Benthic Microbial Fuel Cells Project

Summer, 2020

Author: Emerick Gilliams
Supervisor: Dr. Jordon Beckler

Table of Contents

Overall Project Summary and Introduction:

Hello!

My name is Emerick Gilliams, and I am a rising junior studying chemical engineering at the University of Florida. During the summer of 2020, I was taken on as a renewable energy intern at Harbor Branch Oceanographic Institute where I worked remotely on benthic microbial fuel cells (BMFCs).

BMFCs are an emerging option for renewable energy. BMFCs utilize microbes in the sediment of the benthic layer in a body of water to create an electric current as the microbes oxidize organic and inorganic matter. This creates a redox gradient between the anoxic sediment and oxic water.

The importance of this research and all research with BMFCs strongly relates to oceanographic research, as BMFCs are being investigated as a possible power source for subsea sensors. BMFCs are currently being used as battery substitutes and battery chargers for offshore dataloggers and sensors. Successfully making BMFCs a consistent power source would reduce the costs for device deployment, maintenance, and replacement, which happens to be some of the most expensive portions of marine research.

The overall goal of my project is to explore the capabilities of small-scale BMFCs. During my project, I have created ten indoor BMFCs using sediment and water samples from a nearby lake. I have been testing various sets of three variables: salinity, BMFC design, and addition of carbon-based additives. My project has shown that with even a fuel cell design size of a few inches, harvestable power is accomplishable.

During this research, I have also been working with Arduino circuitry to create a battery-powered device that can monitor the voltage of a fuel cell while recording that data to a micro SD card. This device can monitor up to six BMFCs for about 100 days while connected to a battery. Various code for the device can be found on the projects GitHub: Click Here...

For the conclusion of my project, three BMFCs were constructed in situ in a nearby lake and was monitored by my Arduino circuit in a waterproof enclosure. The data was analyzed after the completion of the program.

In this document, you can find information regarding:

- Materials required to make anything in my project….
- The procedures I created for my project…
- Detailed write-ups and data on each part of my project…
- Recommendations for anyone looking to follow in my footsteps…

Let us begin!

My Indoor Fuel Cells

This part of the document is dedicated to my indoor fuel cells used in my project from the summer of 2020.

You will find the reason why I constructed these, the variables I used, how I collected data, and the results from that data.

For how I constructed these and pictures what they look like, check the procedures section.

Why Create Indoor BMFCs?

There are a couple reasons why you would want to make indoor BMFCs.

The main reason is to control variables you cannot control in the environment. For example, one of the variables I tested was the effects of salinity. I would have never been able to do this if I were setting up my BMFCs in situ. Adding salt (or chemicals if you want to test the effect of that) can and will affect the environment you are testing in. Furthermore, do you know how much salt would be needed to change the salinity of an entire body of water? The only reasonable option for certain variables would be to create an isolated and easily changeable environment; Hence, create yourself a set of indoor BMFCs!

Of course there are other reasons, such as not having access to a good body of water to test in or wanting to test multiple variables at the same time; however, the main benefit is creating an isolated and changeable place to deploy your BMFCs.

My Variables

During my project, I tested three different variables:

1. Salinity
2. Fuel Cell Design
3. Carbon-Based Additives

Why These Variables…

We tested the only two levels of salinity: freshwater from a lake and simulated ocean water. The simulated ocean water was taken from the same sample as the freshwater except for the addition of sea salt in the ratio of salt-to-water that would be found in the ocean. These two levels were chosen since most BMFCs were deployed in the ocean, and the nearest body of water to the location of the project at the time was a lake. An interesting project could be created around testing multiple levels of salinity though!

We also only tested two different fuel cell designs. The designs were the same excluding the fact that one had a chambered anode and one had an unchambered anode. This was done since these are the two classifications of BMFC design. See the

Fuel Cell Related Procedures regarding what these designs looked like.

The final variable was the addition, or non-addition of carbon-based additives to the bottom of the fuel cell container. This was done to see if the additives would encourage bacteria growth at a quicker pace, leading to a stronger voltage, than if no additive were added.

Container Setup…

I set the variables up in the following five combinations with two replicates each:

1. Freshwater/Chambered Design/Additive
2. Ocean Water/Chambered Design/Additive
3. Ocean Water/Chambered Design/No Additive
4. Ocean Water/Unchambered Design/No Additive
5. Ocean Water/Unchambered Design/Additive

You may be asking why I only had one freshwater group. The reason for this is I had eight possible combinations of variables, and I wanted to have two replicates each. One of the downsides of indoor BMFCs is you must have the space to place them. I only had the space for ten fuel cells, so three possible variable combinations had to be discarded.

After review, it was obvious that salt water would be a better environment for BMFCs due to the increased ionization in the water around the BMFCs. Due to this project being remote and the desire to strong results, Dr. Beckler and I decided to do mostly saltwater fuel cells.

Data Gathering and Analysis

Daily Voltage Readings…

The main data that I gathered during my project was the daily voltage readings using a multimeter with leads connected to the anode and cathode. This was done twice a day at 10 AM and 4 PM. I find that this was unnecessary and that once a day is sufficient for future projects. These numbers gave an idea on how developed the redox gradient was at the time of measurement.

You can expect anywhere between 400 mV and 600 mv for a strong BMFC. In even the most ideal settings, the strongest redox potential you will see is 1.2V, so do not be disappointed by low voltages! Some of my fuel cells began with a negative voltage and slowly moved to positive over time. I began testing my cells the day after they were constructed, and I was already getting some interesting results.

Other Data Collection…

There are other tests you can run to gather data from as well. In the results section, there are some graphs regarding what I called “discharging” and “charging” tests. In the discharging test, the cathode and anode were set in series with a resistor. The voltage was monitored over a period and then plotted to observe the behavior of the fuel cells when discharging. This process is actually very important for constructing a polarization curve, something I never had the opportunity to do. See the Recommendations for more information on polarization curves.

The charging test was the exact opposite of this. The fuel cell would be discharged completely by touching the cathode and anode together, then the fuel cell would be monitored while the voltage increased back to its current high point, which is the same as that days daily measurement voltage values.

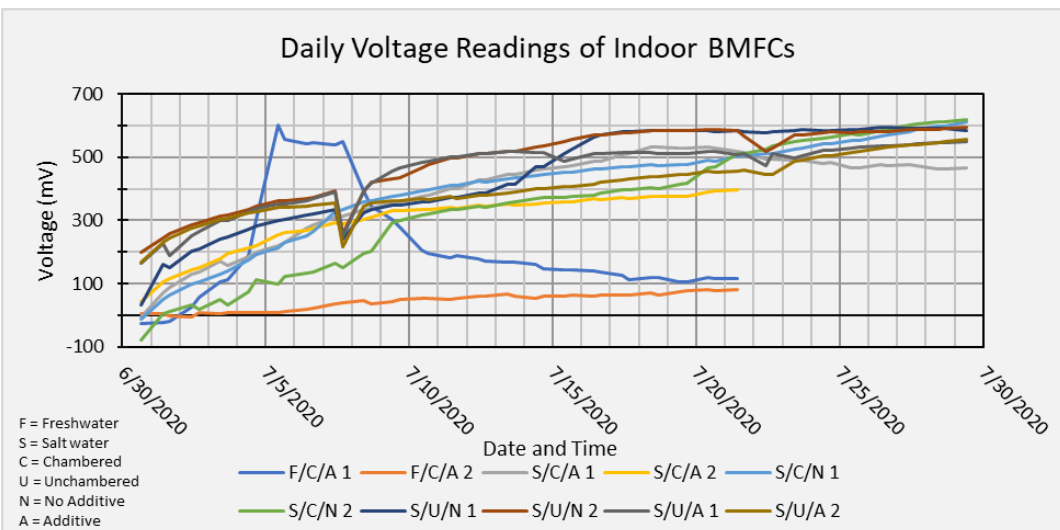
Both these tests can be done with the custom-made Arduino circuit and the *"Voltmeter_SD_RTC2_Sleep_Mini_Double_Charging_Test"* code. This code will monitor the fuel cell and record a measurement every 10 seconds. Make sure to calibrate and test on batteries to make sure it is accurate!

One test I never conducted but is very interesting to conduct would be an oscillating circuit. See the Recommendations.

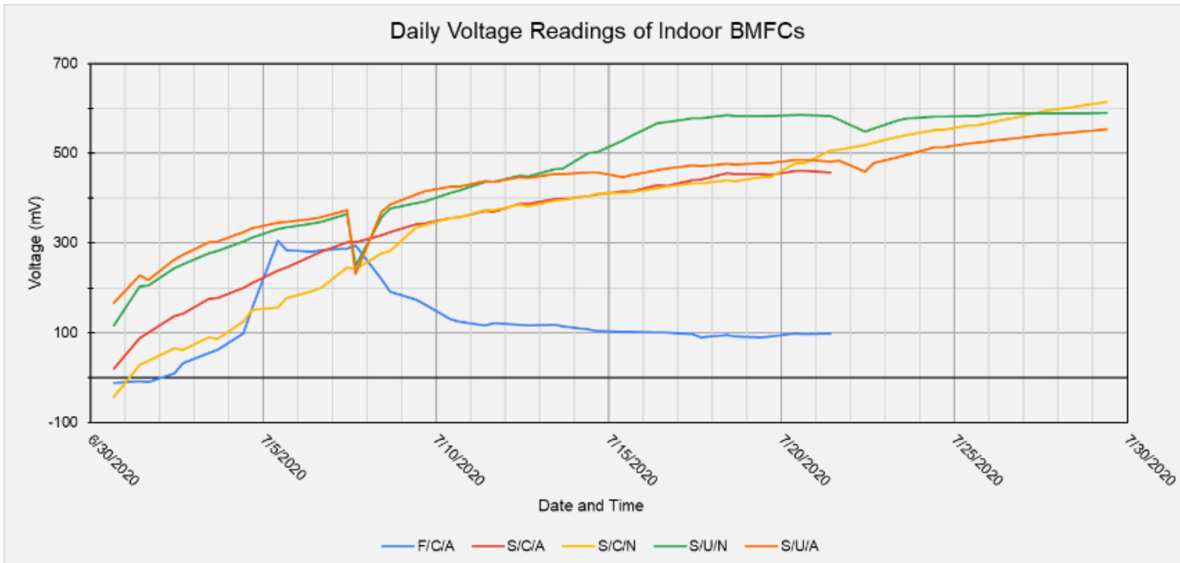
Results

Daily Voltage Measurements…

The following graph shows the daily measurements taken twice a day. Although some fuel cells started in the negative, a large majority of the fuel cells showed strong steady growth over the testing period. The highest points are in the low 600 mV range. The downward spike of the highest performing fuel cells near 7/7/20 was due to testing done on the cells that lowered the voltage temporarily. The three worst performing fuel cells were salvaged at a certain point as the materials were needed to construct the three in situ fuel cells used later in the project. This is seen in both graphs.



