FLOATING FRESH AND SALT WATER TEMPERATURE AND PH SENSOR APPARATUS

by

Arthi Iyer

Grade 10

# ABSTRACT

As a tenth grader in California, I took on the challenge of familiarizing myself with Arduino software and hardware development by constructing a floating structure that could take continuous readings for the pH and temperature of water. I was successful in creating the hardware and software necessary to achieve this after a few months of learning and making mistakes.

TABLE OF CONTENTS

ABSTRACT 2

2.0 INTRODUCTION 4

3.0 PROTOTYPING PROCESS 5

3.1 MATERIALS 5

3.2 CIRCUIT DIAGRAM 6

3.3 PROTOTYPE 1 7

3.4 PROTOTYPE 2 7

4.0 EXPERIMENTAL RESULTS 8

4.1 FIRST LAUNCH: AUG 15 4:35 PM 8

4.2 SECOND LAUNCH: AUG 19 10

5.0 PERSONAL CHALLENGES 10

6.0 CONCLUSION 11

7.0 REFERENCES 12

# 2.0 INTRODUCTION

In 2013, Liquid Robotics released a revolutionary product called the Wave Glider[[[1]](#endnote-1)]. It has proved to be incredibly useful in almost all marine biological research as an unmanned data collection tool[[[2]](#endnote-2)]. It is constructed as a surfboard equipped with countless sensors on top that store and transmit data that can be used for the progression of oceanic science.

As someone who is incredibly interested in marine biology and intrigued by the ocean in general, I thought the Wave Glider was utter genius. I thought about what could be done with that data and how it could help scientists everywhere.

And on top of that, other people have done similar things also. The National Oceanic and Atmospheric Administration (NOAA)[[[3]](#endnote-3)] has created a drifter mechanism that tracks ocean currents and transmits to satellites, and the International SeaKeepers Society[[[4]](#endnote-4)]has created one that measures temperature, velocity, latitude and longitude.

So, I wanted to make my own.

My goal was to take a small step to learning the electronics and functionality of the Wave Glider. To do this, I used Arduino[[[5]](#endnote-5)] because it is a known microcontroller that I deemed powerful enough for my purposes.

# 3.0 PROTOTYPING PROCESS

## 3.1 MATERIALS

* 1 Arduino board
* 1 set of stackable headers
* 1 waterproof DS18B20 digital temperature sensor[[[6]](#endnote-6)]
* 1 Vernier pH Sensor[[[7]](#endnote-7)]
* 1 Sparkfun Vernier Interface Shield[[[8]](#endnote-8)]
* 1 Sparkfun MicroSD Shield[[[9]](#endnote-9)]
* 1 breadboard
* 1 battery pack ≥5V
* 2 white wires
* 2 red wires
* 2 black wires
* Duct tape
* A jar large enough to fit the electronics inside with a plastic lid
* Hand drill
* A drill bit with at least a 15mm diameter

## 3.2 CIRCUIT DIAGRAM

Macintosh HD:Users:ArthiIyer:Desktop:Sensor_Proj_Schematic*_schem.pdf

## 3.3 PROTOTYPE 1

The first step I took towards diving into this whole project was to figure out how to hook up temperature sensors and pH sensors to an Arduino board. The temperature sensor required the use of a breadboard, which proved to be difficult for me to figure out, but the pH sensor was quite simple. I used a Vernier pH sensor, a common classroom science tool, so it came with its own Arduino shield.

After hooking up the sensors, I had to figure out how I was going to store the data. After some thought, I decided that an SD card was the way to go, so I got my hands on an SD card shield for Arduino from Sparkfun and set to work. After some struggling, I was able to successfully write strings to an SD card. From there, the software was quite smooth sailing.

In terms of the hardware, there were a few more twists and turns. Initially, I tried actually building a sealed box-like structure out of Plexiglas. However, after a few attempts with it, I decided that I had neither the prowess necessary with power tools, nor the proper equipment. It became clear that I was spending more time trying to build a ship than I was my actual product.

So, I fell back on the trusty option of a sealed water bottle. I ended up just putting all the hardware in a water bottle with a hole drilled into it and duct taped it to make it water tight.

