

Construction and Operation of the

OpenCTD



oceanographyforeveryone.com

Construction and Operation of the OpenCTD

Andrew Thaler, with contributions from Russell Neches, S. Kersey Sturdivant, and Ian Black.

Second Edition: March 2022.

Cover Art by Andrew Thaler.
Photographs by Andrew Thaler unless otherwise noted.



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

Contents

1 Overview	5
1.1 Building Skills	8
1.2 Project Status and Compatibility	9
1.3. Acknowledgements.....	9
2 Materials, Tools, and Consumables.....	11
2.1 Tools and Consumables	12
2.2 Andrew’s Toolkit	13
2.2 Illustrated Parts List	16
3 3D-Printing	26
4 Preparing the Housing.....	29
5 Arduino Preparation and Software.....	31
5.1 (Carrier Board) Preparing the M0 Adalogger, Real-Time Clock, and carrier board.....	32
5.2 (Protoboard) Preparing the Adalogger and Featherwing RTC	37
5.3 Software and Libraries.....	39
5.4 Required Arduino Libraries	39
5.5 Flash the OpenCTD software to the Adalogger.....	40
5.6 Benchmark: Confirm Software Installation and Adalogger/RTC assembly.....	42
6 Sensor Assembly and Breadboard Testing	43
6.1 Sparkfun Pressure Sensor	43
6.2 DIY Pressure Sensor	44
6.3 Temperature Sensors.....	47
6.4 Conductivity Sensor.....	48
6.5 Benchmark: Breadboarding the Circuit (Protoboard).....	50
Breadboard Diagram.....	51
7 (Protoboard) Assembling the Protoboard Control Unit.....	53
7.1 Preparation	53
7.2 Powering the Protoboard.....	53
7.3 Populating the Perma-Protoboard.....	55
7.4 Benchmark: Testing the Control Unit (Protoboard)	59
8 Building the Sensor Package and Potting Sensors.....	60
8.1 Baseplate and Potting the Pressure Sensor	60

8.2 Attaching JST connector.....	62
8.3 Creating ad-hoc connector	64
8.4 Potting the OpenCTD	66
9 Powering the OpenCTD.....	68
9.1 Internal Power Switch.....	68
9.2 External Magnetic Switch.....	69
10 Casting and Ballast	72
10.1 Casting Loop.....	72
10.2 Weighted Casting Line	73
10.3 Internal Ballast	74
10.4 External Ballast.....	75
11 Calibration and Data Management	76
11.1 Pressure (Depth).....	76
11.2 Temperature	77
11.3 Conductivity (Salinity).....	79
11.4 Data Management	81
11.5 Stewardship.....	82
12 The First Deployment	83
13 Works Cited	85
14 Resources and Datasheets	86
15 Bill of Materials	87
16 Consumables	90

1 Overview

There is something radical about making and using your own scientific instruments. The structure of scientific inquiry has coalesced around a model that is, in general, both expensive and exclusive. This centralizes knowledge production within a circle of individuals, organizations, and institutions which rarely reflects the breadth of identities, experiences, and ways of knowing of those most directly connected to the places being explored.

By building your own instruments to study and understand the natural world, by taking ownership over the tools of scientific inquiry, you contribute to expanding that circle of knowledge production. A scientist who can make their own instruments is not beholden to the cycles of funding and access that pervade formal, institutional inquiry. A researcher who can build and repair their own equipment is not dependent on the ever-changing wind of academic sentiment to decide what is and is not worthy of study. A community leader who has the tools to create their own data does not have to wait for institutions to take notice of an emerging crisis before taking action.

You don't need to ask for permission to understand your world.

Nowhere is this inequality of access more pronounced than in the ocean sciences, where all but a few entities have the capital to mount major oceanographic research campaigns. Even localized coastal research can be stymied by the need for vessels, equipment, and instruments, access to which is often controlled by research institutions. As the need to understand the dramatic changes happening both at the surface and beneath the waves accelerates, barriers to access that precludes the participation of ocean stakeholders further erode our potential to understand, anticipate, and mitigate those changes.

The ocean belongs to everyone. The tools to study the ocean should be accessible to anyone with the curiosity and motivation to pursue that inquiry.

Chief among these tools is the workhorse of oceanography, the CTD.

The ocean is not uniform, it is filled with swirling eddies, temperature boundaries, layers of high and low salinity, changing densities, and other physical characteristics invisible to surface observers (Sverdrup et al., 1942). By measuring salinity, temperature, and depth, scientists can begin to unlock ocean patterns hidden beneath the sea's surface. To reveal these patterns, oceanographers use a CTD—an instrument that measures **C**onductivity (used to calculate salinity), **T**emperature, and absolute pressure (used to determine water **D**epth; Thomson and Emery, 2014).

