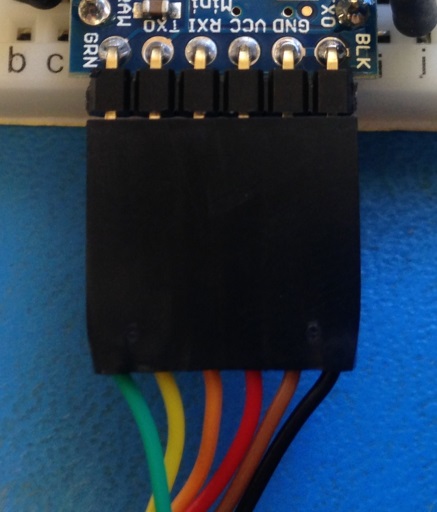
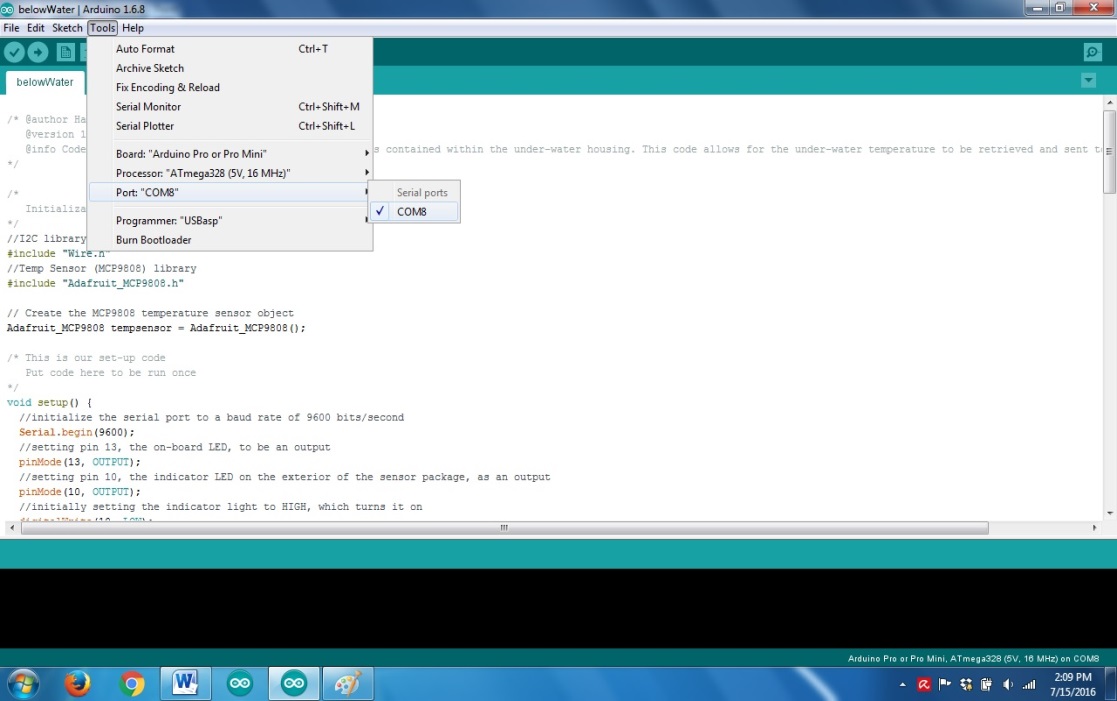
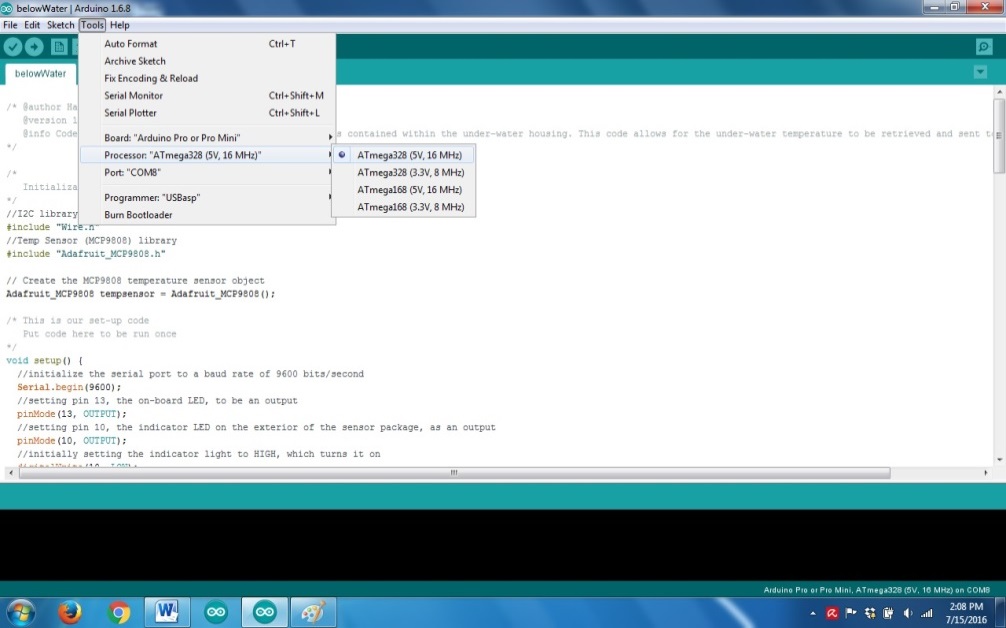