The following graph shows the daily measurements, but the replicates are grouped and averaged into their variable sets. From this you can see that the freshwater cells are the least consistent, while the saltwater cells are all consistently strong and steadily growing. This is most likely due to the increased ionization in the fuel cells with saltwater. The simple conclusion is saltwater fuel cells are stronger and a better option than freshwater.



You can also see that unchambered vs chambered are generally the same, but with the slight advantage to the unchambered designs. This leads to the conclusion that the real variable here is ease of deployment. The easy the design is to deploy, the better chance the electrodes will be setup properly. The unchambered design is extremely easy to deploy in an indoor fuel cell, which many have given it the slight advantage it is shown to have. The chambered design would be easier to deploy in situ since the sediment in situ is much more compact, making it more difficult for an unchambered design to pierce the sediment.

The final variable, the carbon-based additive, seems to have little to no effect on the overall fuel cell voltage over longer periods of time; However, if you look at the first two weeks of deployment, you can see slight advantages to the fuel cells with additive. Coincidentally, the additive used was a 14-day slow-release vacation fish food. There may or may not be a connection. More testing needs to be done. There is no negative to adding an additive, only a possible benefit in having higher voltages earlier on. I would recommend the use of an additive if possible.

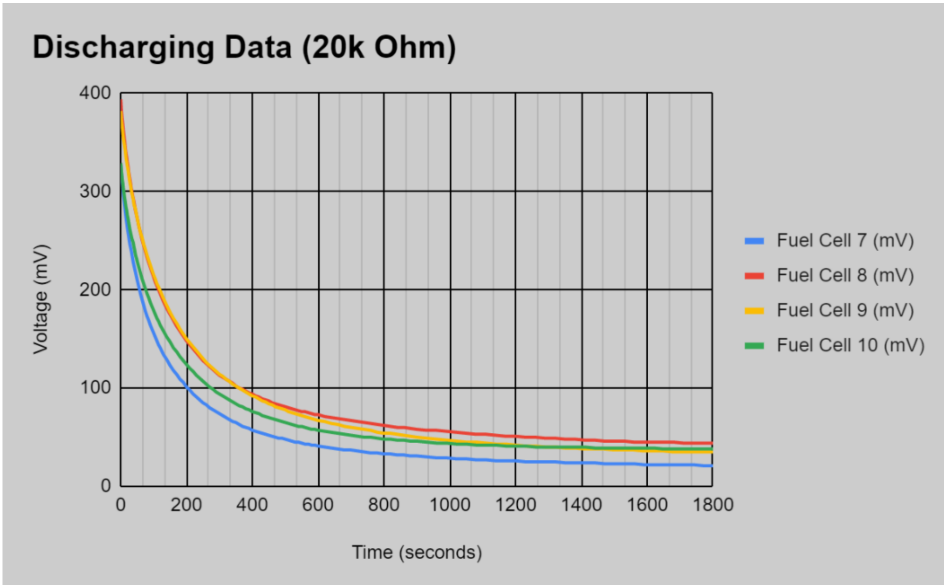
Discharging Tests…

This data was taken by placing a 20k Ohm resistor in series with the cathode and anode. The voltage was then measured across the resistor for 30 minutes and plotted. The four strongest fuel cells were fuel cells 7-10, which is why they were chosen for this test.

As you can see, the fuel cells will rapidly lose their charge and then stabilize at a lower voltage after a certain period. That stabilization voltage is the value needed to construct a polarization curve, which was not known to me at the time of the test. See the Recommendations section for more information.

The starting voltage is purely dependent on how developed the fuel cell is. The starting voltages coincide with the daily voltage reading of the day of the test.

This test was taken by recording a multimeter, but if you would like to do a test like this, use the code that is described in the "Other Data Collection…" section.

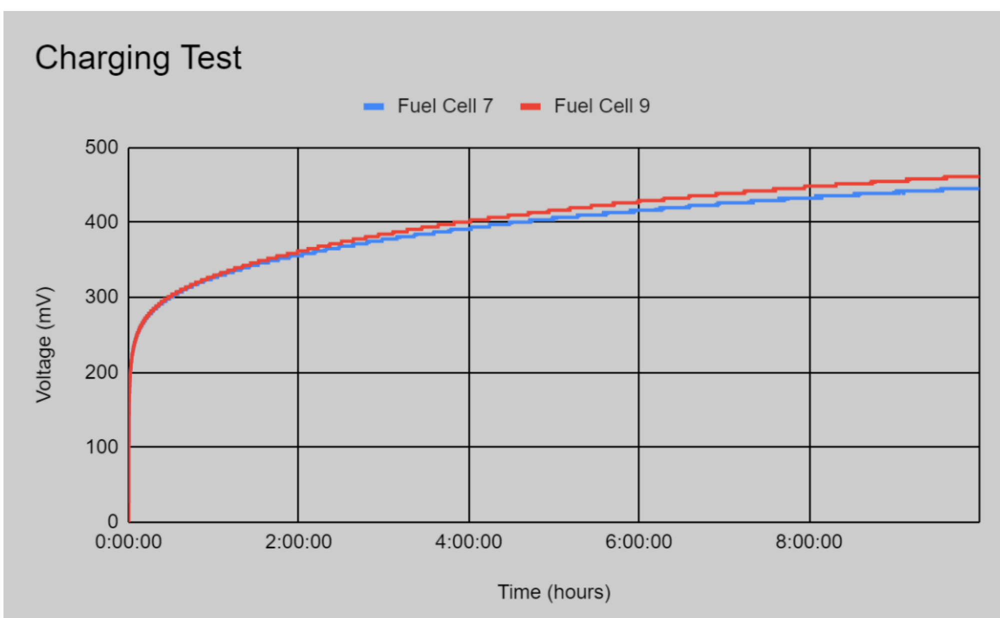


Charging Tests…

This data was taken by the code described in the "Other Data Collection…" section. The two of the strongest fuel cells at the time of testing was Fuel Cell 7 and 9. They were also chosen in a pair due to their proximity on the physical workstation.

As you can see, the charging behavior is very similar to the discharging behavior, as there is rapid change in the beginning and slow change as time goes on.

The test was stopped after 10 hours but could have continue to reach a better stabilizing voltage.



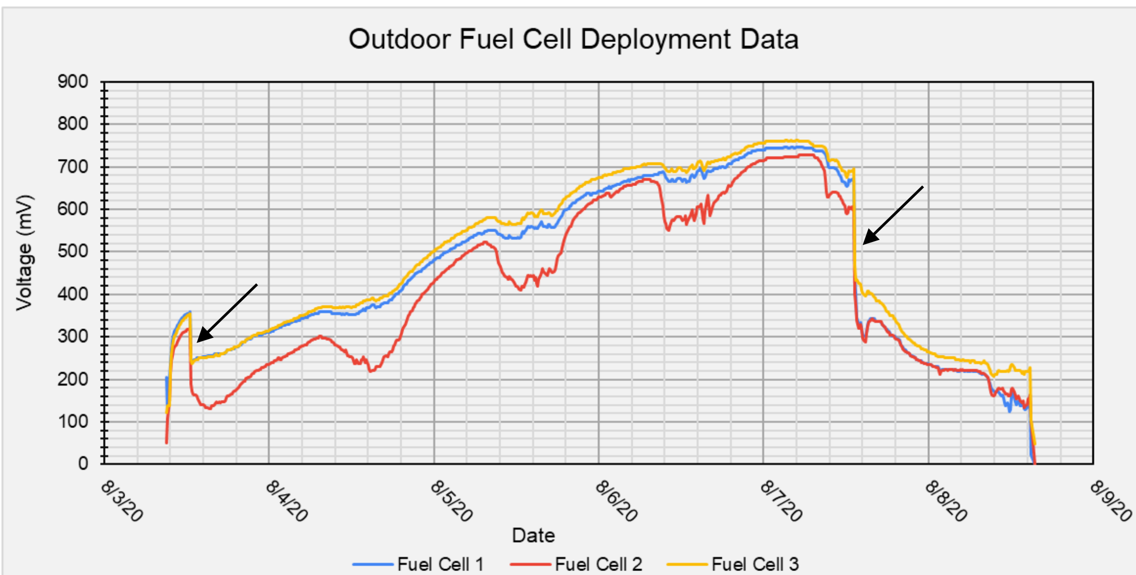
My Outdoor Fuel Cells

Data Collection:

Three fuel cells were deployed in a lake near me at the time. These fuel cells were being monitored by the custom-made device that is described further in the document. The fuel cells were deployed for slightly under a week before they were retrieved.

The device measured the voltage in mV every 10 minutes for the entire time. This data was transcribed onto a micro SD card for later analysis.

Results:



The graph above is the data for the outdoor deployment.

The first thing to discuss are the issues with deployment and how it affected the data.

I checked the state of the fuel cells every 1-2 days. When I went to check a day after deployment, Fuel Cell 2 was floating. I decided against redeploying the cell for these reasons.

1. I was uncertain if the data collector was fully operating due to a previous deployment issue.
2. if the measuring device was operating correctly, I did not want to disturb the sediment and affect the data collection.
3. I had two more, so I considered losing one was not an issue.

A few days after Fuel Cell 2 became dislodged, Fuel Cell 1 also began to float. You can see the effect of these changes on the voltages when looking at the two black arrows on the plot. The left most arrow is showing when Fuel Cell 2 dislodged, and the right arrow is Fuel Cell 1.

Now, why are the voltages all about the same and follow the same patterns even though fuel cells were dislodging?

I believe this is due to the concept that when multiple BMFCs are in the same body of water, they act as one BMFC.

As you can see, on the day Fuel Cell 2 dislodged, the red line dipped the strongest and was always under the other two lines. It also affected the overall voltage growth as you can see all the lines were increasing at a high rate in the beginning, then that rate decreased after the first dislodge. As time went on the yellow and blue lines both stayed about the same, with the red line being consistently worse.

It appears the lines were going to plateau at around 800 mV when the other fuel cell dislodged causing the voltage to drop significantly again. The overall plateau also decreased to around 200 mV. Furthermore, the fuel cell that dislodged next (Fuel Cell 1) then joined the red line and was the same until they were pulled from the sediment.