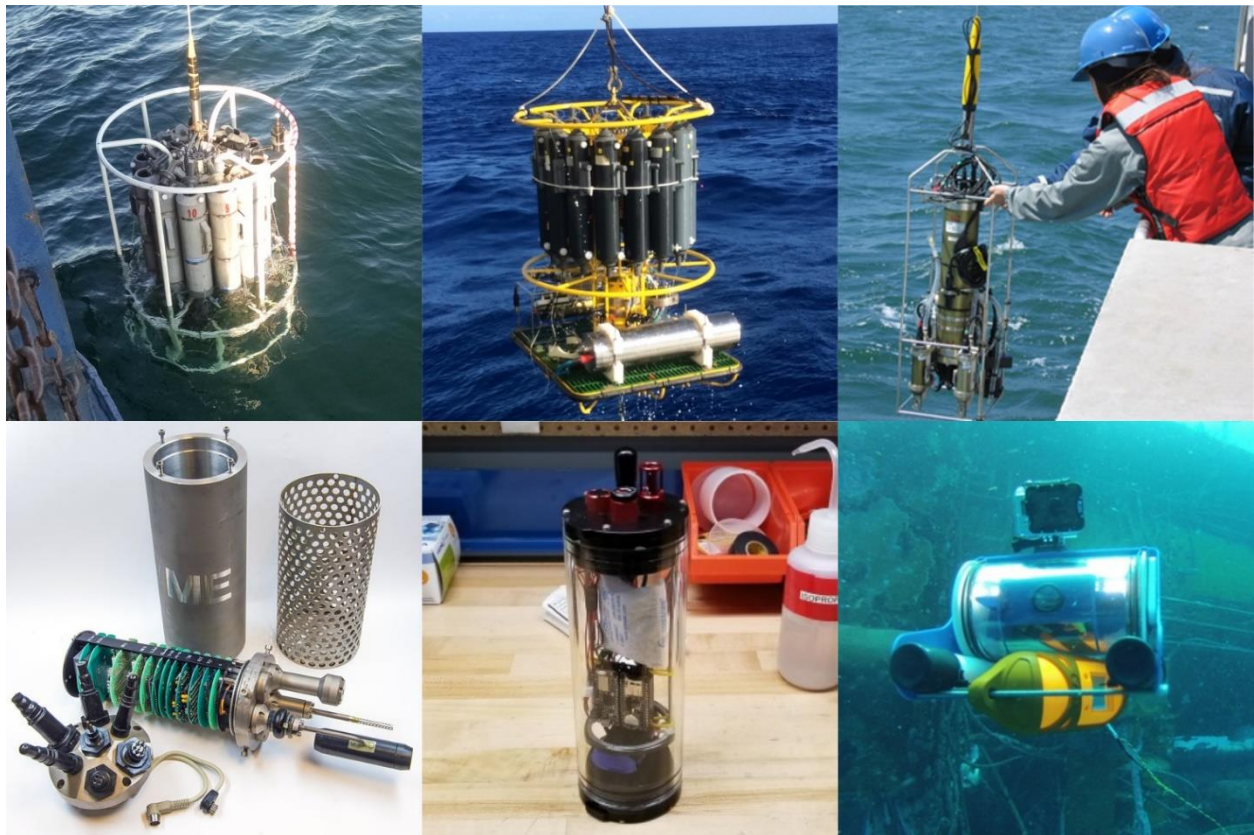


Figure 1. A selection of various CTDs used for field work. Top left: A CTD and rosette deployed from a NOAA boat (WHOI); top center: A CTD and rosette deployed by IFREMER; top right: a student deploys a CTD (MIT); bottom left: an 80s era ME GRISARD CTD; bottom center: the CTDizzle open-source CTD by Ian Black; bottom right: a YSI Castaway affixed to an OpenROV 2.3 (Sofar Ocean).

CTDs come in a variety of shapes, sizes, and applications. Most oceanographic research vessels have a CTD connected to a rosette platform, which houses other instruments and collects water samples in parallel with real-time data. CTDs are also commonly attached to fixed moorings, autonomous underwater vehicles (AUVs), remote-operated vehicles (ROVs), and, on occasion, marine animals (Hooker and Boyd, 2003).

