**Arduino pH sensor project**

Amelia Ritger

A picture containing outdoor, sky, water, sea

Description automatically generated

This work is licensed under the [Apache License 2.0](https://github.com/ameliaritger/arduino-ph-sensor/blob/main/LICENSE).

**Contents**

1. Introduction
2. Materials, Tools, and Consumables
3. Voltage divider circuit schematic
4. Component layout on FeatherWing
5. Preparing the Arduino
6. Preparing the battery pack
7. Communicating with the Arduino
8. Deploying the Arduino

**1. Introduction**

Ocean acidification monitoring efforts are a crucial component of tracking the impacts of climate change in marine ecosystems. However, the high instrument cost presents a major barrier to the production of scientific knowledge and has resulted in a highly fragmented understanding of the global progression of acidification. Our project built upon an existing nearshore marine pH sensor design which utilizes a Durafet pH electrode. Our goal was to lower barriers to access by significantly lowering the cost of, and improving approachability to, the design and use of pH sensor electronics. We have created a more compact design using open-source components based on the popular and easy-to-use Arduino platform that eliminates over $900 from the cost of the sensor electronics. We demonstrate with lab and field testing that switching to Arduino-based sensor electronics maintains high data fidelity. Our design supports open science by allowing more individuals and research groups to engage in high-quality oceanographic research.

This document provides guidance on the construction of the data-logger electronics associated with the Durafet-based pH sensor. *This document is a work in progress, please stay tuned for updates!*

**2. Materials, Tools, and Consumables**

|  |  |  |
| --- | --- | --- |
| **ITEM** | **QUANTITY** | **PRICE** |
| **Tools** | | |
| Wire strippers | 1 | TBA |
| Soldering iron | 1 | TBA |
| Multimeter | 1 | TBA |
| Needle nose pliers | 1 | TBA |
| Flush cutters | 1 | TBA |
| Solderless breadboard | 1 | TBA |
| **Materials** | | |
| Adafruit Feather M0 Adalogger | 1 | [$19.95](https://www.adafruit.com/product/2796) |
| Adafruit FeatherWing Proto | 1 | [$4.95](https://www.adafruit.com/product/2884) |
| Adafruit ADS1015 12-bit ADC | 1 | [$9.95](https://www.adafruit.com/product/1083) |
| Adafruit ADS1115 16-bit ADC | 1 | [$14.95](https://www.adafruit.com/product/1085) |
| Adafruit DS3231 Precision RTC | 1 | [$17.50](https://www.adafruit.com/product/3013) |
| 2-AA battery holder | 1 | [$0.53](https://www.digikey.com/en/products/detail/nte-electronics-inc/23-BH2-3/16499222) |
| AA batteries | 2 | TBA |
| micro-USB cable | 1 | TBA |
| microSD card (256 - 512 MB or higher) | 1 | TBA |
| microSD card reader | 1 | [$5.95](https://www.adafruit.com/product/939) |
| 1 M resistor | 5 | TBA |
| 15 kΩ resistor | 1 | TBA |
| 10 kΩ resistor | 2 | TBA |
| 20 μF capacitor | 1 | TBA |
| 100 μF capacitor | 3 | TBA |
| **Consumables** | | |
| Lead-free solder | 1 | TBA |
| Electrical tape | 1 | TBA |
| 22 AWG solid core wire | 1 | [$2.95](https://www.adafruit.com/product/290) |
| 22 AWG stranded core wire | 1 | [$2.95](https://www.adafruit.com/product/2997) |
| Crimps | 10 | [$2.29](https://www.digikey.com/en/products/detail/te-connectivity-amp-connectors/104479-8/) |
| 90 degree male pin header (3 position) | 1 | [$1.08](https://www.digikey.com/en/products/detail/te-connectivity-amp-connectors/103634-2/291595) |
| Female pin socket (3 position) | 1 | [$0.55](https://www.digikey.com/en/products/detail/te-connectivity-amp-connectors/104257-2/) |
| Vertical male pin header (6 position) | 1 | [$0.85](https://www.digikey.com/en/products/detail/te-connectivity-amp-connectors/103638-5/) |
| Female pin socket (6 position) | 1 | [$0.59](https://www.digikey.com/en/products/detail/te-connectivity-amp-connectors/104257-5/) |
| Stacking female headers | 1 | [$1.25](https://www.adafruit.com/product/2830) |
| Nylon screws | 1 | [$16.95](https://www.adafruit.com/product/3299) |
| Dupont cables (M-M) | 1 | [$1.95](https://www.adafruit.com/product/1956) |
| Dupont cables (M-F) | 1 | [$1.95](https://www.adafruit.com/product/1953) |

**3. Voltage divider circuit schematic**

The voltage divider, originally designed by Gernot Friederich at Monterey Bay Aquarium Research Institute, joins the Durafet pH electrode to the ADC inputs. The resistor values used are 1 M (R1, R4, R5, R6, R7), 10 kΩ (R2, R3), and 15 kΩ (R8). The capacitor values used are 20 μF (C1) and 100 μF (C2, C3, C4), although substituting C1 with a 22 μF capacitor is acceptable.

