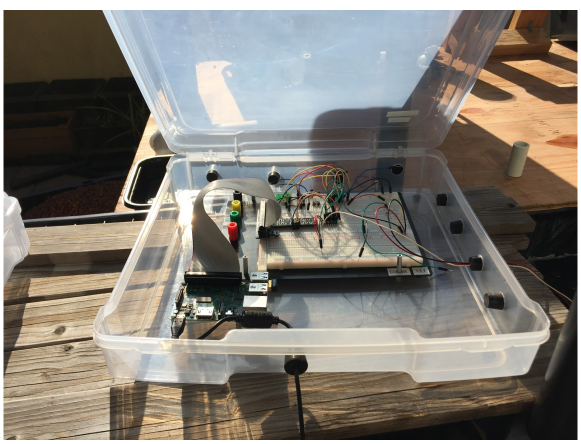


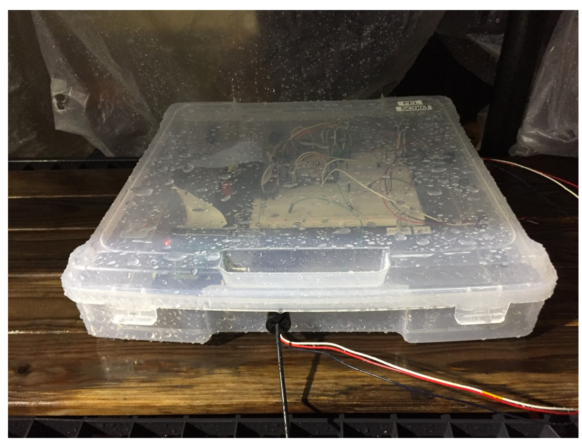
Building

9. Construction of the system (Greenhouse and Waterproofing)

After this was completed, we constructed the greenhouse. The frame of the greenhouse was put together with ¾” PVC pipes. The frame is then wrapped in a translucent sheet. We cut out a hole on the side to allow the filter outflow into the growth bed.

All that was left after this step was to waterproof the container in which the Pi and the wiring would be contained. We achieved this by cutting holes into the sides of a container and filling those holes with rubber stoppers. These plugs have holes drilled into them so that wires can be fitted into the circuit without compromising the watertight seal.





Building

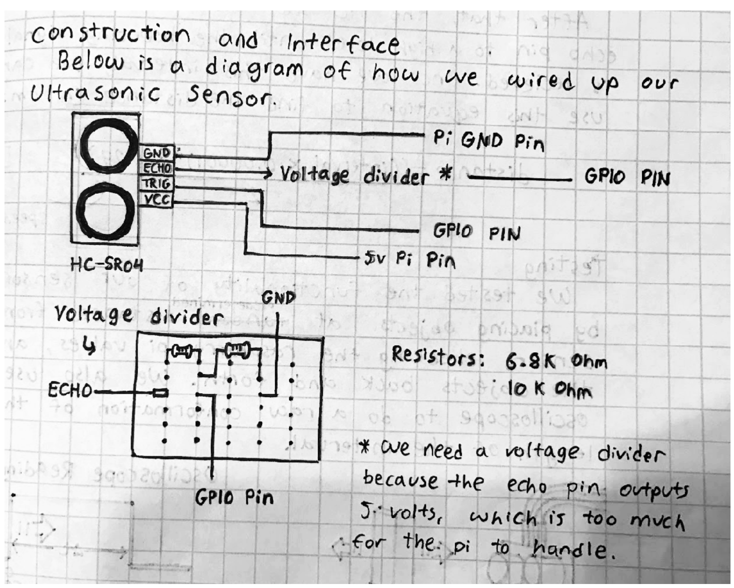
10. Implementation of Sensors (Water Level Sensor)

The first sensor that we implemented was the water level sensor, which was actually a HC-SR04 Ultrasonic Sensor. This sensor has 4 pins: Ground, Echo, Trigger, and VCC. Ground is connected to the RPI's ground, and VCC has to be connected to 5 volts. To operate the sensor, a 1000ms duration pulse must be sent through the Trigger Pin. This causes the sensor to activate, sending ultrasonic pulses out of the transmitter and sensing pulses reflected from an object with its receiver. The Echo pin sends back a signal to the RPI whereby the pulse duration is proportional to the 2 times the distance to the object. A python program is then used to measures the duration of the pulse sent from the sensor to the RPI. To convert the pulse duration into distance between the sensor and the object, the following equation must be used:

$$\frac{(Signal \times 0.0001) \times 340.27}{2} = distance(cm) \,,$$

where 340.27m/s is the speed of sound.