Another interesting thing that happens is **every day at the same time** there are periods of instability in the voltage. Furthermore, both fuel cells dislodged at a time during this instability. My guess is it has to do with the movement patterns of life in the lake. It makes sense that an increased amount of movement around the BMFCs would affect the voltage. It also makes sense that increased movement around the BMFCs could have jostled them enough to dislodge them from the sediment.

Interesting conclusions:

1. Supporting evidence that all BMFCs in a single body of water act as one BMFC.
2. If a BMFC dislodges, it will still hold a voltage, but one slightly lower than the ones still in the sediment
3. The amount of BMFCs in the sediment determines the overall maximum voltage.
4. If all fuel cells stayed in place, we could have gotten a value well over a volt in all cells, which is very interesting since the highest we saw in a month of indoor testing in environments with salt and additives is only around 625 mV.
5. Movement of life around the fuel cells could have cause periods of instability.
6. **Make sure future BMFC designs are not buoyant to avoid u*nintentional* dislodges.**

Materials:

All materials that are listed can be found in the Supply List.

This section is divided into categories for each part of the project, so some materials are listed multiple times if needed for different sections. Make sure you double check to not order things multiple times!

That supply list will give you an estimated cost for each item suggested, and a link or description of where you can find one!

For the sake of simplicity, we are going to use the symbol X to represent the number of BMFCs you would like to create.

For example, in my project X = 10.

If you see an *[X * #]* that is the quantity needed. No brackets imply 1.

Materials for a Deployable Fuel Cell:

The Electrode System…

We are first going to list the materials for what I call the "electrode system" (For me, X = 10):

1. [X * 2] Electrodes
    a. I used Carbon Fiber Brushes from Mill-Rose. I highly recommend these electrodes if your budget allows it. I used 2-inch brushes with 1-inch bristle length and 1-inch bristle radius
2. Electrical Wire
    a. If you have the chance, get two colors! Black and red works perfectly!
3. Soldering Kit
    a. The one I linked came with wire strippers and tweezers; **if you decide to buy a different one, purchase these as well.**
4. Wire Quick Connect Kit
5. Epoxy
6. Heat shrink tubing
7. Heat Gun
8. Multimeter
    a. Test your connections every time you solder, epoxy, or connect wires!!!!
9. Pack of AA batteries
    a. Use these with your multimeter or Arduino to test your connections or code/circuit.
10. Rosin Core Solder
    a. In the Arduino Purchase section…
11. Painters Tape
12. Cling Wrap
13. Small zip ties
14. Pliers.

15. Some type of PVC to hold your electrodes (I used the ¾ inch PVC)
    a. Mine is listed in the Fuel Cell PVC Design Section
16. Cardboard from boxes items came in

These materials are all highly recommended and should be purchased if you want to recreate the project.

Fuel Cell PVC Design…

These materials are for the PVC housing you that hold your electrodes system.

Depending on the design you chose, this will be different. I listed my materials if you wish to replicate my design exactly. See the Recommendations ***section…***

For my 10 fuel cells, I had 6 chambered designs and 4 unchambered designs.

Here are the materials I used for my unchambered design (X = 4):

1. 3/4-inch diameter PVC pipe segment
2. [X] 3/4 elbow PVC pipe
    a. I recommend using a vertical coupling if you have enough space in the enclosure. I used the elbows, so they are what are listed.
3. PVC pipe cutter

Here are the materials I used for the chambered design (X = 6):

1. 3/4-inch diameter PVC pipe segment
2. [X] 3/4 elbow PVC pipe
3. PVC pipe cutter
4. 3-inch diameter PVC pipe segment
5. [X] 3-inch x 1.5-inch PVC Coupler
6. [X] 1.5-inch x 3/4-inch PVC Bushing
7. Hacksaw
8. Sandpaper (The coarser the paper, the better.)

Once again, all these materials are subject to change if you use a different design.

Materials for Arduino Circuitry and Enclosure

Arduino Circuitry…

1. Micro SD Card reader
    a. Needed for the circuit to save the data it is monitoring
2. Micro SD Card
3. M-F, F-F, M-M Wires Kit
    a. Needed for certain connections. Does not hurt to have extra wires either.
4. Pre-Formed Jumper Wire Kit
    a. Needed for most connections. Keeps breadboard clean and easy to organize.
5. DS3231 Real Time Clock

    a. Needed to allow Arduino to log the time measurements were taken
6. Battery for DS3231
7. Arduino Pro Mini 3.3V 8Mhz
    a. Arduino of choice for the circuit.
8. Rosin Core Solder
    a. In Electrode Materials
9. Breadboards
10. Lithium Polymer Battery Connector
    a. This makes plugs into the charger and allows you to connect it to breadboard
11. Lithium Polymer Charger Circuit
    a. This makes the battery connection safer and allows charging if needed
12. Lithium Polymer Battery
13. FDTI Breakout
    a. This allows the communication between the mini and the cable. Needed for programming.
    b. Get the circuit that matches the voltage of the Arduino Pro Mini you have. Mine were both 3.3v, and that is what I recommend.
14. FDTI Cable
    a. This allows the communication between the breakout and a PC or laptop. Needed for Programming.
15. Header pins

Arduino Enclosure…

This is the materials you need to construct the enclosure for the Arduino to go underwater. Please pay attention to the actual design of the enclosure. The number of penetrators, thruster cable lengths, and blanks depend on how many cells you are testing.

1. Potting Kit
    a. Needed to apply the epoxy to the penetrators.
2. Marine Epoxy
    a. Needed to seal the penetrators
3. M10 Cable Penetrator for 6mm Cable
    a. Needed to pass the cables through the enclosure cap.
    b. For 1 fuel cell to be monitored: 1 Penetrator needed
    c. For 2-3 cells: 2 penetrators needed
    d. For 4 cells: 3 penetrators needed
    e. For 5-6 cells: 4 penetrators needed
4. Penetrator Wrench
    a. Used to tighten the penetrators
5. M10 Cable Penetrator Blank (No Hole)
    a. Needed to fill the empty holes in end cap
    b. For 1 fuel cell to be monitored: 3 blanks needed
    c. For 2-3 cells: 2 blanks needed
    d. For 4 cells: 1 blank needed
    e. For 5-6 cells: 0 blanks needed
6. 1 Meter of Thruster Cable (3 conductors, 18 AWG)
    a. Same amount of lengths needed as penetrators.
    b. Adjust length if you want more space between cells.
7. 200 mm of Customizable Cast Acrylic Tube (4″ Series)
8. Aluminum End Cap with 5 Holes (4″ Series)
9. 2 O-Ring Flanges (4″ Series)
    a. Needed for both sides
10. Aluminum End Cap (4″ Series)

11. Enclosure Vent and Plug
    a. Needed for all enclosures. Cannot skip.
12. Silicon Grease
13. Hex Driver
14. Exacto Knife
    a. Needed to strip thruster cable.

Materials for Fuel Cell Deployment

Variables…

These final materials are based on what variables you test. I tested Salinity, Chamber design, and the addition of a Carbon based additive.

For salinity:

1. Salt

For Chamber design:
Look at previous section

For Carbon-Based Additive:

1. Vacation slow release fish food

Indoor Deployment…

These are the materials needed to deploy your fuel cells into indoor containers. For me, X = 10. If you are deploying your fuel cells outdoors or in situ, you may not need these materials.

1. [X] Deployable Fuel Cell
    a. You need something to deploy obviously. Consult the electrode system materials/PVC housing materials to find out what you need.
2. [X] Plastic Containers for the fuel cell
    a. You can find cheaper ones not on amazon
3. [X * 0.5] Air Pumps
    a. I only need 5 since the pumps I got were double tubed. HIGHLY RECOMMEND DOUBLE TUBED.
4. Power strip
    a. To plug the Air pumps into. If you need more than 6 air pumps, pick a strip that fits your needs.
5. Painters Tape
    a. In Electrode System materials…
6. Shovel
7. 2 Large 17-gallon bins
8. Cardboard from boxes items came in
    a. Same as from Materials for Electrode System. Recommended if you do not have workstation.

If you are deploying indoors, you will want to actively test and collect data. This is what you will want.

1. Multimeter
    a. Same as Electrode System section
2. Alligator Clip kit

3. Variable Resistors/resistor kit
    a. You will need these to conduct discharge tests or polarization curves. See Polarization Curves… and Data Gathering and Analysis.
4. Completed Arduino Circuit
    a. Helpful to have for discharging and charging tests. Not needed though. See Arduino materials for what you need for that.

Outdoor Deployment…

These are the materials for outdoor deployment.

1. [X] Deployable Fuel Cell
    a. You need something to deploy obviously. Consult the electrode system materials/PVC housing materials to find out what you need.
2. Completed Arduino Circuitry
    a. Needed to monitor your in-situ cells. Consult the Arduino Circuitry Materials
3. Completed Arduino Enclosure
    a. Needed to keep your circuit waterproof. Consult the Arduino Enclosure Materials
4. Cinderblock or weight of some kind
    a. Needed to weigh down the completed enclosure.
5. Something to loosen the sediment
    a. Optional, but very helpful. I used an extra PVC pipe sediment to stab at the sediment to loosen it.

Procedures

Arduino Related Procedures

Arduino Material Modifications…

The first step before modifying the modules is to understand why.

These modifications were made to lower the current consumption to extend the battery life.

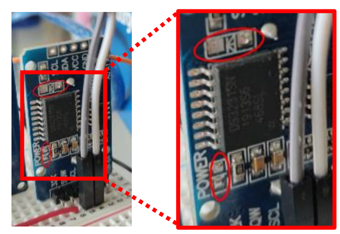
After the modifications, the current during sleep mode of both the triple input, double input and single input circuit is 0.8mA.

After the modifications, the current during active measurement is 4.8mA for a single input, 5.0mA for a double input, and 5.2mA for a triple input.

Why the following modifications were made:

DS3231 Real Time Clock Module:

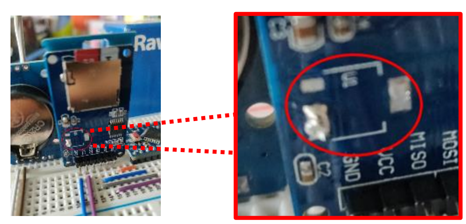
- Removal of the diode in the top left of the module...
    - This was removed to disconnect the trickle charge current that goes into the battery socket which attempts to charge the battery.
    - I recommend the use of a CR2032 3V Lithium Coin Battery. If you use this battery, the trickle charges the module is attempting to provide can explode the battery.
    - If you decide to use a rechargeable battery, you can keep this. I suggest against using the rechargeable battery over the CR2032 3V Lithium Coin Battery.