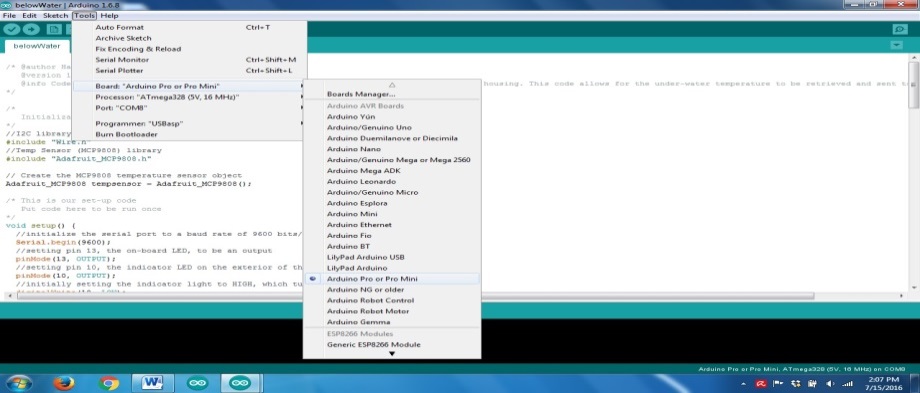
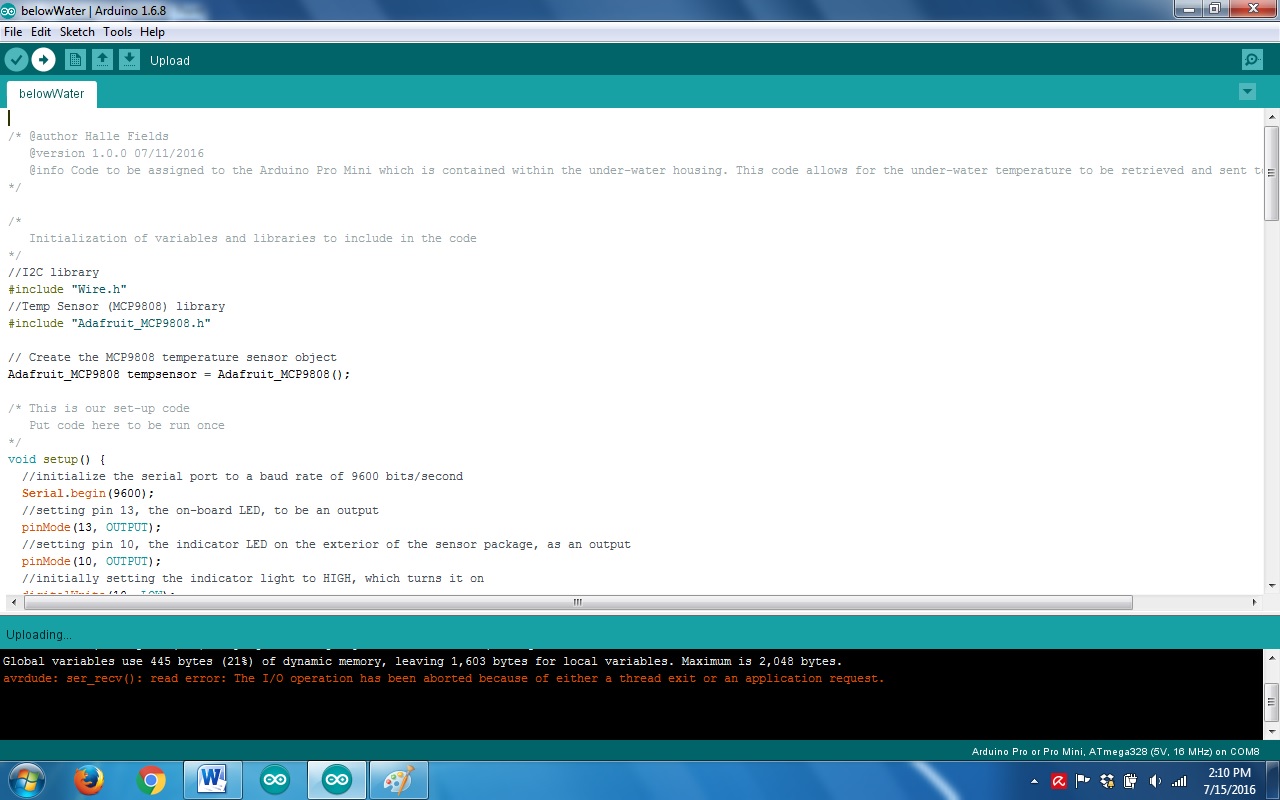
**Programming Components**