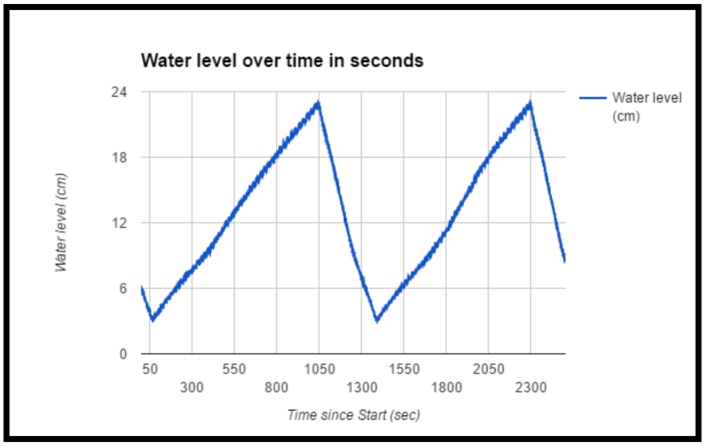
However, the Echo maximum pulse signal that is received by the RPI will be 5 volts. Since the Raspberry Pi can only handle 3.3 Volts, a voltage divide+r is required to make sure that the maximum voltage returned to the RPI is scale to an acceptable level of 3.3 Volts. The voltage divider used for this purpose is shown below:



Building

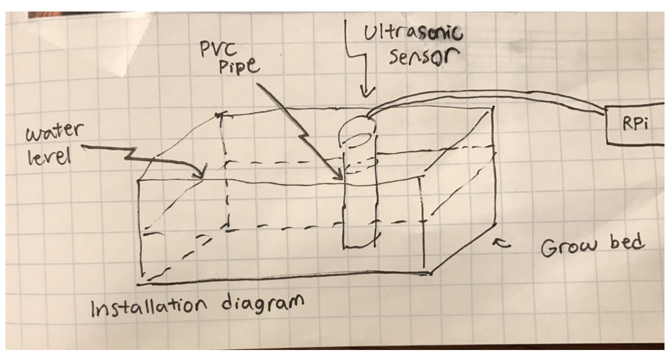
10. Implementation of Sensors (Water Level Sensor)

With this in place, we proceeded to begin our data collection. We measured the water levels in the growth beds for about 40 minutes, then took the collected data and graphed it. Our data is as follows:



As you can see, the general direction of the water level goes through one cycle every 14.9 minutes. This data was particularly useful when comparing our past to present iterations. For example, we can compare this data to our present readings, and then analyse what may be causing a difference between the two.

We then installed the sensors into the growth beds through a second capped PVC pipe. These PVC pipes have holes in them so that wires can be connected back to the RPI. The water level sensor is placed in the cap and protected by a piece of wood with silicone waterproofing. A diagram and picture of our method are shown.



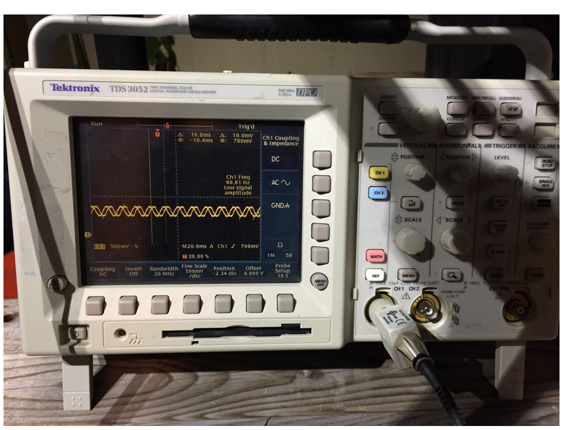
Building

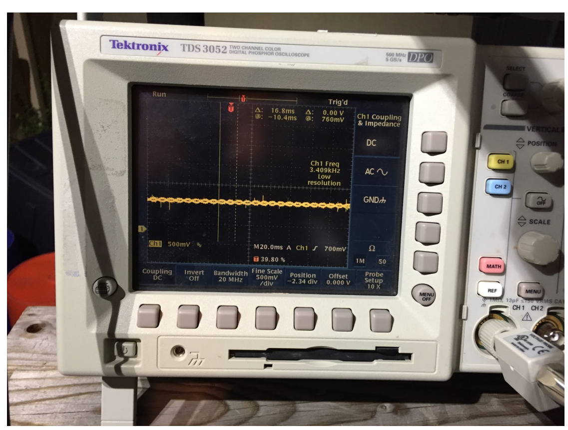
10. Implementation of Sensors (pH sensor)

A Neptune pH probe was used to measure pH. The sensor works with a special glass bulb which allows ion exchange across it. When the probe is immersed in an acidic solution, it will produce a positive voltage. Conversely, when the probe is immersed in a basic solution, it will produce a negative voltage. The approximate difference in voltage between each step in pH is 60mV. Since this value is too small to be accurately read and interpreted by the Raspberry Pi, we decided we needed to amplify the result by increasing gain. We also needed to turn the indirect relationship (voltage decreases as pH increases) into a direct relationship (voltage increases as pH increases). We solved these issues by using 3 operational amplifiers (op-amps). The circuit we wired (shown below) was modified from one mentioned in the Texas Instrument Application Note for AN-1852.