- Removal of the Power LED in the bottom left of the module...
    - This was removed to lower the power consumption of the module in sleep and active mode, as the power LED is always on.
    - The removal of this saved about 2 mA in sleep and active mode. This is a must remove to maximize power consumption.

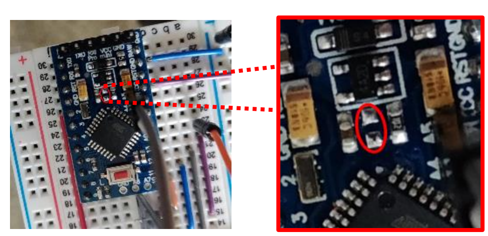
Micro SD Card Module:

- Removal of the AMS1117 Voltage Regulator...
    - This is a must remove IF you are using a 3.3V, 8MHz, Pro Mini Module (Or any Arduino/Microcomputer setup operating at 3.3V).
    - Micro SD cards operate at 3.3V originally. The Micro SD module used in the project has a voltage regulator that takes a 5V input from an Arduino/breadboard/source and regulates it to a 3.3V for the SD card.
    - Since we are operating at 3.3V, this regulator will not work correctly. This is a MUST remove if operating under these conditions.



Arduino Pro Mini

- Removal of the Power LED...
    - This was removed to lower the power consumption of the module in sleep and active mode, as the power LED is always on.
    - The removal of this saved about 2 mA in sleep and active mode. This is a must remove to maximize power consumption.



Supplies to make the Modifications:

Supplies Needed:

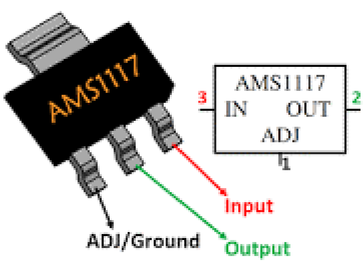
1. Heat Gun
2. Insulated Tweezers (Should have come in your soldering kit)
3. Soldering Iron
4. Rosin Core Solder

How to make the Modifications:

The removal of the components is relatively easy! All components follow the same steps to be removed from the PCBs (Printed Circuit Board).

1. Plug in Heat Gun
2. Turn on Heat Gun
3. Use Heat Gun to warm the ENTIRE board.
    a. The board does not need to be as hot as the component you are heating, but it is helpful to heat the whole board slightly.
4. Focus the Heat Gun on the component you are removing with one hand and pick up the tweezers with the other hand.
5. Either pick up the component with the tweezers once the solder is warm enough, or you can just knock the component off its pads by tapping it if there is enough room.
6. Turn off Heat Gun

This is all you need to do for all the modifications **excluding the Micro SD Module**.



For the Micro SD Module, after removing the regulator, the voltage input and output pins need to be connected.

For the AMS1117 module on my Micro SD Module, here is the relevant pinout. **Your Micro SD Module may be different so double check!**

Since the relevant pads for the AMS1117 module were next to each other, I connected the pads with a solder blob. It is ugly, but it works.

Make sure you double check the model of regulator you have before doing this to make sure you know which pads are which!

Arduino Circuitry Construction…

This is much easier than it sounds if you have never done this before.

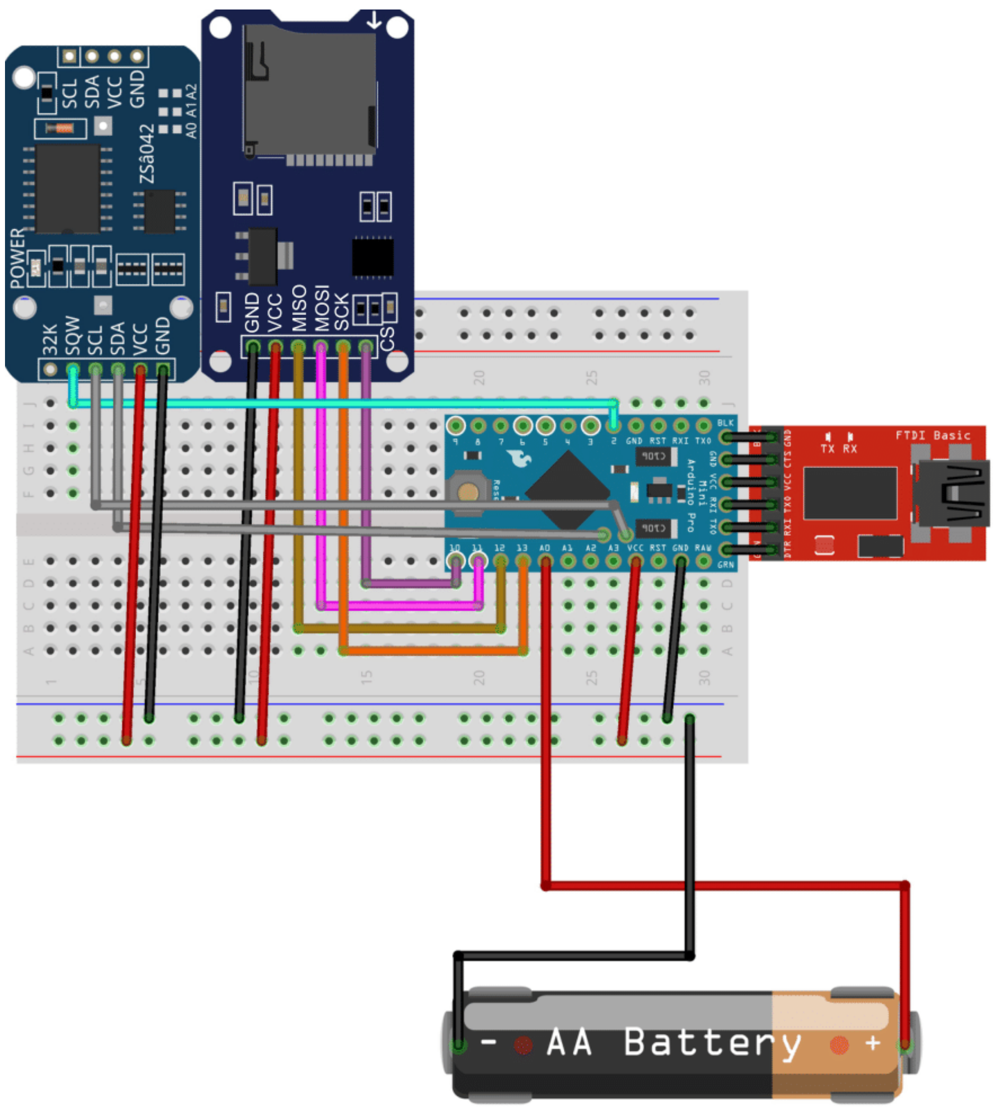
Here is my favorite Pro Mini Pinout. Very Useful!!! Pro Mini Pinout…

The hardest part is soldering together the Pro Mini if it is unsoldered. Here is a tutorial on how you can do that: https://www.youtube.com/watch?v=37mW1i_oEpA. I suggest using some vertical header pins instead of the right-angle ones you were probably given with your Pro Mini. This will make it easier to program the Arduino. **You will also need to solder the A4 and A5 pins to use the Real Time Clock**, and the A6 and A7 pins if you want to monitor 5 or 6 fuel cells (Not all Pro Minis have these pins, but mine did).

The picture below is the circuit diagram of the voltmeter that measures a **SINGLE** fuel cell. This is what you should build first **after the modifications to your PCBs and you have soldered the Pro Mini**. You can ignore the Red FDTI plug on the left connected to the Pro Mini. This will be addressed in the Programing the Arduino… section.

What if you want more than one fuel cell? Construct this first then you can add the additional wires at a later point after you verify that your circuit is working on a single fuel cell. The information for how to do this is in the Monitoring Multiple Fuel Cells… section.

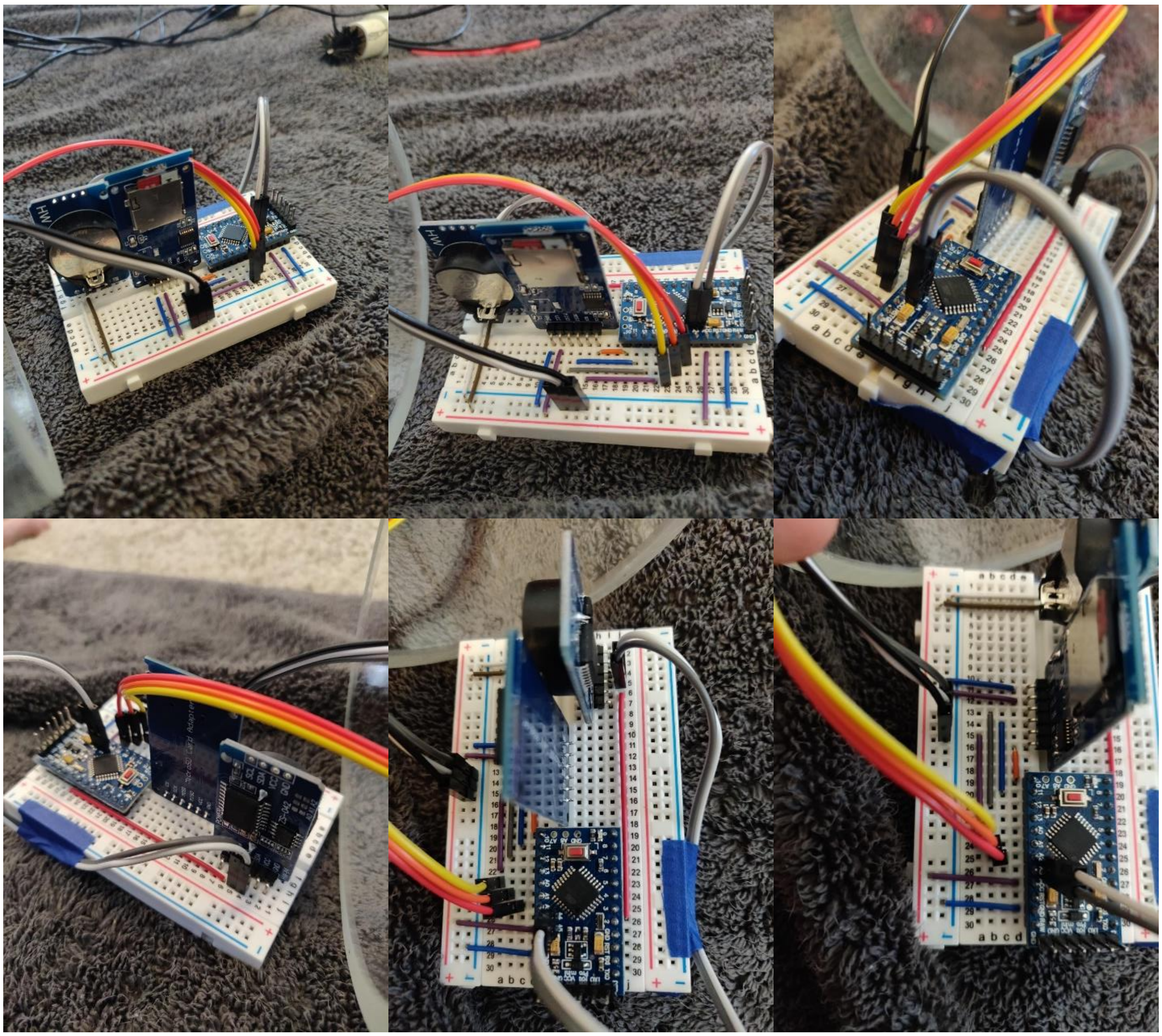
The AA battery is only a placeholder and an example on how you would connect something you want to see the voltage of to the voltmeter. You can read more about this in Testing and Measuring with the Arduino Voltmeter….