**Part 1: Programming the Under-water Arduino Pro Mini**

1. The programming portion of the under-water Arduino build requires:
2. Arduino IDE
3. The most recent version of the belowWater software
4. FTDI to USB cable
5. Arduino Pro Mini
6. A computer with a USB port
7. Plug the FTDI to USB cable in to your Arduino, so that when the Arduino is right-side-up, the cable has the green wire on the left and the black wire on the right.  
   The Arduino has BLK and GRN written next to their respective pins, so it’s easy to get this arrangement right.
8. Open the Arduino IDE and pull up the belowWater file.
9. Under the “Tools” menu:
   1. Select the “Board:” option, and change that to “Arduino Pro or Pro Mini”.
   2. Make sure that the “Processor:” says “ATmega328 (5V, 16MHz)”.
   3. Change “Port:” to whichever COM port your cable is plugged into on the computer.
10. Hit the upload button and you’re done!



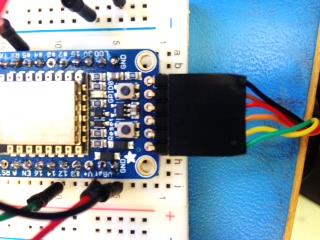


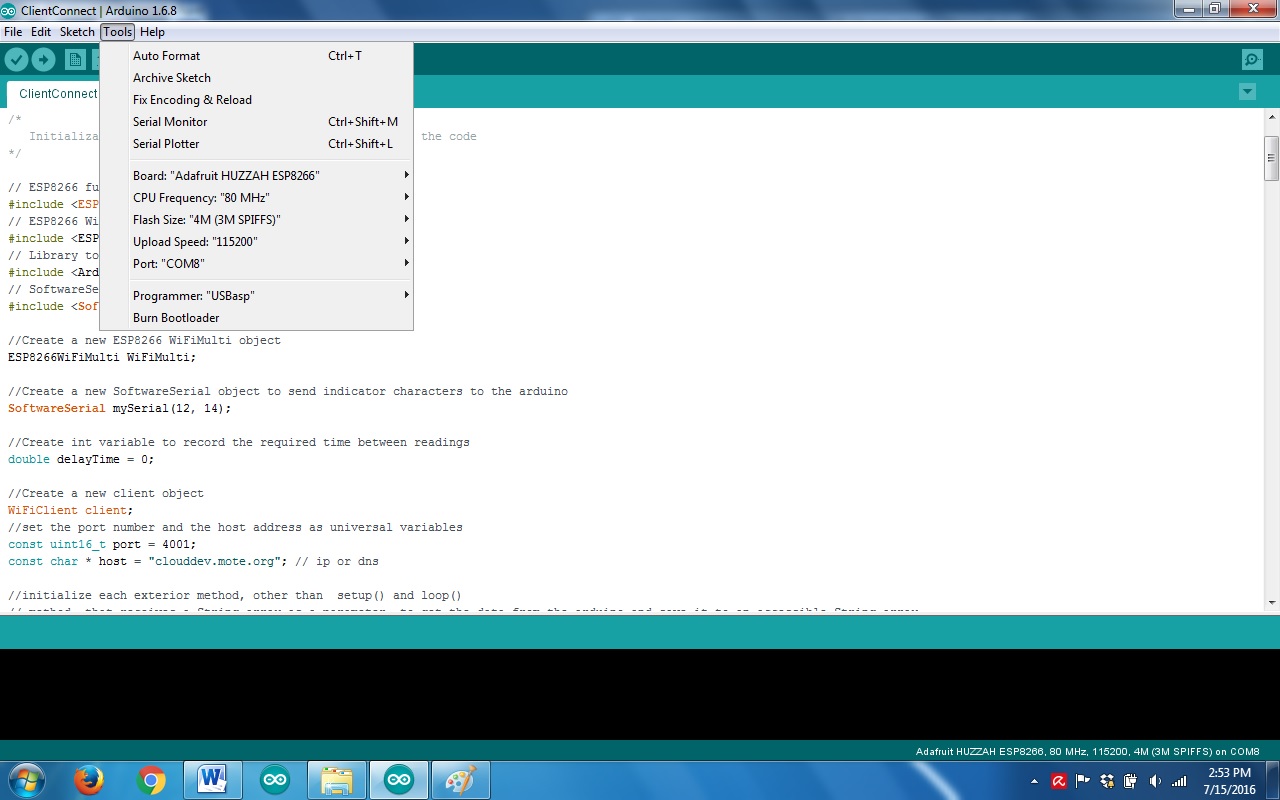


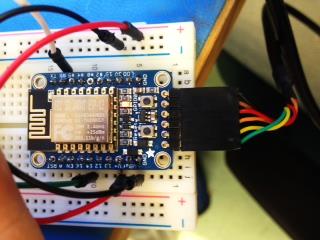
**Programming Components**

**Part 2: Programming the HUZZAH ESP8266 Breakout**

1. The programming portion of the above-water WiFi build requires:
2. Arduino IDE
3. The most recent version of the ClientConnect software
4. FTDI to USB cable
5. HUZZAH ESP8266 Breakout
6. A computer with a USB port
7. Plug the FTDI to USB cable into the HUZZAH, so that when the HUZZAH is right-side-up, the cable has the green wire on the left and the black wire on the right.
8. Open the ClientConnect file in the Arduino IDE.
9. Under the “Tools” menu:
10. Select the “Board:” option, and change that to “Adafruit HUZZAH ESP8266”
11. Make sure the “CPU Frequency:” is set to 80 MHz, the “Flash size:” is set to 4M (3M SPIFFS), and “Upload speed:” is set to 115200.
12. Finally, make sure the “Port:” option is set to your correct COM port.
13. You must put the board into Bootloader mode by pressing and holding the GPIO button, pressing and releasing the reset button, then releasing the GPIO button.
    1. The red LED above the GPIO button should be dimly lit if you did it correctly.
14. Press the upload button, and wait for the upload process to complete.