The first op-amp is a voltage follower (gain = 1) which offsets the voltage at 1.65V, half of the 3.3V that the GPIO pin can measure. The second op-amp is a voltage follower to buffer the voltage generated by the pH probe, which does not have to ability to drive sufficient current. The last op-amp inverts the voltages from op-amp 2 so that a higher voltage output corresponds to a higher pH.

A challenge we faced was that the readings would fluctuate once the pH probe was placed in the water, yielding very inconsistent pH readings. Using an oscilloscope to determine the root cause, we discovered that it was the pump that was causing the fluctuations. The frequency of the interference signal was about 60 hertz. The picture on the left shows the interference stigma when the pH probe is submerged. The picture of the oscilloscope on the right displays the signal when the pump is turned off.





Building

10. Implementation of Sensors (pH sensor)

In order to make the signal more stable, we sought to isolate the pH sensor from the other sensors three ways: powering the Raspberry Pi from a different outlet, powering it using rechargeable batteries, and implementing an optocoupler. The above solutions did not eliminate the fluctuation in the signal, and we concluded that the signal was being conducted through the water itself. A low-pass filter was incorporated into the circuit after the third op-amp to filter signals with a frequency higher than 60 hertz. This final approach stabilized our output signal.

Calibration

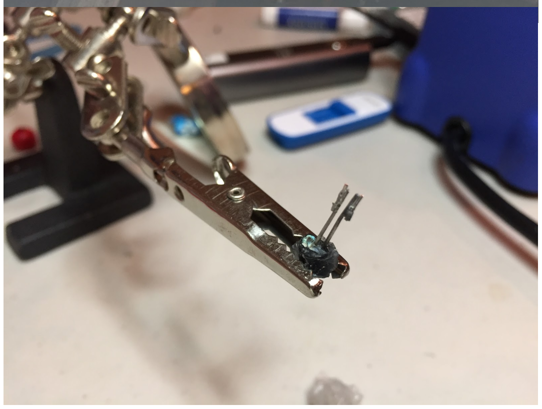
In order to interpret the signal, we used the equation:

$$pH = 0.3908 \times 1.024$$

Building

10. Implementation of Sensors (Temperature Sensor)

When placing the temperature sensor directly in water, we discovered that it was very difficult to completely seal the sensor with silicone sealant. After running the temperature sensor continuously for a few weeks, we discovered that the copper connectors became corroded. the sensor was also shorting the pH sensor because it was not completely sealed. As a result, we decided to waterproof the sensor more completely using a glass test tube.

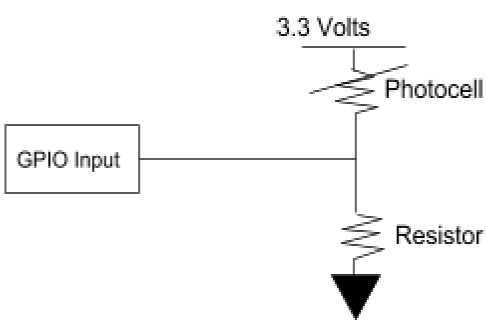


Building

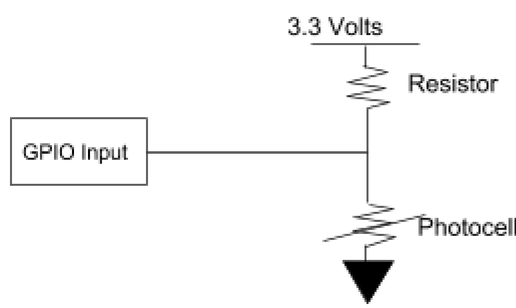
10. Implementation of Sensors (Light Sensor)

The last sensor we implemented was the light sensor. The light sensor is a radioshack Cadmium-Sulfide photocell. It works by changing its resistance based on how much light it senses - if there is more light, it becomes less resistant. If there is less light, it becomes more resistant. We can measure this voltage, and use the corresponding voltage to determine the level of brightness outside.