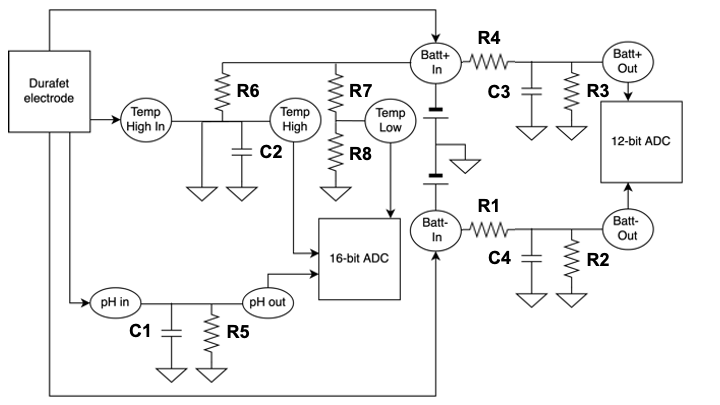
****

Fig GG. Circuit schematic of the voltage divider circuits joining the Durafet electrode to the ADC inputs.

**4. Component layout on FeatherWing**

Cost savings are maximized with the DIY build by soldering all components of the voltage divider. This is the most challenging soldering step of the build, as it requires a steady hand and thoughtful consideration of the order in which components are soldered. You must plan ahead with layering components as you solder, as some components overlap with wiring/solder bridge connections. We highly recommend laying out the components on a breadboard and testing connections with a multimeter and with the Arduino IDE prior to soldering to ensure the system is operating correctly.

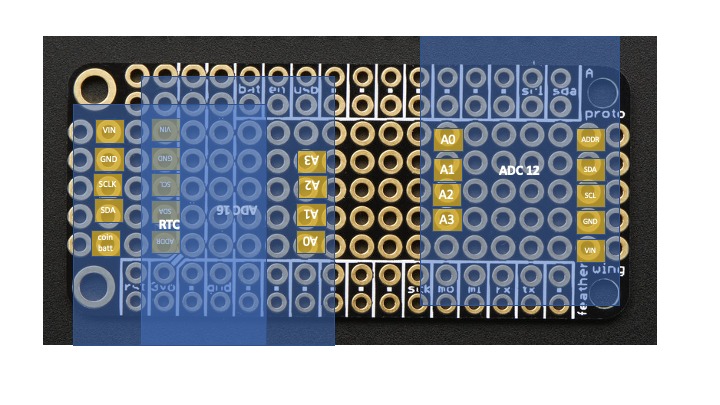


Fig AA. Diagram of ADC (ADS1115 and ADS1015) and RTC (DS3231) alignment on front side of FeatherWing.

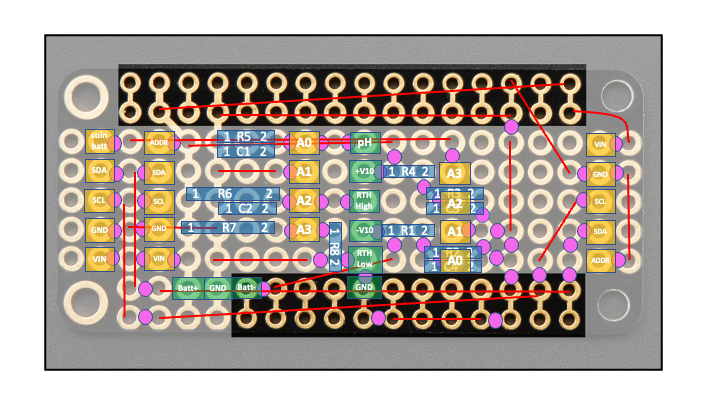
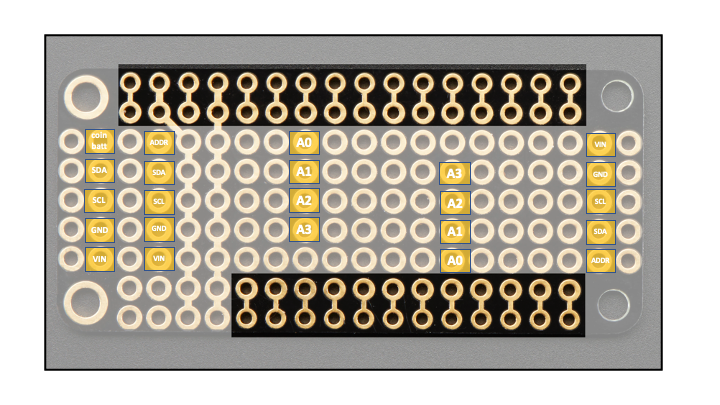