Commercial CTDs are expensive, with the even most affordable models costing several thousand dollars. For near-shore oceanographic research on the continental shelf, where the water depth rarely exceeds 200 meters, this cost can be prohibitive. The expense of the CTD is a barrier-to-entry for formal and informal researchers working with limited budgets, including scientists from low- and middle-income countries and small island states, as well as citizen oceanographers, environmental educators, conservation and management practitioners, and students of all levels interested in understanding their local coasts and waterways.

Climate change, ocean acidification, and sea level rise have created an urgent need for extensive measurements of oceanographic conditions both globally and locally (Stammer et al., 2016; Stephen et al., 2016). At the same time, government funding at both state and federal levels for climate change research can be unreliable, if not wholly non-existent. This creates a pressing

need for low-cost alternatives to common scientific instruments that allow formal researchers to stretch the impact of extant funding while simultaneously enabling informal researchers (such as environmental monitoring organizations and non-governmental organizations, as well as concerned community members) to contribute water-quality measurements.

The OpenCTD is a low-cost, open-source CTD designed intentionally for budget-restricted scientists, educators, and researchers working in nearshore coastal ecosystems, where entire research projects can be conducted for less than the cost of a commercial CTD. It was developed by a core team of marine ecologists in collaboration with a distributed community of scientists, engineers, makers, and conservation practitioners. It is assembled from components commonly available at large hardware stores or through major online retailers. An Arduino-based microcontroller controls a battery of sensors sealed within a PVC pipe. Power is provided by a standard 3.7V lithium polymer battery and data are stored in a tab-delimited text file accessed via SD card. All OpenCTD software is released open source with no restrictions on use.

The OpenCTD is designed to be built by the end-user, providing both access to the tools of oceanography as well as the skills to maintain, repair, and replace OpenCTDs. For scientists working in remote settings, where commercial repair solutions may be non-existent, the capacity to repair the OpenCTD in-house provides an extra buffer against equipment failure. For educators looking for novel, in-depth, hands-on STEM experience for advanced students, the process of building an OpenCTD offers an introduction to coding, 3D-printing, hardware prototyping, and electronics. Construction of an OpenCTD can provide a practical foundation for courses in oceanography and marine or environmental science.

1.1 Building Skills

Constructing an OpenCTD provides a great foundation for developing hardware and software skills. With a few hours of practice, you can learn the soldering, electronics, and coding skills necessary to build and operate an OpenCTD.

For absolute beginners, we recommend working through **Soldering is Easy: Here's How To Do It** by Mitch Altman, Andie Nordgren, and Jeff Keyzer, which is an accessible and approachable introduction to soldering fundamentals:

- **Soldering is Easy: Here's How To Do It:** mightyohm.com/files/soldercomic/FullSolderComic_EN.pdf

If you would prefer to practice on something simpler to get started, the MightyOhm Geiger Counter Kit, the Mega Menorah 9000 Soldering Kit, and the Gikfun MDS-60 Metal Detector Module for Arduino DIY Soldering Practice Kit are great introductory kits to teach soldering and basic electronic principles to students of all ages. Preparing the components needed for an OpenCTD will provide most students with sufficient grounding to tackle the more complex elements of constructing an OpenCTD.

- **Geiger Counter Kit - Radiation Sensor:** www.adafruit.com/product/483
- **Evil Mad Scientist Labs Mega Menorah 9000:** www.adafruit.com/product/2291
- **Gikfun Metal Detector Soldering Practice Kit:** <https://amzn.to/2NIBdX8>

Arduino is a coding environment built on C/C++ and designed for flexibility and ease of use, but it can still seem daunting to first-time users. Practically, the OpenCTD source code is complete and most users will only need to make small alterations to 3 lines of code during assembly and testing, but it is helpful to understand what the software is doing. Calibration and troubleshooting will require users to load different programs onto the completed OpenCTD and if you want to add new sensors, change the sampling rate, or simply alter the format for the data file, you'll need to be comfortable making modifications to the source code.

Basic familiarity with the Arduino Integrated Development Environment (IDE) is essential. **Environmental Monitoring with Arduino: Building Simple Devices to Collect Data About the World Around Us** is a short, easy to understand, though dated, introduction to the Arduino ecosystem, and will provide you with many ideas for expanding your toolbox of open-source environmental sensors.