This brings up a problem similar to the one faced when implementing the water level sensor. Since the RPI has to read from a pin, we have to make sure that the voltage can be no more than 3.3 Volts. We decided to create a voltage divider, which had two possible configurations:



Configuration #1



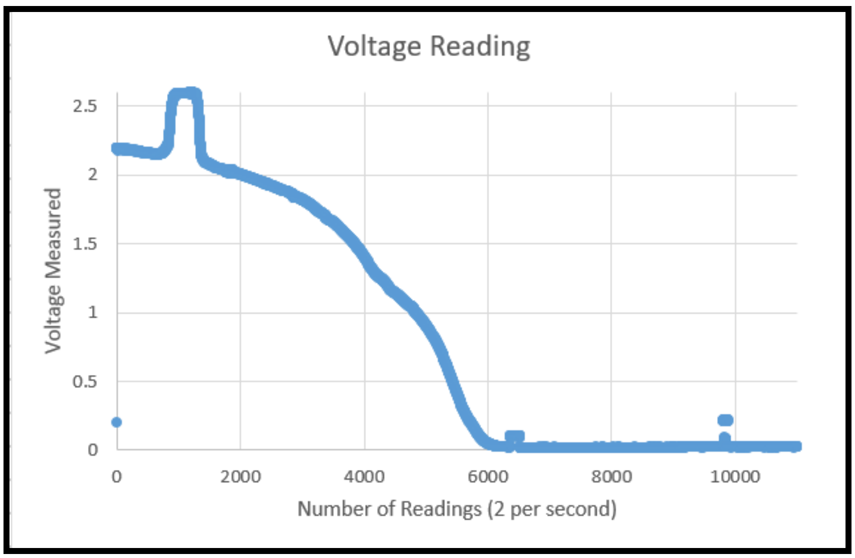
Configuration #2

Each configuration has a different equation that corresponds to the voltage outputted. The two equations are derived below:

let r be the rating of the resistor and rcell be the rating of the photocell
vin is the input that the pi reads

Configuration #1	Configuration #2
$V = IR$ Ohm's law	$V = IR$ Ohm's law
$V' = I(R + Rcell)$	$V' = I(R + Rcell)$
$I = \frac{R+Rcell}{V'}$	$I = \frac{R+Rcell}{V'}$
$vin = (\frac{V'}{R+Rcell}) \times R$	$vin = (\frac{V'}{R+Rcell}) \times Rcell$
$vin = (\frac{R}{R+Rcell}) \times V'$	$vin = (\frac{Rcell}{R+Rcell}) \times V'$

Either equation can be used to solve for the voltage, so we chose the first equation. After wiring up the sensor, we ran the program with the equation and collected data from the afternoon through the evening.



At the peak of the afternoon, the sensor read a 2.6 Voltage. After that, the sun started to set, and the photocell recorded the change in brightness. This showed us that our sensor was fully functional and that we were finally done with all of our sensor integration.

Project Results

11. Cost Evaluation and Comparison to Products on the Market

The cost of our sensor system totaled $113.60, falling well within our target range of $100-$150.

http://www.osmobot.com/

https://www.cooking-hacks.com/open-garden-hydroponics

https://incredibleaquagarden.co.uk/more-info/aquaponics/about-the-aquaponics-control-system/

12. Conclusion and Ideas for Future Improvement

AquaSense was developed to help farmers and hobbyists rear fish and grow plants hydroponically in a symbiotic aquaponic system, and to easily and continuously monitor pH, water level, temperature, and light. We constructed growth beds and bell siphons as an extension of our existing pond to make a basic aquaponic system. Sensor feedback was integrated into the system using a Raspberry Pi microcontroller, which helped us to monitor the conditions in the growth beds and the fish tank. As we constructed the system, we encountered different challenges. For example, we observed that the pH probe's output voltage would fluctuate when it was submerged in the pond because of interference caused by the pump. We solved the problem by implementing a low-pass filter which eliminated the AC interference. A web server using Node.js streamed the data graphically; it also took into account user-defined limits and alerted the user via text or email when the parameters exceeded those limits. The final product met our design criteria of maintenance of conditions, data collection, a web interface, user-defined limits, and alerts based on those limits. With a total cost of $113.60, the sensor system is affordable for homestead farmers and hobbyists. To extend this project, we plan to install a solenoid on the bell siphons in order to ensure that the water lock breaks reliably. We also plan to add a pressure and humidity sensor to the sensor system.

Project Results

12. Conclusion and Ideas for Future Improvement

- Add light control to maximize plant growth at night.
- Integrate a Pressure and Humidity Sensor (BME280)
- Store data in a database (MongoDB).
- Develop a mobile app.

13. Acknowledgements

Special thanks to Ping-Chiek Koh for advice and mentoring, as well as for funding purchase of the components of the system.

14. Bibliography

[1]*"AN-1852 Designing with pH Electrodes".* Texas Instruments. April 2013.

[2]"*NTC Thermistors".* muRata Datasheet. Oct 6, 2016.

[3]*"AD592 - Low Cost, Precision IC Temperature Transducer.*

[4]"*MCP3004/3008 2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface"* Microchip Datasheet. 2008.

[5]"Fish Pond Water Quality: As Simple as Chemistry 101". Noble Foundation. Jul 1, 2009