I am not going to give a detailed procedure on how to put this together. You literally just plug the wires into the right slots, and you are good to go. I have provided a couple pictures of my organized breadboard from different angles. This will give you an idea on how to organize yours. Feel free to copy mine. Keep in mind that the pictures being provided are for the **Triple Fuel Cell Circuit.** This means that **you do not need as many wires coming from the ground or A0-A2**. When looking at the pictures, you should **imagine that the gray + black wires and the red + orange wires from the two wire triplets do not exist!**

Expand the pictures for a closer look at row numbers etc.

The white and gray wire doublet is taped just to save space. These pictures are from right before inserting the circuit into its enclosure. Don't worry, we'll get there soon.



Programing the Arduino…

<u>Before this section, I am going to preface that if you are having any issues with the beginning portion of this section, look up some online tutorials. They will explain things better than I can.</u>

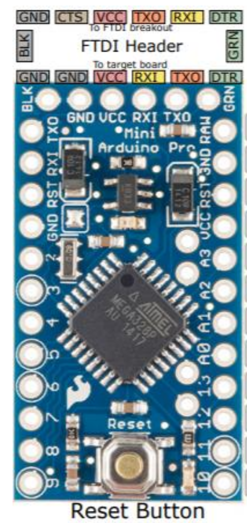
You will need to have a fully constructed circuit before attempting to program it. If you have not already done so download the Arduino IDE. I recommend the desktop offline version. [Click Here…]()

Scroll down and select what operating system you are using.

After this is completed, **DOWNLOAD THE REQUIRED LIBRARIES FROM THE GITHUB:** [Click Here] for the files and [click here] for a tutorial on how to actually put those libraries into the Arduino IDE.

To give you a general idea of what to do…

1. Download the Zip from the GitHub.
2. Extract the Zip
3. Place the entire folder in the "Arduino" folder.
4. Drag the three libraries into the "libraries" folder in the "Arduino" folder.
5. Now all your libraries are in the right place, and you have all the codes I created in your Arduino folder as well.



After that, you are going to want to plug your FDTI cable into your FDTI breakout circuit. The breakout will attach to your 6 Arduino Pro Mini FDTI pins. Use the image attached to the side to double check what pins go into what slots. My breakout matched my pins in reverse, so I had to attach it upside down. Make sure to double check!

After you do that, your Arduino Pro Mini should receive power. Make sure you are **using the right port, processor, and board type** in the IDE. This will cause issues, and I have made plenty of those mistakes! You should open the "Blink" example and play around with the timing of the built-in LED to make sure the programmer works.

If this is your first time messing with Arduino, just familiarize yourself with the IDE and understand the basics. Watch some tutorials on the basics like uploading and compiling.

I will say it again, if you are having any issues with anything up to here, I may have missed something. I am not an Arduino expert; I am a chemical engineering student. Look up some tutorials and I am certain it will fill in the blanks.

After all these steps, you just need to open my Arduino code called "Voltmeter_SD_RTC2_Sleep_Mini." This will be the code that allows you to measure a single item, whether it is a battery or BMFC.

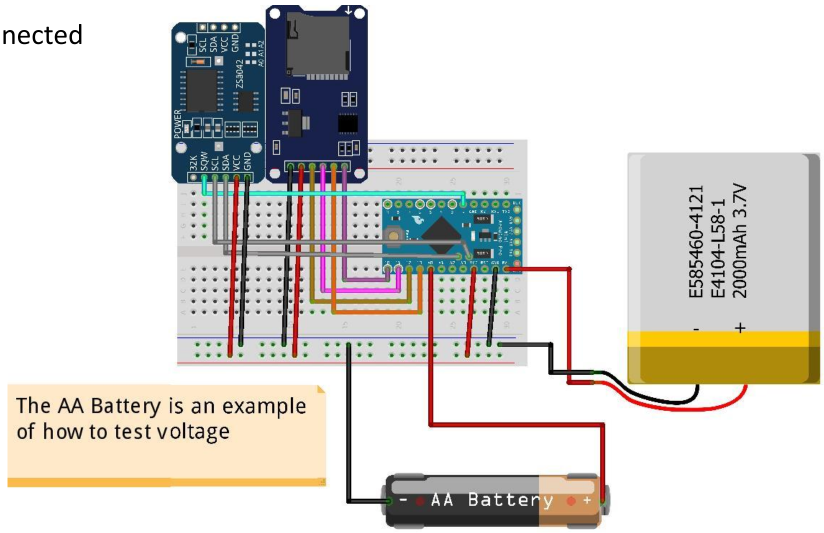
If you are curious with the naming of the code, this is the breakdown: The circuit is a **Voltmeter** that uses an **SD** module and the **2**nd **R**eal **T**ime **C**lock library I tried to use. It saves power using **Sleep** mode and it is compatible with the Pro **Mini**. Nothing too crazy there. You can probably guess what order I added things…

Here are the other circuit diagrams as well. The first diagram shows how you connect the LiPo to the circuit of the single input. All other diagrams have the LiPo connected as well

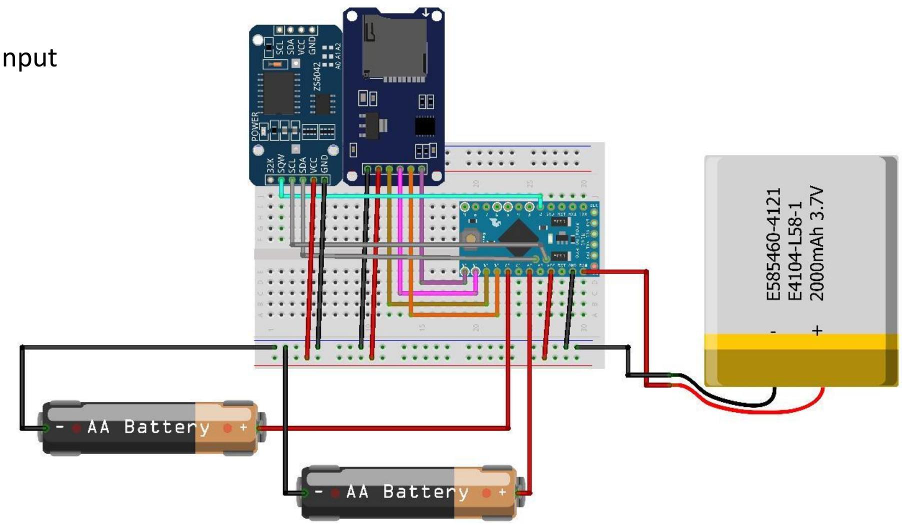
The next diagrams are for a double input, and the final diagram is the triple input.

The diagram files are all on the GitHub as well.

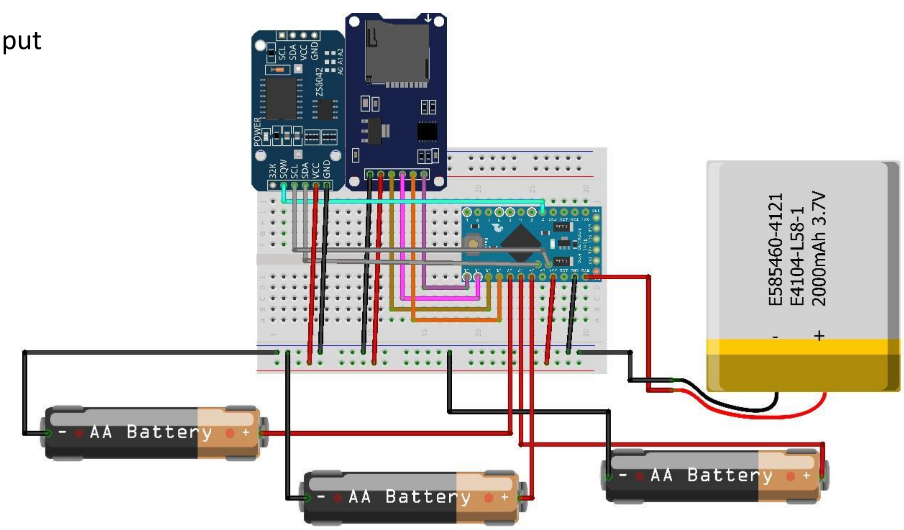
1. LiPo Connected



2. Double Input



3. Triple Input



Calibrating the Arduino Voltmeter…

Now that your code is uploaded, you need to calibrate your voltmeter.

Calibrating your Voltmeter code to get accurate voltmeter readings is very easy!

All you must do is take the positive lead of a multimeter and connect it to the VCC pin either through a breadboard or directly. (I prefer using a wire connected to the same row as the VCC to connect my positive lead too.)

You then connect the negative lead to the ground of a breadboard (if connected to the Arduino ground) or to the ground on the Arduino directly. (I prefer a wire connected to the breadboard to connect my negative lead to.)

Read your Multimeter's voltage reading. It should be near 3.3V if you have the 3.3V Pro Mini.

The voltage your multimeter is reading is what should be entered into the "Cali" variable in any of the codes I wrote.

Make sure to recalibrate when switching between like LiPos, PC USB slots, and other power sources.