Towards the end of the experiment, I tried to consolidate all my code from the pH sensor, temperature sensor, and SD card. This code amounted to more than 600 lines of code and this turned out to be an unexpected problem. The Arduino compiler produces more bytes than the Arduino controller can store in its memory after about 550 lines of code. My main challenge was optimizing the code I got form the Vernier github page without compromising its purpose.

After struggling for a good 3 hours, I found a blog[[[10]](#endnote-10)] that did it for me. After using that code instead of the official Vernier code[[[11]](#endnote-11)], my code compiled and finally worked.

## 3.4 PROTOTYPE 2

My second prototype came about after testing occurred for my first prototype, which is detailed in the section labeled ‘FIRST LAUNCH: AUG 15 4:35 PM’. This version is an exact replica in terms of electronics and is quite similar structurally as well.

However, it has a few differences. In order to put an extra line of defense against water, the electronics are submerged in sand, creating a strong insulator that absorbs the water before it has the option to harm the electronics. Of course, sand particles are also very bad for circuitry, so the electronics are then packaged and sealed into Ziploc bags using duct tape.

In order to give the device a buoy-like effect where it floats right-side up in the water instead of on its side, 3 sand bags are secured to the outside of the device to balance it in the water.

# 4.0 EXPERIMENTAL RESULTS

## 4.1 FIRST LAUNCH: AUG 15 4:35 PM

The first round of testing I completed happened with Prototype 1. It simply involved throwing the device into my family’s pool and leaving on a four-day trip.

When I returned, I was a little bit disheartened to find that the device had filled up with some water and that the electronics inside were undisputedly fried. However, I was able to salvage some of the data from the SD card inside. It is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| Temperature | pH | Time | Date |
| 26.62 | 5.79 | 16:18:40 | 13 8 2015 |
| 26.56 | 5.80 | 16:20:41 | 13 8 2015 |
| 26.19 | 5.80 | 16:22:41 | 13 8 2015 |
| 26.00 | 5.80 | 16:24:42 | 13 8 2015 |
| 26.44 | 5.71 | 16:26:43 | 13 8 2015 |
| 26.44 | 9.05 | 16:28:44 | 13 8 2015 |
| 28.19 | 9.16 | 16:30:45 | 13 8 2015 |
| 28.19 | 13.72 | 16:32:46 | 13 8 2015 |
| 28.19 | 13.72 | 16:34:47 | 13 8 2015 |
| 28.19 | 13.72 | 16:36:48 | 13 8 2015 |
| 28.12 | 13.72 | 16:38:49 | 13 8 2015 |
| 28.12 | 13.72 | 16:40:50 | 13 8 2015 |
| 28.12 | 13.72 | 16:42:51 | 13 8 2015 |
| 28.06 | 13.72 | 16:44:52 | 13 8 2015 |
| 28.12 | 13.72 | 16:46:53 | 13 8 2015 |
| 28.06 | 13.72 | 16:48:54 | 13 8 2015 |
| 28.06 | 13.72 | 16:50:56 | 13 8 2015 |
| 28.06 | 13.72 | 16:52:57 | 13 8 2015 |
| 28.06 | 13.72 | 16:54:58 | 13 8 2015 |
| 28.06 | 13.72 | 16:57:00 | 13 8 2015 |
| 28.06 | 13.72 | 16:59:01 | 13 8 2015 |
| 28.06 | 13.72 | 17:01:02 | 13 8 2015 |
| 28.06 | 13.72 | 17:03:04 | 13 8 2015 |
| 28.06 | 13.72 | 17:05:05 | 13 8 2015 |
| 28.06 | 13.72 | 17:07:07 | 13 8 2015 |
| 28.06 | 13.72 | 17:09:08 | 13 8 2015 |
| 28.06 | 13.72 | 17:11:10 | 13 8 2015 |
| 28.06 | 13.72 | 17:13:12 | 13 8 2015 |
| 28.06 | 13.72 | 17:15:13 | 13 8 2015 |
| 28.06 | 13.72 | 17:17:15 | 13 8 2015 |
| 28.06 | 13.72 | 17:19:17 | 13 8 2015 |
| 28.06 | 13.72 | 17:21:19 | 13 8 2015 |
| 28.00 | 13.72 | 17:23:20 | 13 8 2015 |
| 28.06 | 13.72 | 17:25:22 | 13 8 2015 |
| 28.06 | 13.72 | 17:27:24 | 13 8 2015 |
| 28.00 | 13.72 | 17:29:26 | 13 8 2015 |
| 28.00 | 13.72 | 17:31:28 | 13 8 2015 |
| 28.00 | 13.72 | 17:33:30 | 13 8 2015 |
| 28.00 | 12.59 | 17:35:32 | 13 8 2015 |
| 28.00 | 12.35 | 17:37:34 | 13 8 2015 |
| 28.00 | 12.31 | 17:39:36 | 13 8 2015 |
| 28.00 | 11.21 | 17:41:39 | 13 8 2015 |
| 28.00 | 9.99 | 17:43:41 | 13 8 2015 |
| 27.94 | 10.64 | 17:45:43 | 13 8 2015 |
| 27.94 | 10.85 | 17:47:45 | 13 8 2015 |
| 27.94 | 10.38 | 17:49:47 | 13 8 2015 |
| 27.94 | 10.62 | 17:51:50 | 13 8 2015 |
| 27.94 | 10.32 | 17:53:52 | 13 8 2015 |
| 27.94 | 11.66 | 17:55:55 | 13 8 2015 |