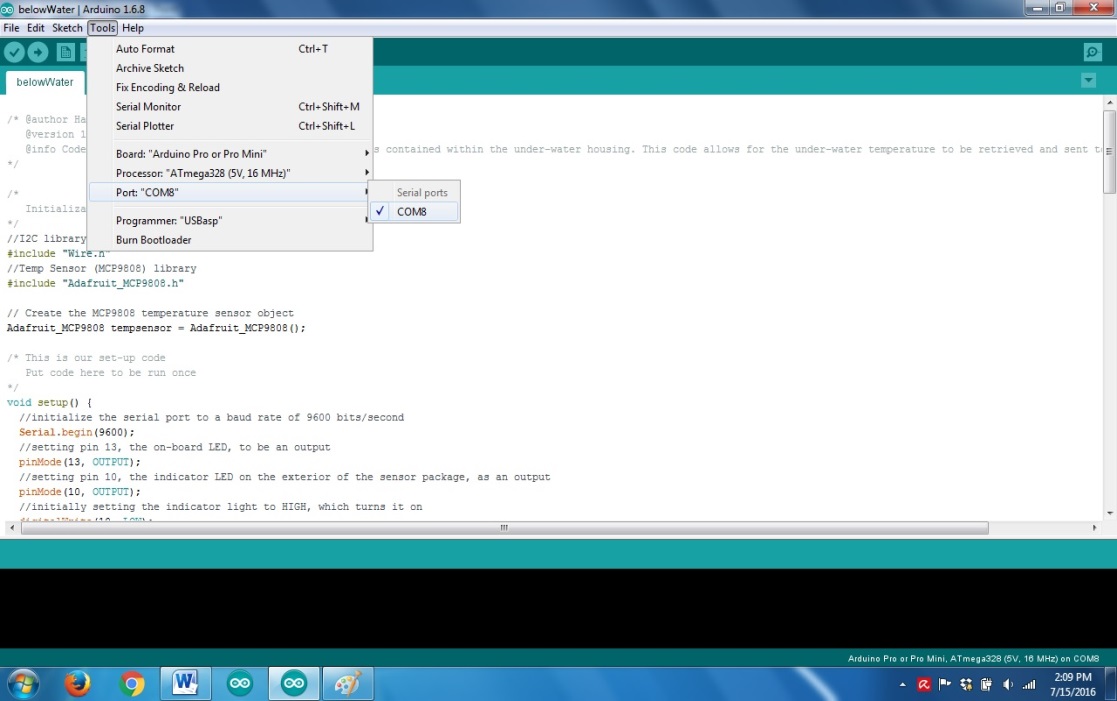
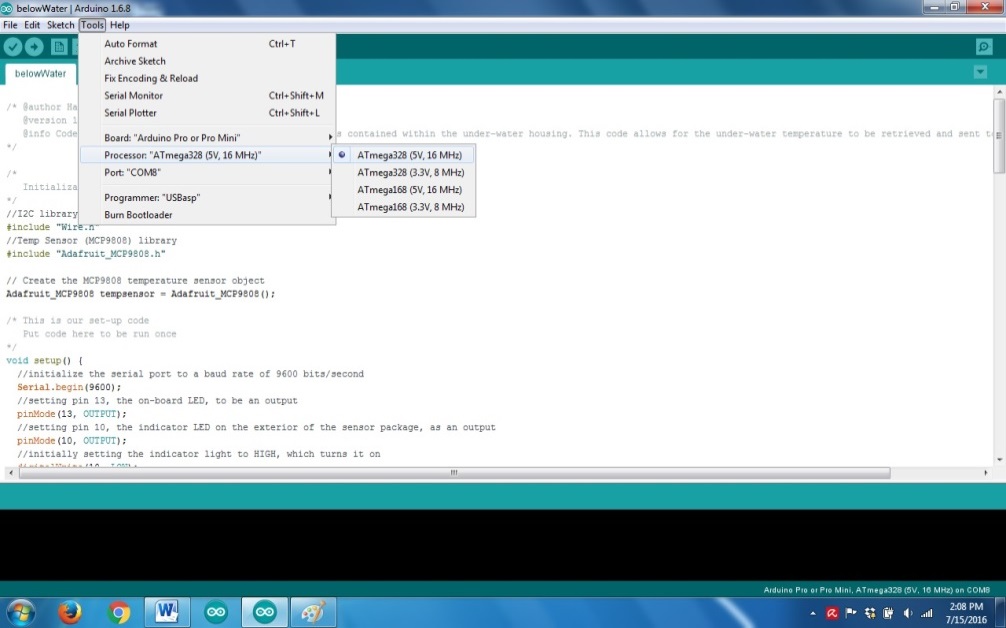
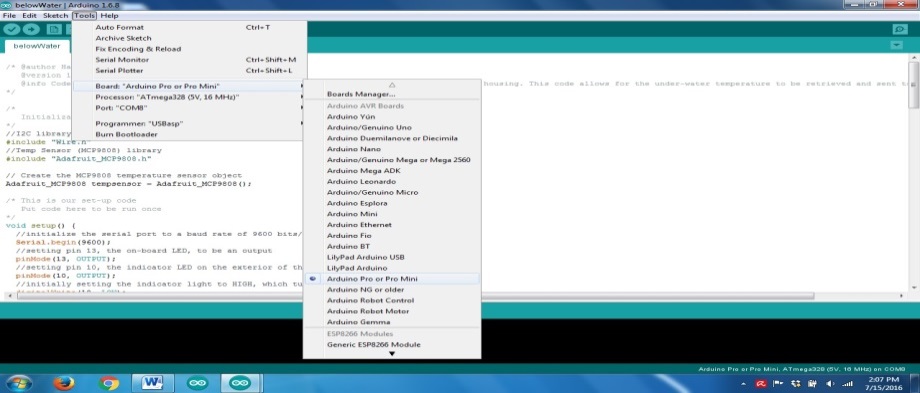