Testing and Measuring with the Arduino Voltmeter…

After the construction and calibration of the voltmeter, you should test the voltmeter is getting the correct voltage readings. To do this, I would use the multimeter to check the voltage of an AA battery. If the battery is new, the voltage should be around 1.5-1.6V. You need a voltage source less than 3V since that is the maximum reading for the voltmeter you created.

Once you have a reliable reading from your AA battery, then take the positive lead (The analog pin wire) and the negative lead (The ground wire) and attach them to the positive and negative ends of the AA battery, respectively. To see your voltage reading, you must have the device being powered by your PC or laptop. The voltage reading will be shown in the serial window. If you are not getting a similar voltage reading, something is wrong with your design. Make sure your connections between the Arduino voltmeter and the AA battery/whatever you are testing are very secure. Do not be afraid to touch wire ends with your fingers.

Measuring your fuel cells is the same process. You connect the positive wire of the Arduino to the cathode of the fuel cell. You connect the ground wire to the anode wire of the fuel cell.

Monitoring Multiple Fuel Cells…

To monitor 2 or 3 fuel cells, you just need to upload the version of the code for either the double or triple input.

For the double input, use A0 and A2 as your analog pins. All you need to do change the single input hardware to the double input is add a new positive wire to the same row as A2 and an additional ground wire to the ground strip on the breadboard.

For the triple input, use A0, A1, and A2. To change from double to triple input, add the additional positive wire to A1 and an additional wire to the ground strip.

For monitoring more than three fuel cells, you must adjust the code. I will give you some of the things that need to be adjusted to change the triple input to a quad input. Changing from a quad to quintuple or sextuple input is the same changes. **I will be referencing lines in the directions below, use the pictures provided, not your own code lines. Your lines will be changing as you add lines to your edited code.**

- Add an additional float variable for input_voltage4 under "Voltmeter variables" around line 35.
- Add an additional line at the end of the following block before the close command. Copy and

```
106   // If the file opened, write to it:
107   if (voltData) {
108     // Write to file
109     voltData.print("Date");
110     voltData.print(",    ");
111     voltData.print("Time");
112     voltData.print(",     ");
113     voltData.print("Voltage (A0)");
114     voltData.print(", ");
115     voltData.print("Voltage (A1)");
116     voltData.print(", ");
117     voltData.println("Voltage (A2)");
118     voltData.close();}
```

paste 116 and 117 below it so there are now two duplicates of the comma and voltage (A2) lines right after each other. Change the top duplicate from println to print and change the second (A2) to the analog you are using for the next input (**USE A3 OR CHANGE IN FOLLOWING DIRECTIONS**).

- In the following lines, copy this under to line 149:
  int analog_value4 = analogRead(A3);
  *input_voltage4 = (analog_value4 * (Cali)*1000) / 1024.0;*

```
140 //Conversion formula for voltage
141 //Change the 3.35 to whatever the voltage is coming from the VCC pin on
142
143     int analog_value = analogRead(A0);
144     input_voltage = (analog_value * (Cali)*1000) / 1024.0;
145     int analog_value2 = analogRead(A1);
146     input_voltage2 = (analog_value2 * (Cali)*1000) / 1024.0;
147     int analog_value3 = analogRead(A2);
148     input_voltage3 = (analog_value3 * (Cali)*1000) / 1024.0;
149
150     if (input_voltage < 0.01)
151     {
152       input_voltage=0.0;
```

- Copy this to line 162:
  if (input_voltage4 < 0.01)
  {
  input_voltage4=0.0;
  }

```
149
150     if (input_voltage < 0.01)
151     {
152       input_voltage=0.0;
153     }
154     if (input_voltage2 < 0.01)
155     {
156       input_voltage2=0.0;
157     }
158     if (input_voltage3 < 0.01)
159     {
160       input_voltage3=0.0;
161     }
162
```

- Change the *println* to *print* on line 168 and copy and paste the following to 169:
  Serial.print(", ");
  Serial.println(input_voltage4);

```
162
163    Serial.print("mV = ");
164    Serial.print(input_voltage);
165    Serial.print(", ");
166    Serial.print(input_voltage2);
167    Serial.print(", ");
168    Serial.println(input_voltage3);
169
```

- Change the *println* to *print* on 224. Copy and paste the following so it is between 224 and 225.
  voltData.print(", ");
  voltData.println(input_voltage4);

```
219
220    voltData.print(input_voltage);
221    voltData.print(", ");
222    voltData.print(input_voltage2);
223    voltData.print(", ");
224    voltData.println(input_voltage3);
225    voltData.close();
226
227    Serial.print("Written to file: ");
228
```

- Change the *println* to *print* on 249. Copy and paste the following so it is after 249.
  Serial.print(", ");
  Serial.println(input_voltage4);

```
244
245      Serial.print(input_voltage);
246      Serial.print(", ");
247      Serial.print(input_voltage2);
248      Serial.print(", ");
249      Serial.println(input_voltage3);
250    }
251
252    // If the file didn't open, print an error:
253    else {
254      Serial.println("error opening file");
255    }
```

That is all the changes you should have to make. I may have missed something though since I did not have an actual chance to test these changes. Adding more inputs is not difficult and mostly copy and pasting what commands are needed to be followed for the previous analog pins 0-2. If you understand all the code, you should not even need these rough instructions.

For the hardware, you just need to add another wire to the additional analog pin being used as another input (For the recent instructions, it was A3) as well as another wire to the ground strip.

I think you can guess how to modify the circuit and code to add a 5^{th} and 6^{th} input.

Conducting Discharging and Charging Tests…

To conduct the discharge test, first connect the Arduino voltmeter leads to both sides of a resistor and turn on the device. Once the device is actively measuring the voltage over the resistor, you can place the resistor in series with the cathode and anode. Make sure the positive lead is on the side of the cathode and the negative lead is on the side of the anode. Use gator clips to secure the connections

The charging test is simple. Secure the positive lead of the Arduino device on the cathode wire and the negative lead on the anode wire. Turn the Arduino device on so it is recording data. Touch the cathode and anode wires together and hold them for a few seconds. This should drain the voltage of the fuel cell. Allow the fuel cell to build its voltage up and the device will monitor that for you.

You can see examples of my results for these tests in the Results section.

Finding the Active and Sleep Current of the Arduino Circuitry…

This is also not very difficult. To find the current draw of your circuit, you need to have your FTDI breakout and cable and some F-M wires with a single M-M wire.

Take 6 F-M wires and attach those to your Arduino Pro Mini FTDI pins. Plug your M-M wire into the VCC female header on the FTDI breakout. Insert the rest of the respective F-M wires that are attached to the Pro Mini so only one side of the M-M wire and the M side of the F-M wire connected to the VCC of the Pro Mini are not connected to anything.

You can now take some gator clips and attach the M-M wire to the positive lead of a multimeter and the M side of the F-M wire to the negative lead. Set the multimeter to the current reading mode of DC. Plug the FTDI cable into the breakout and PC.

You should see the active current when the Pro Mini is actively measuring (I changed the time interval of the code so the device would measure long to get a more accurate current reading). I got a 4.8 mA reading when using the single input. This is assuming you followed all the modifications and removed all the excess components from the PCBs.

You will see the sleep current when the device is not actively measuring. I got around 0.8mA for this.

Powering the Voltmeter via Battery…

To power the Pro Mini via battery, I used a 2000 mAh LiPo battery and a battery manager. I recommend these, but you can use different batteries. If using a LiPo battery, I highly recommend a battery manager to increase safety.

I plugged the LiPo into the BATT IN slot of the battery manager and then plugged in a non-connected JST wire to the SYS OUT slot of the manager. The wires of the JST connector were too thick to plug into the breadboard, so I created a plug out of 4 male header pins.

I first removed the middle two header pins with pliers. I then soldered the JST wires to the short end of the header pins. The battery can now be connected directly to the breadboard. 4 header pins were used as this is the perfect amount of space to insert the "plug" into the ground strip as well as the **RAW** pin of the Pro Mini. **Make sure you plug the battery into the RAW pin and then wire the VCC pin to the positive power strip of the breadboard.**

DO NOT PLUG BATTERY IN AS WELL AS USE THE FTDI BREAKOUT. THIS WILL DAMAGE YOUR PC AND CIRCUIT.

Building the Arduino Enclosure…

First watch these tutorials:
https://www.youtube.com/watch?v=G6PqEsKjxHM
https://www.youtube.com/watch?v=mKaJLWv1SCw

These should show you how to construct the enclosure, strip the thruster wires, and pot the penetrators. They explain it much better than I can. I will give you some things you should do while following the tutorials above that will help when recreating my enclosure:

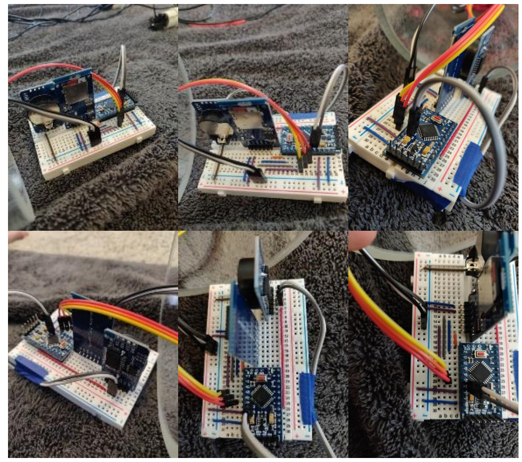
- Strip the thruster cable so about 3 inches of wire are exposed on the other side of the penetrator when it is inserted fully.
- Before potting the penetrator, fully construct the end caps by attaching and tightening all of you blanks, penetrators, and vents. Then pot the penetrators while they are already inserted into the end cap. This will give you a sturdy place to pot them if you do not have a vice, and then you do not need to try and tighten them with a cable in the way.

To build the enclosure like I have, you will need 18-22 AWG wire quick connects. If you are reading this part, I am assuming that you have already potted the penetrators and have attached the O-rings to the end caps and vents by following the tutorials linked.

1. Decide which wires will be your cathode wires and which will be your anode wires…
    a. For me, this was simple. I had three fuel cells, so three cathodes and three anodes. I denoted one of my two thruster cables cathode wires only and vice versa for the anodes. I then simply matched the colors to match a pair of electrodes.
    b. I recommend cutting a red heat shrink in thirds (or in even pieces related to how many fuel cells you are measuring) so you have three heat shrink tubes you can shrink onto the three cathode wires. This will be very helpful in the future!