Fig BB. Diagram of all RTC and ADC pins (yellow), passive components (blue), PCB mount headers (green), and wiring (red)/solder bridge (pink) connections on back side of FeatherWing.

Fig CC. Alignment of only RTC and ADC pins on back side of FeatherWing.

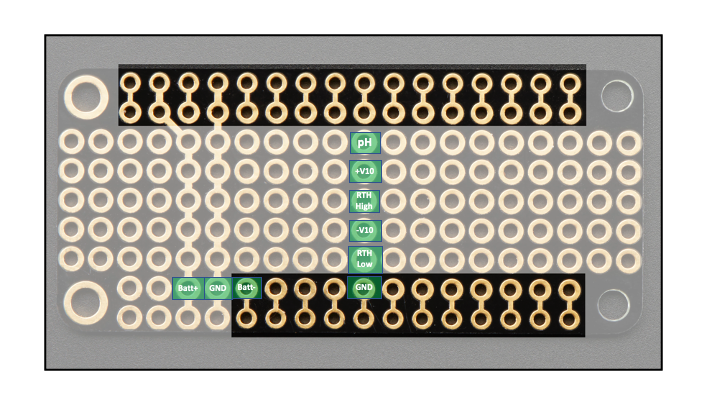


Fig DD. Alignment of only PCB mount headers on back side of FeatherWing.

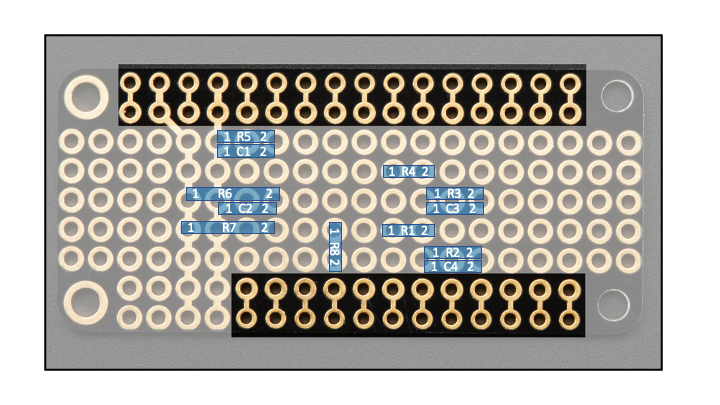


Fig EE. Alignment of only resistors and capacitors on back side of FeatherWing. See Section XX for details on component values.

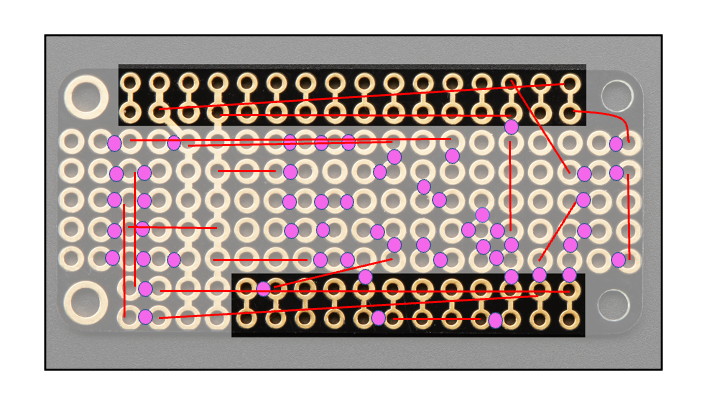


Fig FF. Alignment of wiring and solder bridges between pads on back side of FeatherWing.

**5. Preparing the Arduino**

This project uses an Adalogger M0 paired with a real-time clock (RTC) and two analog-to-digital converters (ADCs). Together, these components allow users to log timestamped Durafet data onto a microSD card. The ADCs have differential channels which allow the Adalogger to read negative voltages from the Durafet's pH and battery inputs.

The following paragraph has been copied over from the Oceanography for Everyone OpenCTD project documentation, as they also use the Adalogger M0 and have done an excellent job of explaining the process of preparing the Adalogger for assembly. You can find the full documentation associated with OpenCTD, which includes step by step instructions and pictures of the soldering process, at the following link: https://oceanographyforeveryone.com/project/openctd/.