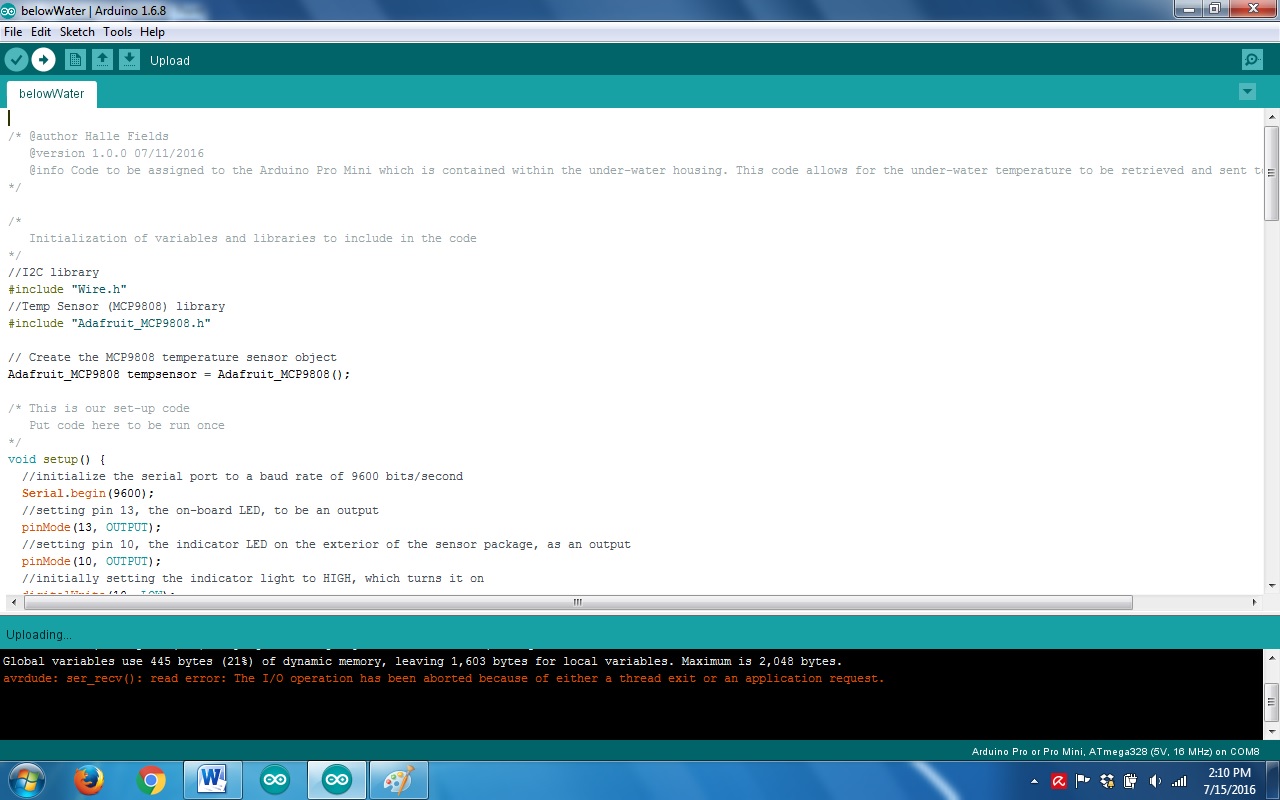
**Programming Components**

**Part 3: Programming the above-water Arduino Pro Mini**

1. The programming portion of the above-water Arduino build requires:
2. Arduino IDE
3. The most revent version of the avoceWater software
4. FTDI to USB cable
5. Arduino Pro Mini
6. A computer with a USB port
7. Plug the FTDI to USB cable in to your Arduino, so that when the Arduino is right-side-up, the cable has the green wire on the left and the black wire on the right.  
   The Arduino has BLK and GRN written next to their respective pins, so it’s easy to get this arrangement right.
8. Open the belowWater file in the IDE.
9. Under the “Tools” menu:
   1. Select the “Board:” option, and change that to “Arduino Pro or Pro Mini”.
   2. Make sure that the “Processor:” says “ATmega328 (5V, 16MHz)”.
   3. Change “Port:” to whichever COM port your cable is plugged into on the computer.
10. If you’re setting up the project for the first time, you’ll need to program the real-time clock. To do so, refer to this segment from the README file on your microSD card (next page).
11. Press the upload button and wait for the upload process to complete.

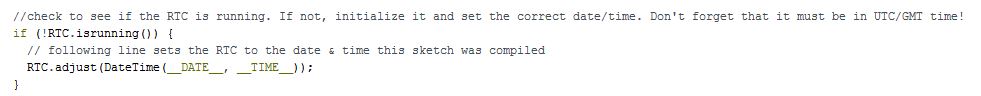






**Additional Instruction: Setting the time on your new Real-Time Clock**

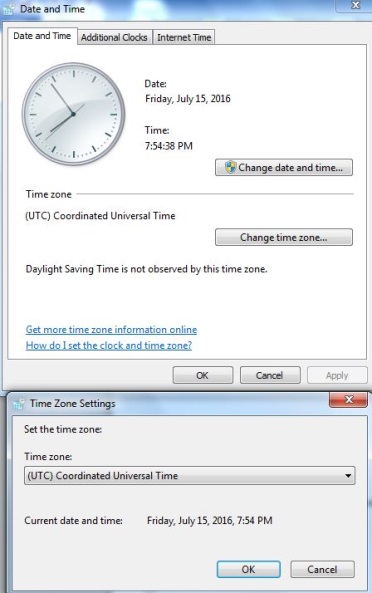
The RTC Time is in GMT time. Currently, if you update the code it will NOT reset the clock. This is because the segment to program the clock reads:

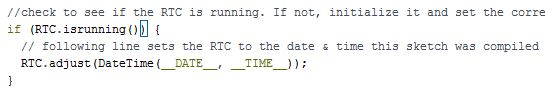


The first line makes it so the time will only be reset if the RTC turns off or malfunctions

If you need to reset the time for any reason:

* Put your computer in GMT (UTC) time to make sure the clock functions correctly



* Remove the exclamation point from the inside of the if statement, so it reads 
* Immediately replace the exclamation point, so it looks like it originally did, and then re-upload that code. This makes the clock not reset every time the device restarts.