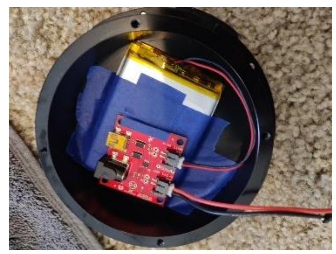
2. Attach the quick connects to the thrusters…
    a. Attach the sockets (The quick connect that folds over the wire) onto all the thruster cables regardless of cathode or anode classifications.

3. Modifying jumper wires…
    a. Since I was measuring three fuel cells, I used triplets stuck together of M-M wire.
    b. Remove any plastic housing near the tip of M-M wires.
    c. Slide the quick connect plugs (The ones that are crimped) onto the exposed metal points. Its okay if you use different color plugs and sockets, they all fit onto each other. Choose the plug that will be crimped easiest onto the exposed metal of the wire end.
    d. Crimp these to the wire ends so you have two triplets of wires with one end that can go into the breadboard and one end that can be quick connected to the thruster cable.



4. Insert the modified wires into the breadboard…
    a. I inserted one triplet into the A0, A1, and A2 analog pin rows.
    b. The other triplet was inserted into the ground strip.
    c. You can remove the wires that were acting as your Arduino voltmeter wires as well. So only the triplets are being used to measure voltage. See the images below. The Black-Gray-White triplet was for my ground and my Yellow-Orange-Red triplet was for my analog pins.
    d. You can now connect the Pro Mini to the thruster cable by connecting the quick connects.

5. Secure the battery and charging circuit…
    a. To save space, I taped the LiPo onto the bottom of an end cap. I then attached the battery manager circuit on top of that and had the battery plug attach to the breadboard from there.



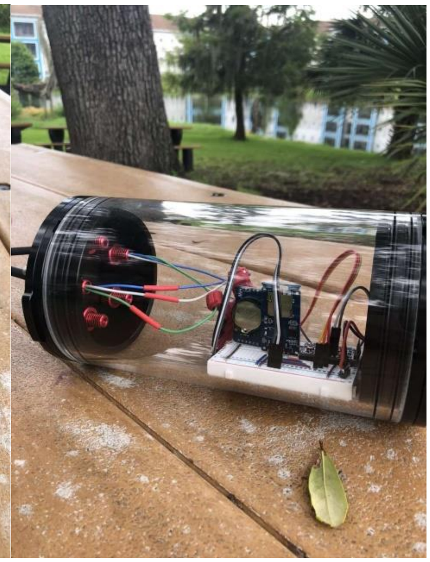
6. Upload the proper code…
    a. Depending on how many fuel cells are being deployed, upload the proper code to the Pro Mini and make sure it is running correctly on the PC.

7. Attach your fuel cells…
    a. Solder and heat shrink your fuel cell cathode and anode wires to the other side of the thruster cables.
    b. Keep your cathodes and anodes consistent!
    c. I am not going into details for this part since it is entirely dependent on your fuel cell design.

8. Complete the enclosure…
    a. Connect the quick connects and open the vent. Close the end cap with the penetrators over the acrylic tube.
    b. Plug the battery into the breadboard and reset the Pro Mini to make sure the program is running correctly.
    c. Close the battery endcap to completely close the enclosure tube.
    d. Close the vent.

9. You are done!

a. The enclosure is completely closed and waterproof (if you followed the video tutorials above as well).

Fuel Cell Related Procedures

Creating a Deployable Fuel Cell…

This entire procedure list is based on fuel cell design. Keep that in mind to adjust whatever you need to fit your design!

1. Prepare your brushes and wires for soldering…
   a. For this you should cut your wires to a uniform length (2-3 feet is sufficient) and strip the ends.
   b. If you have different colors of wire available, Great! You should decide on half of your electrodes being cathode and half being anode. Simply assign one color wire for each and use those wires with the respective electrodes you want designated as anode or cathode. If you only have one type of wire color, no worries! We will address this in future steps!
   c. I twist the exposed wires to create a more uniform strand.
   d. Wrap the wire around the electrode stem and solder them together.
   e. Put to the side and let cool while you do the next one.
   f. After you complete the next wire-brush connection, move the cooling brush into a storing box and move the new wire-brush to the cooling spot.
   g. Repeat this until all brushes are soldered.
   h. Make sure to test your connections with a battery and multimeter!

2. Cut your PVC pipes…
   a. According to your design, cut the ¾ inch PVC pipe and 3-inch pipe to your desired height.
   b. I cut my ¾ inch PVC into 5/4-inch lengths and my 3-inch PVC into 5/2-inch lengths. Make as many ¾ lengths as you have brushes and as many 3-inch lengths as you have chambered designs (if you are making them).

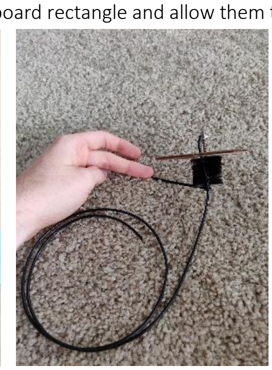
c. You will need to sand the edges of the 3-inch PVC after cutting with hacksaw.


3. Seal cut ¾ inch PVC pipes with cling wrap and masking tape…

a. This is to prevent any leaking when applying epoxy to PVC pipes.


4. Prepare Brushes for Epoxy…

a. Zip tie the wire to the tip of the titanium brush stem. This will help keep the brush straight in the epoxy
b. Cut rectangles from old cardboard. Cut very slim slits into the cardboard that are just wide enough you can fit the brush stem into it, but it will not wiggle.
c. Slide brush into slit.
d. Add tape to the sides of the cardboard rectangle and allow them to curly upwards.


5. Prepare for Epoxy…

a. Create a workstation if you do not have one. I covered a living room table with old cardboard.
b. Follow directions on epoxy container to prepare desired amount.

6. Set in Epoxy…

a. Pour the epoxy into the open side of a ¾ inch PVC pipe segment leaving between ¼ inch and 1/8 inch from the top of the epoxy to the lip of the PVC pipe.
b. With one hand, firmly hold the PVC pipe while you use the other hand to place the cardboard-brush assembly into the center of the epoxy filling.

c. Firmly press down on the cardboard sides while making the tape taut between the cardboard and the workstation table.
d. After applying the tape, you can adjust the cardboard slightly to try and center the brush in the PVC pipe if it is not centered anymore.
e. Do one at a time to make sure you do not make a mistake! Repeat this for all your brushes.



7. Allow the brushes to cure for specified time on the epoxy directions…

8. After curing, remove the tape and cardboard from all the brushes. Remove all the tape and cling wrap as well…
    a. Your finished product should resemble the pictures below. The picture on the right is the bottom view of the PVC.


9. Heat Shrinking the Electrode wire…
    a. If you do not believe your wires will be long enough after the electrodes are inserted into their respective designs in an enclosure, this is an easy fix.
    b. Simply strip the ends of the wire from the electrode and an additional wire length of your desired length. Twist the lengths around each other and solder them together.
    c. Slide a heat shrink sleeve around the solder point and use the heat gun to fit it properly.
    d. If you did not solder any additional wire lengths, I still **HIGHLY RECOMMEND** that you label your electrodes as cathodes and anodes using black and red heat shrink. You can ignore this if you have two wire colors and designated cathodes and anodes in step 1. If you are using the black wire I suggested, you only need to heat shrink the red sleeves onto the electrodes you are denoting as cathode. This will help you greatly in your organization.

10. Construct the Chambered Design… (Skip this if you are not following this design)
    a. Insert a 3-inch PVC length into the 3-inch x 1.5-inch coupling.
    b. Insert a 1.5 inch to ¾ inch bushing into the top of the coupling.
    c. Insert a designated anode (see last step) wire into the bushing and feed it through.
    d. Insert the anode PVC housing upside down so the brush is facing the bottom of the chamber.
    e. Insert an elbow on the exposed ¾ inch PVC pipe.

f. Insert the cathode brush out of the elbow PVC pipe.

11. Construct the Unchambered design…

a. Use the pictures above as a guide.
b. Insert the back end of an anode PVC piece into an elbow or vertical PVC coupling.
c. Insert the back end of a cathode PVC piece into an elbow or vertical PVC coupling.
d. The finished design should look like a chamber design without the chamber in the first picture.

12. Add the quick connections…

a. Add the quick connects to the ends of the anode and cathode wires. You **MUST** stay consistent with what end gets a plug and what end gets a socket. My cathodes always got the socket, and my anodes always got the plug.
b. If you are not consistent, it will make monitoring the fuel cells much more annoying, and it will make it impossible to connect all your cells in series!
c. If you purchased the connectors I recommended, here is how you apply the connectors:
    i. If you are applying a plug (the piece you slide the wire into), strip the wire and twist the expose wire to make it easier to thread through the plug hole. Once inserted into the plug hole, crimp the plug with a set of pliers.
    ii. If you are applying a socket (the piece that is folded), simply place an unstripped wire into the metal prongs, and use the pliers to fold the piece over until it snaps.
d. Make sure you check these connections by making a dummy pair of quick connects that have a stripped end on the other side of the wire. You will also need these to test the voltage of the fuel cells in the future!

Indoor Deployment…

1. Assign your designs to their respective containers…

a. Label the containers with what variables they are assigned with.

b. For example, for a container using salt water, unchambered anode, and additive, I labeled it “S/U/A”

2. Figure out where you are putting the finished containers…

a. I used the same table that I cured the epoxy on.
b. I divided my table into sections using masking tape so that there were 10 sections on the table. (5 rows of 2)
c. Set up your air pumps so that they can reach the desired container. Use the power strip to plug the air pumps in so you can keep all the air pumps together.
d. I kept all my air pumps on one side of the table, and I had the tubing go between all the containers.

3. Gather your sediment and water samples…

a. I used the shovel to gather the sediment samples and placed it into a 17-gallon container.
b. I dipped the second container into the water directly to get the needed water samples.
c. Make sure to adjust your water sample with salt if you are varying salinity levels in different containers!
d. Make sure to homogenize the samples by stirring them!

4. Add the additive if needed…

a. If you are adding an additive to a container, insert a small amount of sediment on the bottom of the container and place the additive on top of that, twisting it into the sediment to secure its position.

5. Add the sediment into the container…

a. Fill the sediment up in each container to a uniform level. I added about 5 inches of sediment into the containers with unchambered electrodes, and about 4 inches of sediment into containers with chambered electrodes.