The Adalogger and RTC (*and ADCs*) will need headers. Headers are the male or female pin assemblies that allow you to connect and disconnect electronic. A “pin” refers to any electrical contact point and will generally be used in this guide to refer to the holes in circuit boards through which wires, headers, and other components will be soldered. You will be soldering components directly to a circuit board. Assembling the Adalogger and Real-Time Clock (*and ADCs*) is a good, low-risk, opportunity to practice your soldering skills.

**6. Preparing the battery pack**

Cut a piece of stranded core wire and solder it to the metal wire between the two battery cells. This wire is now your GND, the red wire is your Batt+ and the black wire is your Batt-.

**7. Communicating with the Arduino**

The following paragraph has been copied over from the Oceanography for Everyone OpenCTD project documentation, as they also use the Adalogger M0 and have done an excellent job of explaining the process of communicating with the Adalogger. You can find the full documentation associated with OpenCTD, which includes step by step instructions about the process of using the Arduino IDE with the Adalogger, at the following link: https://oceanographyforeveryone.com/project/openctd/.

To prepare your computer to talk to the M0 Adalogger, download and install Arduino IDE (https://www.arduino.cc/en/Main/Software). You will need to install additional boards so that Arduino IDE will be able to recognize the Adalogger. For a detailed walkthrough of this process, with screenshots included, please visit: https://learn.adafruit.com/adafruit-feather-m0- adalogger/setup. Please be aware that the Arduino, Adafruit, or Github Software may have been updated more recently than this manual and that the location of certain tools and functions may differ from the instructions.

You will also need to manually download and install the following libraries for this project:

* Adafruit\_ADS1X15.h
* ArduinoLowPower.h
* DS3232RTC.h
* Streaming.h
* Time.h
* SdFat.h

Please refer to https://www.arduino.cc/en/Guide/Libraries for detailed instructions for installing libraries.

You will want to use the serial monitor in Arduino IDE to verify that everything is working correctly. You should see messages about a successful SD card initialization and a successful creation of a new file and header on the SD card. You can then unplug the Arduino from your computer, remove the SD card, and confirm that data was recorded on the SD card.

**Deploying the Arduino ("arduinoCode")**

The project GitHub repository includes CONFIG.h and SRC.h files alongside a main arduinoCode.ino file, all of which are essential for proper communication and deployment. You should only be editing the CONFIG file for deployment.

To change the file name printed to the SD card, edit the contents within the parentheses on line 4. The filename must be equal or less than 8 characters and must have a .csv file extension. *Default: "filename.csv".*

To change the sampling interval, edit the number within the curly brackets on line 6. This number represents the frequency of sampling in seconds. For example, if you want sampling to occur once every 30 minutes, you will set this number to 1800. *Default: 600.*

To change the deployment start time and date, edit the numbers within the curly brackets on lines 7-9. These numbers represent start date (line 7), and start time in hours (line 8) and minutes (line 9). For example, if you want sampling to occur on the third day of the month at noon, you will set these numbers to 3, 12, 0. Use the numbering intervals recommended in the comments (e.g., 0, not 00, to start sampling at the top of the hour). *Default*: *12, 0, 0.*

To match the RTC time with your computer time, edit the number on line 15. This "fudge factor" represents a number, in seconds, and allows for fine-scale tuning of the timestamp printed to the SD card. Open up the Serial while the Arduino is running to check the difference between timestamp printed on the Serial monitor (left) and the timestamp of the RTC that is printed on the SD card (right of the colon). Alter the fudge factor so that the value of the RTC timestamp aligns with your computer/Serial time, and run the code again to check for alignment. For example, if the Serial timestamp is 5 seconds ahead of the RTC timestamp, then add 5 to the "fudge factor". The fudge factor addresses differences in Arduino code upload times when using different computers - typically, faster processors will have smaller fudge factors (0-10) while slower processors can have fudge factor values higher than 15. *Note*: Uploading code to the Arduino on a new day will often take significantly longer than subsequent uploads, and so it is recommended that you upload at least twice to check your fudge factor is correct. *Default: 15.*

Your Adalogger is now ready for deployment!

**Processing the data ("rCode")**

After deploying your Adalogger, you will be able to collect the .csv data file from the microSD card. These data can then be imported into an R project alongside the R code found on the project GitHub repository for processing. There are four .Rmd files:

* Final\_deploy\_voltages.Rmd
* Final\_OMEGAS.Rmd
* Final\_standardize\_data.Rmd
* Final\_Tris\_Cal.Rmd