- **Environmental Monitoring with Arduino:** <https://amzn.to/32myCR3>

You will also want to familiarize yourself with resistors and how to read the color code used to demarcate resistance. Arrow provides a good overview of the concept:

- **Cracking the Resistor Color Code:** <https://www.arrow.com/en/research-and-events/articles/resistor-color-code>

1.2 Project Status and Compatibility

OpenCTD is stable (as of 3/3/2021) on the ATSAMD21G18 ARM Cortex M0 processor. Construction workflow, as well as pinouts, bill of materials, and operation are designed around the Adafruit Adalogger M0. Adjustments may need to be made to adapt to other M0-based microcontrollers. There are currently two parallel OpenCTDs currently in development based on different sensor platforms.

An older version of OpenCTD is stable on the ATMEGA32U4 processor common to many Arduino systems. Construction workflow, as well as pinouts, bill of materials, and operation are designed around the Qduino Mini. Adjustments may need to be made to adapt to other ATMEGA32U4-based microcontrollers. This version is no longer supported.

1.3. Acknowledgements.

From its inception, the OpenCTD and Oceanography for Everyone has benefitted from the support and expertise of an incredible community. Funding to support the OpenCTD was provided by the Bureau of Ocean Energy Management as well as through an initial crowdfunding campaign and an ongoing subscription patronage program through Patreon. Dr. Kim Martini provided exceptional guidance throughout the development process. Dr. Harold Tay provide extensive assistance in designing the OpenCTD control board, as well as general guidance during the entire development process. Kyle Worcester-Moore provided significant assistance streamlining the core OpenCTD code.

We would also like to thank David Lang, Eric Stackpole, Walt Holm, Zack Johnson, and Brian Grau of OpenROV, as well as Dr. Miriam Goldstein, Dr. Amy Freitag, Dr. Craig McClain, Murt Conover, Jake Levenson, Dr. Alex Deghan, Dr. Barbara Martinez, Andrew Quitmeyer, and the participants of GOSH Santiago for their support, advice, and guidance, and several anonymous GitHub contributors who provided significant assistance with early code development.

Understanding the different OpenCTD models:

There are currently 2 different models of OpenCTD under active development. The standard OpenCTD and the OpenCTDX, a slightly more advanced model with deeper operating parameters and tighter sensor tolerances.

This manual is primarily concerned with the construction of the standard OpenCTD, but throughout the guide we will discuss upgrades and alternatives to the standard construction pathway. There are two potential build paths for the OpenCTD: one in which the control unit is constructed using a solderable breadboard (the protoboard) and one which uses a prefabricated custom carrier board that dramatically simplifies the construction process. The recommend pathway is to use the custom carrier board.

Where the build process differs, alternate assembly pathways using the Adafruit Perma-protoboard will be highlighted in a yellow box like this one.

Understanding the different pressure sensor options:

There are a variety of pressure sensors available for OpenCTD builders, including the recommended Sparkfun 14bar Breakout Board, a much more complex, but less expensive do-it-yourself alternative, and a 30bar IMU Breakout Board originally developed for the OpenROV but repurposed for deeper operating OpenCTDs.

Where the pressure sensor build process differs from the recommended Sparkfun Breakout Board, alternate assembly pathways will be highlighted in a green box like this one.

2 Materials, Tools, and Consumables

Our approach to designing the OpenCTD focuses on accessibility—both accessibility of the components and the relative ease of construction. Our goal is that any user should be able to source the parts and tools necessary to build their own OpenCTD without needing to rely on custom manufacturing or highly volatile small-batch producers. We try to keep the number of non-standard parts to a minimum and, when unavoidable, use manufacturers with a long history of consistent quality who will continue to manufacture and support these components into the foreseeable future.

While we now provide a custom printed circuit board that dramatically improves the construction experience, we also provide an alternative process using standard prototyping boards.

The OpenCTD can be assembled entirely from components available from large online electronics vendors such as Adafruit, Sparkfun, Amazon, or AliExpress as well as local hardware stores. Many components can also be purchased directly through their supplier or sourced through auction sites at lower cost. Quality across suppliers for most components has been generally consistent.

Some parts, such as resistors and capacitors, are cheap but hard to source individually. “Introduction to electronics”-type kits, which include many standard resistors and capacitors in small batches tend to be the most economical choice.

Raw, calibrated absolute pressure sensors are available in both 14-Bar and 30-Bar modules from several vendors. These are surface mounted sensors that require significant soldering skills to mount properly. Beginners may prefer to use the simpler, but more expensive, pressure sensor breakout board.

While we provide a list of recommended parts and their suppliers, these are not necessarily the least expensive option and prices can vary significantly depending on local availability and number of units being built. Part availability varies from month to month, and some components may become permanently unavailable or listed under a new manufacturer and product name. We do our best to keep the bill of materials up to date. If you find any discrepancies, please contact us.