6. Constructing the chambered design container…

a. Add about 2 inches of water into the container.
b. Remove the anode PVC from the bushing on the chambered design so you have a chamber in one hand and essentially the unchambered design in the other hand. Make sure you keep the wire end through the chamber, so it looks like the second picture of step 10.
c. Insert the chamber into the sediment up until the point where if the brush is inserted, it would be slightly above the sediment. **Always keep the quick connect out of the container! Do not take any chances and do not allow it to touch any water or sediment.**
d. If the chamber is not under water at this point, add more water into the container until the entire chamber is under the water level.

e. Once the water is above the bushing hole, reinsert the anode into the bushing so that the design is one piece again.
f. Fill the container with water until it is at the top.
g. Use the masking tape to create a make-shift cap. Leave room to insert the air pump.
h. Insert the air pump and close the rest of the cap with more tape.
i. Repeat this with all your chambered designs.

7. Construct the unchambered design container…
   a. Insert the anode directly into the sediment.
   b. **Always keep the quick connect out of the container! Do not take any chances and do not allow it to touch any water or sediment.**
   c. Fill the container with water.
   d. Use the masking tape to create a make-shift cap. Leave room to insert the air pump.
   e. Insert the air pump and close the rest of the cap with more tape.
   f. Repeat this with all your unchambered designs.

8. You are done!
   a. To check the voltage, I would take the dummy quick connects and attach it to the cathode and anode. I would then attach the gator clips to the stripped wire ends and the leads of the multimeter. Turn the multimeter on and you have your voltage reading! I did this twice a day for near a full month.
   b. You can do a bunch of interesting tests like discharge and charging tests where you monitor the behavior of the fuel cell while and after draining the voltage.
   c. Have fun!




Outdoor Deployment…

1. Complete your Arduino enclosure…
   a. Your fuel cells should be soldered to the thruster wires, and your battery should be plugged in, so the Pro Mini is running the code that is uploaded.

2. Bring all your supplies to the chosen deployment site…
   a. Choose your deployment site (I deployed very close to shore of a lake).
   b. Bring your enclosure with fuel cells, a cinderblock, zip ties or something to fasten the enclosure to the cinderblock, and a PVC pipe length.

3. Attach the enclosure to the cinderblock…
   a. I used the zip ties to attach the enclosure to the cinderblock to prevent the enclosure from floating.

4. Deploy the enclosure…
   a. Insert the enclosure into the water and place it on the floor of the body of water.

5. Dig holes for fuel cells…

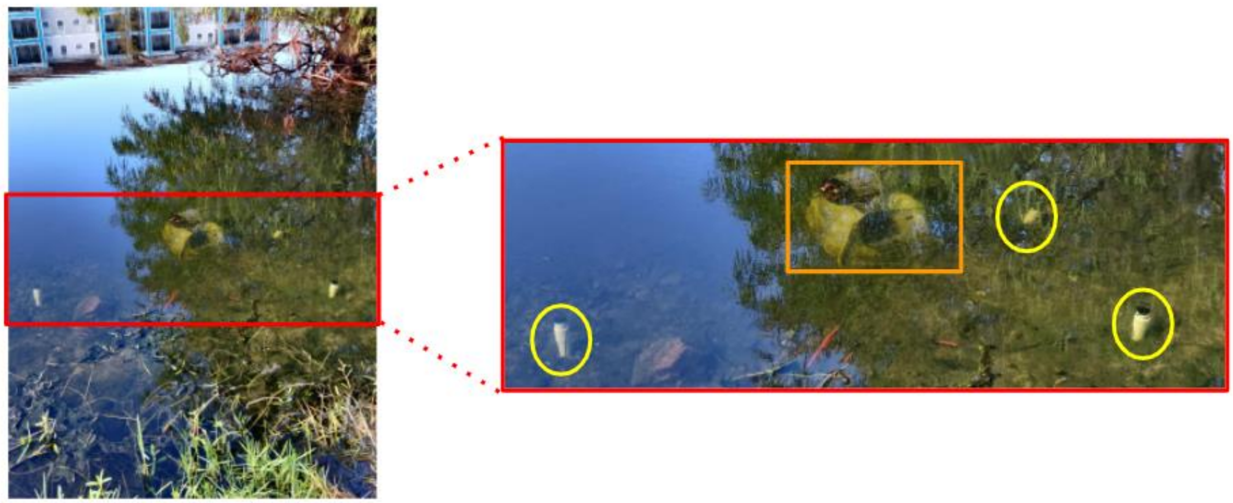
a. The sediment I was deploying in was compact, so I used a PVC pipe to dig holes which were made for the fuel cells.

6. Insert fuel cells into holes…
    a. Insert the fuel cells into the freshly made holes…
    b. Pack the dirt around the inserted fuel cell to secure it in place.

7. Leave for extended period…
    a. Allow the device to gather data for the period that you decided. Mine was running for only a week due to the end of my project.

8. Retrieve the device and analyze the data…
    a. Remove the enclosure and the fuel cells.
    b. Remove the SD card and insert into computer.
    c. Analyze data

9. You are done!



Recommendations

This section is entirely dedicated to recommendations I have for someone who wants to replicate or continue the research seen in the project. You can take my recommendations or ignore them. After all, it is **your** project!

General Recommendations

Variable Selection…

Variable selection is very important and will have a direct correlation to how many and what sized containers you would need for indoor fuel cells.

I personally think it would be interesting to see some data on various levels of salinity in fuel cell water instead of the two levels (freshwater and simulated ocean water) that I used.

I also think a promising idea would be introducing some type of chemical to the water or testing different additives to see how it effects the overall voltage. To be used in a in situ environment, the chemical/ additive would obviously need to be safe for animals, but you can ignore that in an indoor fuel cell.

Fuel Cell Design (Unchambered vs. Chambered) …

One of my largest regrets for my indoor fuel cells were using chambered anodes. I **HIGHLY** recommend against this. My reasoning for this is that they are significantly more difficult to deploy in a small container rather than an entire body of water. I will touch more on this in the next section. I believe that an indoor fuel cell should be unchambered, especially if using a relatively small enclosure.

If you are using 10-gallon fish tanks and you have a lot of room, then go for the chambered or unchambered. If you are working in 1-gallon containers like I was, then unchambered is the best option by far. The enclosure they are in is basically acting as a chamber due to how small the enclosure is, so adding a chamber to your anode is just going to make deployment and deconstruction more difficult.

If you are going to be in an in situ environment, then I highly recommend the chambered design; however, if you decide to use a chambered design, you should figure out a way to incorporate a check valve into the chamber. This is another very important design aspect I did not have the resources to do, and I wish I could have. This will prevent air bubble or even too much water in your chamber. This also allows for the escape of carbon dioxide.

Deployment is Key…

Now why do I recommend the chambered design in situ but the unchambered design for indoors? Deployment is the key here. If you do not properly deploy your fuel cells, there is a large chance the entire cell will not work. I have a strong belief that even if your fuel cell is deployed any way less than ideal, the voltages it will produce will suffer.

I recommend the unchambered design for an indoor enclosure because the sediment in an enclosure will be loose from homogenizing it, making it easy to insert the brush directly into the sediment. The chambered design is also very bulky, making it difficult to move around the enclosure when deploying it. This can lead to air bubbles stuck in the chamber (another reason for a check valve), which is not what you want. The harder to deploy, the less you want it.

I recommend the chambered design in situ mainly due to the more compact sediment found in the environment. You will not be uprooting and homogenizing sediment for a in situ deployment, meaning the sediment could very well be very compact. Trying to drive an anode directly into that hard sediment can lead to the harming or destroying of the anode before your data collection even begins. The chambered design is beneficial here since you are driving the chamber into the sediment, not the anode directly. The chamber will be easier to break into the hard sediment and should not be harmed in deployment.

When constructing you designs, deployment needs to be a top priority! ***MAKE SURE YOUR DESIGNS ARE NOT GOING TO FLOAT. ADD SAND INTO HOLLOW PVC PIPES. THIS AFFECTED MY DATA AS FUEL CELLS BEGAN TO DISLODGE FROM THE SEDIMENT DUE TO BOUYANCY.***

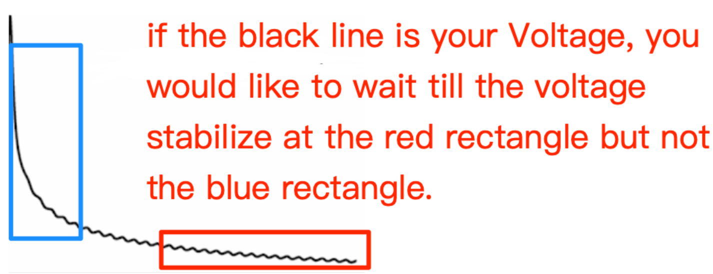
Indoor Deployment Recommendations

Data Collection and Analysis…

I was told by an expert in the field that I should always have connected the anode and cathode by a resistor. A very high resistance would be needed, but this would allow the fuel cell to continuously be creating power. Unfortunately, I was told this a little late into my project. I would recommend further looking into this idea, as I am not too certain about it. Something to think about!

Polarization Curves…

This is another of the larger regrets I have from my project. I never was able to construct a polarization curve. A polarization curve is what every literature paper based on BMFCs will have. You are going to want to construct multiple graphs relating to voltage, current density, and power density. To get the voltage data, you need to set up a fuel cell in series with a certain resistance and allow it to reach a stabilizing point (The red circle in the image below). That is where you take the voltage value with that resistance. You use the voltage and resistance to get the current and power. Those are used to get current and power density when compared to the anode surface area. Once again, I am almost certain this is how it works, but I never got a chance to do this. I would verify this is the correct way to do it before continuing forward with it; however, I **highly** recommend completing this.



Oscillating Tests…

This is another thing I never got to do. This is by far the most difficult recommendation, as it would require some serious modification to any of the code and Arduino hardware to work. The idea is to have an Arduino circuit switch between discharging on a resistor and allowing a fuel cell to charge without any human intervention.

For this to work, you would need some knowledge on transistors (I think). My idea was to use transistors to switch between a resistor and no connection to allow the fuel cell to discharge on the resistor then recharge on an empty connection. I have created a rough idea of how I think the circuit would need to be built for this. It is on the GitHub. Only an idea though, so it may not work. Just to get you started…

Closing Remarks

Well, here we are…

Do not worry, I do not have much else to add at this point.

I hope everything you needed was available to you in this packet. If you have any other questions, please reach out to Dr. Jordon Beckler or me, Emerick Gilliams. You can reach me at gilliamse@ufl.edu or 772-589-2628.

I hope you enjoyed learning about my project and how you can make your own!