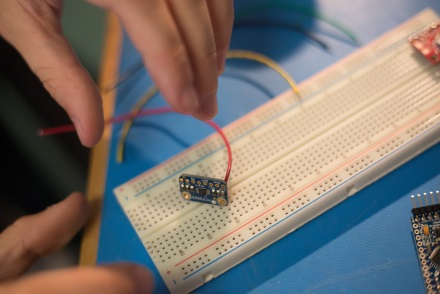
**Under-water Temperature Sensor Build**

**Part 1: Wiring**

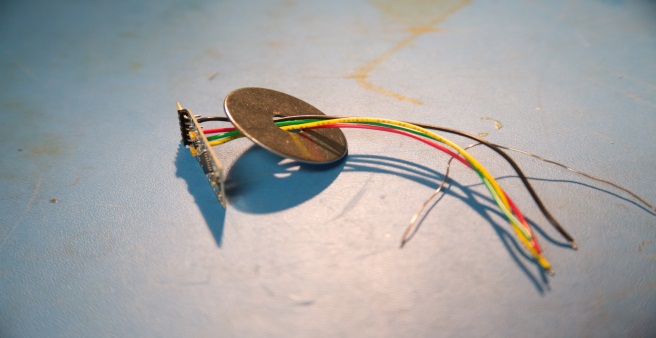
1. The wiring portion of the under-water temperature sensor build requires
2. Arduino Pro Mini (Programmed with the “belowWater” code)
3. MCP9808 Breakout
4. Green LED
5. 220 Ohm Resistor
6. Washer with 1.5 in diameter
7. 24 gauge wire, in colors red, black, green, yellow, and white
8. 6-pin-length of right-angle header
9. Heat shrink tubing
10. The tools you will need include:
11. Soldering iron and solder
12. Heat gun
13. Wire cutters
14. Wire strippers
15. Needle-nose pliers
16. Solder the right angle header on the Arduino Pro Mini and check to make sure your solder was successful.
17. Cut 5 inch lengths from the red, black, green, and yellow wire.
18. Strip about a quarter inch of the insulation off each end and twist the exposed wire gently to ensure they’re bundled together.

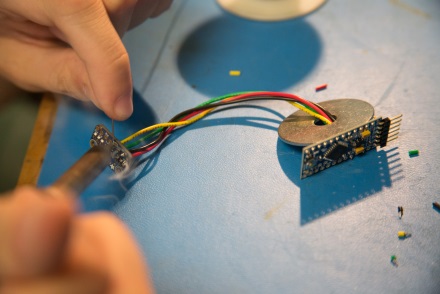


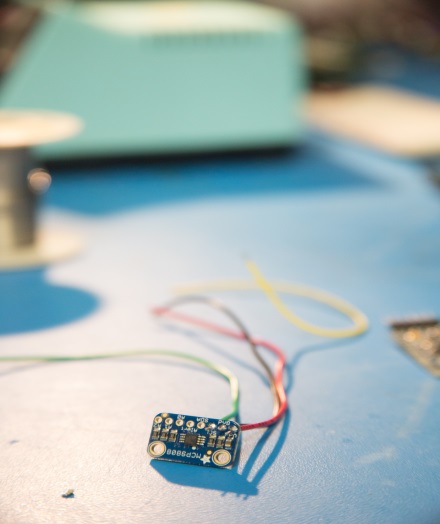




1. Tin each end of the wires using the soldering iron and the solder; check under a light and magnifying lens to ensure that the wire tip was tinned correctly.
2. Before you solder the wires in place, bundle them together and place them through the washer, so that one end will be attached to the sensor, the washer is in the middle, and the other end will be attached to the Arduino.
3. Place the end of a wire through its designated hole, through the back of the device, and solder the wire in place. Check your solder joint for proper flow.
   1. On the Arduino: the black wire will go into Gnd, the red wire will go into Vcc, the green wire will go in to A5, and the yellow wire will go in to A4.
   2. On the MCP9808: the black wire goes into Gnd, the red wire goes into Vdd, the green wire goes into SCL, and the yellow wire will go in to SDA.
4. Cut 4 inch lengths of the black and white wire, and strip a half inch of insulation off one end and a quarter inch off the other end of both.
5. Tin both sides of the wire and check that the tinning worked well.
6. On the half inch side of each wire, create a J-hook.
7. Retrieve your LED and trim the leads to be half an inch long.
8. Create a J-hook on both leads.
9. Cut the leads of the resistor so each is half an inch long, and create a J-hook on each lead of the resistor.
10. Hook the end of the resistor to the hook on the end of a lead of the LED (it doesn’t matter which one) and close the hooks so they’re secure.









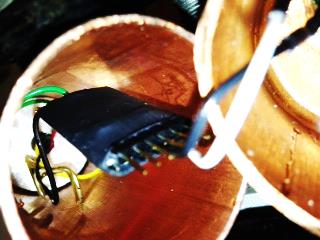


1. Hook the end of the black wire to the negative end of the LED, and the white wire to the positive end. Whichever end you put the resistor on, the wire will be hooked to the resistor, not the actual LED. Close the hooks.
2. Solder over each joint, and check each for proper flow.
3. Cut proper lengths of thin heat-shrink to place over each joint.
4. Shrink the tubing over the wire with the heat gun.
5. Solder the quarter inch end of the wire to its respective hole in the Arduino, and check each joint.
   1. The black wire (negative) goes to the other ground and the white wire (positive) goes to digital IO pin 10.
6. Cut all excess metal from each solder joint and check again that each joint is correct.
7. Take the large piece of heat shrink and cut it so it fits over the Arduino.
8. Arrange the wires so that the temperature sensor wires are pointing down, and the LED wires and the header are going up.
9. Heat-shrink the tubing over the Arduino, making sure that the headers are still exposed so that it can be connected.









**Under-water Temperature Sensor Build**

**Part 2: Housing**

1. The housing portion of the under-water temperature sensor build requires:
2. The part you built in Part 1
3. 1.5 inch diameter copper pipe
4. 1.5 inch diameter copper cap for the copper pipe
5. Hot glue
6. Epoxy
7. 4-wire bundle cable
8. Wire housings
9. Crimp connectors for the wire housings
10. RTV Sealant
11. PTFE Tape
12. Electrical tape
13. The tools you will need include:
14. Hot glue gun
15. Crimpers
16. Wire cutters & strippers
17. Pipe cutter
18. Drill with .199 inch drill bit
19. Much larger drill bit to clean holes
20. Vice
21. PVC pipe for proper washer height
22. Cut a 4 foot section of the bundle cable.