These results represent only the first 49 readings taken from the time that the device was in the water. It shows that the water in my pool is incredibly basic, which is not normal, according to an online swimming pool maintenance website. The readings could attest to a faulty sensor, or maybe even an actual problem with my family’s swimming pool.

## Macintosh HD:Users:ArthiIyer:Desktop:imageedit_7_9977790702.jpg4.2 SECOND LAUNCH: AUG 19

The second launch I did used the second prototype. This time, I took a leap of faith and put the device into the Stevens Creek Reservoir to see what data I could collect there. I left it for two days before going back to retrieve it.

# 5.0 PERSONAL CHALLENGES

When I first learned about the Wave Glider in late winter of 2014, I knew almost no programming. The few things I had picked up were because my father is a computer scientist and he had taught me a few things along the way. So, it took me some time to get to the point of feeling confident enough with my abilities to be able to complete this project.

By the time the Wave Glider came into my mind again was the summer of 2015 after I had completed the AP Computer Science course at my high school. Because I waited for my skills to develop like I did before tackling the project, the software part of it was not very difficult from a technical point of view.

However, the hardware was another story. This was my first time doing anything involving physical wires and circuitry. I learned about everything from what a breadboard was to how to solder headers onto an external shield. That was probably my biggest challenge throughout the process. Since I was learning everything on such a blow-by-blow basis, I found myself having to go back and re-learn things a lot.

# 6.0 CONCLUSION

In conclusion, I believe that this project has helped me to achieve my goal in many ways. I taught myself many things about hardware and engineering in general. And on top of that, I achieved something that I like to believe is similar to something already invented and used by the SeaKeepers Society. Both my device and theirs takes temperature readings, but theirs of course is more advanced in the transmission techniques and their velocity readings.

I do plan on doing this kind of work again, as I believe it is greatly useful to the field of science that I am passionate about.

All of the source code for this project can be found on my Github page. Please contact me at [17arthii@students.harker.org](mailto:17arthii@students.harker.org) so that I can give you access to it.

# 7.0 REFERENCES

1. <http://liquidr.com/technology/waveglider/how-it-works.html> [↑](#endnote-ref-1)
2. <https://www.youtube.com/watch?v=xfJq9nQ_m2A> [↑](#endnote-ref-2)
3. <http://oceanexplorer.noaa.gov/technology/tools/drifters/drifters.html> [↑](#endnote-ref-3)
4. <http://seakeepers.org/ProgramsPolicies/DrifterProgram.aspx> [↑](#endnote-ref-4)
5. <https://www.arduino.cc/> [↑](#endnote-ref-5)
6. <https://www.sparkfun.com/products/11050> [↑](#endnote-ref-6)
7. <http://www.vernier.com/products/sensors/ph-sensors/ph-bta/> [↑](#endnote-ref-7)
8. <https://www.sparkfun.com/search/results?term=vernier> [↑](#endnote-ref-8)
9. <https://www.sparkfun.com/products/12761> [↑](#endnote-ref-9)
10. <http://www.vernier.com/engineering/arduino/analog-sensors/linear/> [↑](#endnote-ref-10)
11. <https://github.com/VernierSoftwareTechnology/arduino> [↑](#endnote-ref-11)