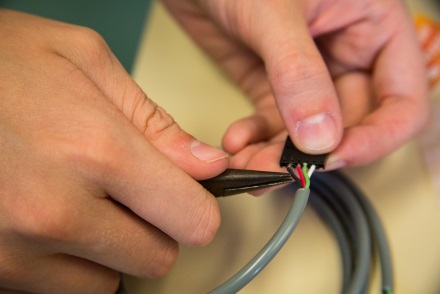
1. Remove a 1 inch section of the insulation from the outside of one end of the cable, so that the 4 wires are exposed.
2. Grab 4 crimp connectors and examine them closely.
3. Strip a section of wire about the length of the wire-containing portion of the connector. Twist those to contain them.
4. Crimp the end of the wire inside the connector using the crimpers.
5. Do that for each of the 4 wires.
6. Arrange the wires so they are in order from left to right: black, red, green, white.
7. With the tab on the connector pointing upwards, stick each connector end into the housing so that the wires are in the middle.   
   It should have an empty hole, black, red, green, white, and another empty hole.
8. Check that the cable is secure.

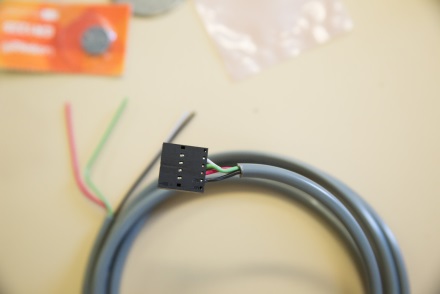








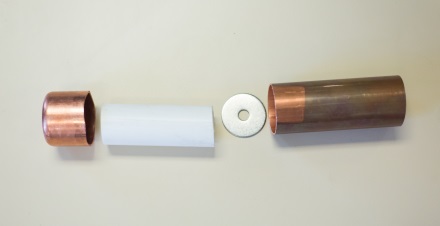




1. Mark a 4 inch section of the copper pipe and cut that using the pipe cutters. Deburr it.
2. Clean the pipe section with dish soap and water, and thoroughly dry. Set aside.
3. Drill two holes in the top of the copper cap, one in the direct center and one to the side of that.
4. Clean up the holes with the larger drill bit.
5. Test to make sure that the holes are the correct size for the LED (to the side) and the cable (in the center).
6. Feed the Arduino and its cables through the PVC sizing pipe, so the temp sensor is on top of the washer, which is on top of the pipe.
7. Check that the washer is level and then put a ring of hot glue around the perimeter of the washer, to secure it and also make it leak proof. You can now remove the PVC.
8. Use hot glue to secure the temperature sensor, level, on the top of the washer.   
   ****









1. Use hot glue to push the temperature sensor wires to one side, and seal the hole in the center. Now, that area should be leak proof. Check to make sure there would be no holes between the top of the washer and inside the tube.
2. Secure the copper pipe inside a vice, level.
3. Pour pre-mixed epoxy in the space above the washer, fully covering the temperature sensor. A thick layer should cover the surface of the washer.

This is important because this is what is water-proofing our sensor!

1. Allow to cure for 24 hours.
2. Once the epoxy is fully hardened, flip the tube over and feed the cable through the hole in the cap.
3. Attach the connector to the Arduino and put the LED in its designated hole.
4. Secure the items in place and leak-proof the holes with RTV sealant. Once you’re sure it is leak-proof, secure the lid in a vice, and pour a thin layer of epoxy into the lid. This helps water-proof the holes in the top.
5. Once that has dried, your housing is complete and should be ready to seal up and go for a swim!
6. To seal, place a layer of PTFE tape around the pipe, secure the cap, and put a ring of RTV sealant around the outside perimeter of the cap. Allow to dry. Place a ring of electrical tape around the deal for good measure.

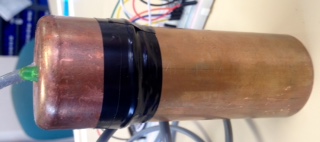












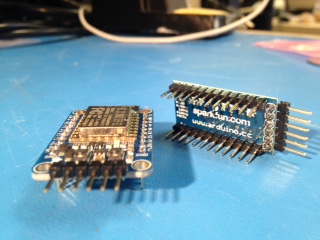
**Above-water Control Module and WiFi**

**Part 1: Wiring**

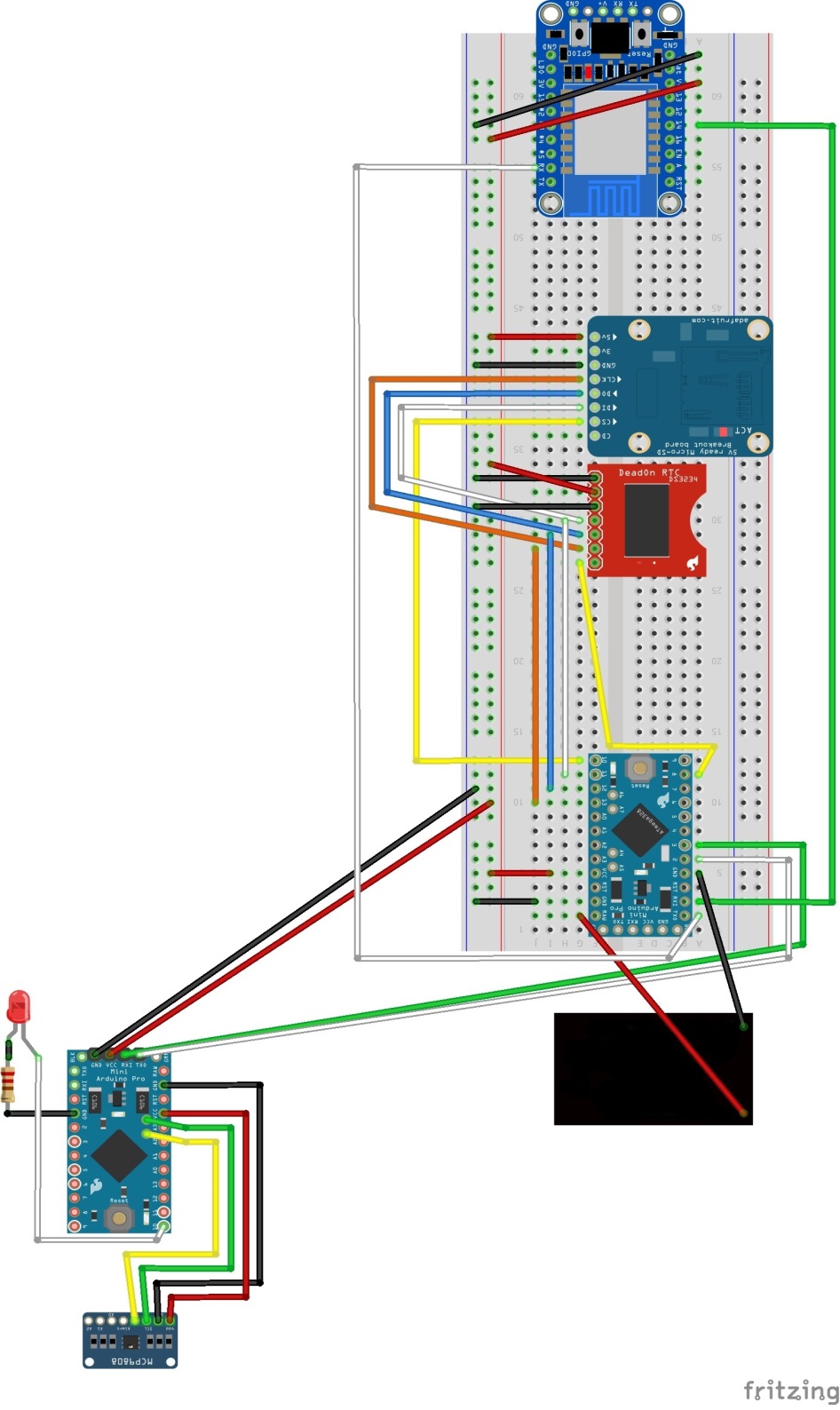
1. The wiring portion of the above-water control module build requires:
2. Jumper Wires
3. Half-sized breadboard
4. Straight-pin headers
5. Arduino Pro Mini (Programmed with the “aboveWater” code)
6. DeadOn Real-Time Clock
7. Adafruit Micro-SD Breakout
8. HUZZAH ESP8266 Breakout (Programmed with the “ClientConnect” code)
9. The tools you will need include:
10. Soldering iron and solder
11. On the end of the cable, coming out the under-water sensor, strip half an inch of wire from the end of each wire in the bundle.
12. Tin the end of the wire and make sure it is smooth, they will need to fit inside the breadboard holes.
13. Optional: put a few small pieces of heat-shrink on the joint to stabilize the wire and prevent breaking.
14. Solder the headers into each hole of the 4 boards. The right-angle headers go on to the front end of the HUZZAH and the Arduino, for easy programming. The rest of the pin holes should get regular straight-pin headers, so they fit into the breadboard.

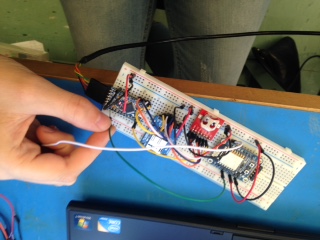






1. Arrange the boards on to the half-sized breadboard according to the attached diagram, and then place jumper wires in the correct places to create the circuit.



…

**Above-water Control Module and WiFi**

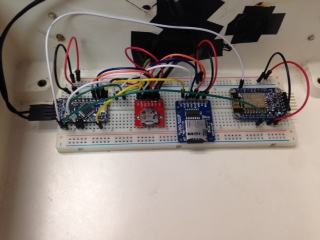
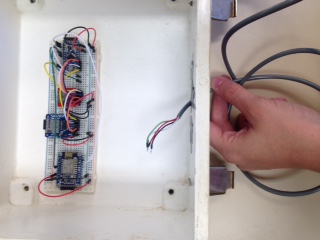
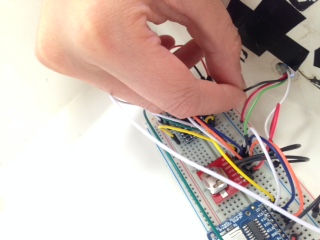
**Part 2: Power**

1. The power portion of the above-water module requires:
2. Solar panel with attached cables
3. Solar panel housing
4. 12 V Battery
5. 12V to 6V Voltage regulator
6. 2 Diodes
7. Battery cables
8. The tools you will need include:
9. Screw driver
10. Soldering iron and solder
11. 7/16 inch wrench

**Above-water Control Module and WiFi**

**Part 3: Housing**

1. The housing portion of the above-water control module build requires:
2. The provided housing container
3. Hot glue
4. Electrical tape
5. The breadboard circuit you laid out previously
6. Power source (battery)
7. Battery cables
8. WiFi Hotspot
9. Power cable for hotspot
10. The tools you will need include:
    1. Hot glue gun
11. Peel the adhesive backing off of the breadboard and place it firmly to the bottom of the container, leaving enough room on either side for the power source and WiFi Hotspot.
12. Place your required cables through the appropriate holes in the side of the housing.
13. Plug all cables in to their breadboard spots and verify that your set-up is working before proceeding.

1. Waterproof both around the cables and over the holes that are unused using glue and electrical tape. Make sure this is sealed properly.
2. Check the housing for any places where it would not be weather-proof, and verify that the housing is ready to face the elements.