When available, we provide Amazon Affiliate links to purchase components. These provide us a small kickback when you buy a component through those links, which helps offset the cost of development and testing for the OpenCTD. We are, however, not endorsing Amazon and encourage users to seek out local suppliers and small businesses whenever possible.

2.1 Tools and Consumables

We have tried to keep the tool list to a minimum. A basic soldering kit will be more than sufficient for assembly. Several components do require a 3D printer.

Tools for general electronics work: flush cutters, wire strippers, soldering iron, solder tip cleaner, helping hands, needle-nosed pliers, utility knife, solderless breadboard, heat gun.

Consumables for general electronics work: lead-free solder, electrical tape, 22-gauge solid core wire (only if building the protoboard version), 20-gauge copper hook-up wire, heat shrink tubing for small gauge wire.

Tools for 3D printing: 3D printer, deburring tool or utility knife.

Consumables for 3D printing: PLA Filament.

Tools for housing and adhesion: Hand saw or PVC pipe cutter, drill with 1” bit or router and jig, 3M epoxy gun for 50ml cartridges.

Consumables for housing and adhesion: LOCTITE 2-part Adhesive Epoxy (Hysol EA-90FL), venturi mixing nozzles for 50 mL epoxy cartridges, LOCTITE Epoxy Five Minute Instant Mix, sandpaper.

Tools for Arduino programming: Computer capable of running Arduino IDE, micro-USB cable.

Consumables for Arduino programming: Calibration standard for conductivity.

A note on Epoxies and Glues:

We have tested several different epoxies and glues to ensure a generally wide variety of options for sourcing. For smaller assemblies, including the sensor connector and for potting the pressure sensor, we use a standard 5-minute marine epoxy.

Hysol EA-90FL has been extensively tested in OpenCTDs as well as other marine hardware, however it is not always easy to source. Other two-part epoxies can also be used for building the sensor pass-through, provided they have a minimum sheer strength of 900 PSI and a medium viscosity rating. Tests with Hysol E-60HP resulted in consistent failures and E-60HP as well as epoxies with similar characteristics, particularly those with higher viscosity, should be avoided.

Any 5-minute marine epoxy is acceptable for sealing the pressure sensor.

2.2 Andrew's Toolkit

Sometimes it is nice to have a list of tried and tested tools rather than having to source everything yourself.

The following are the tools in Andrew Thaler's personal OpenCTD toolkit and his preferred consumables. Though these may not represent the best tools for your needs, this kit has been used to reliably develop and construct dozens of OpenCTDs. Since Andrew runs a fabrication studio involved in multiple projects, in several cases these tools may be excessive.

Soldering Irons (Yes, he has three irons, each of which is useful in different circumstances. Any of them can be used to complete the entire assembly, though the FX-901 is the most difficult with which to work.):

- **For benchtop work with reliable power:** Hakko FX888D-23BY Digital Soldering Station (<https://amzn.to/3vCWNuj>)
- **For field work with limited power access:** Hakko FX-901/P Cordless Soldering Iron (<https://amzn.to/30Uz62r>)
- **For workshops, small spaces, or where USB-C power supplies are preferable:** TS80P 9V/12V Type C USB Programmable Pocket-size Smart Mini Portable Soldering Iron Station (<https://amzn.to/3lunTPF>)
- **Solder Tip Cleaner:** Hakko 599B-02 Wire-type soldering iron tip cleaner (<https://amzn.to/38RF08S>)
- **Helping Hands:** KOTTO Helping Hands Soldering Stand (<https://amzn.to/38PSeD6>)
- **Solderless Breadboard:** Mini Solderless Breadboard (<https://amzn.to/3bYAFmp>)

Hand Tools (for both electronics and housing assembly):

- **Flush cut pliers:** 170-II Micro-Shear Flush Cutter (<https://amzn.to/3bYQPMp>)
- **Wire strippers:** Stanley Hand Tools 84-213 Wire Stripper (<https://amzn.to/30URW9D>)
- **Needle-nosed pliers:** Stanley 84-096 5-Inch Needle Nose Plier (<https://amzn.to/3eS8cAD>)
- **Utility knife:** OLFA 1075449 XA-1 9mm Fiberglass Rubber Grip Utility Knife (<https://amzn.to/3cOixpw>)
- **Hand saw:** CRAFTSMAN Hand Saw, 12-Inch Hacksaw (<https://amzn.to/2OIVGIM>)
- **Epoxy gun:** 3M 08190 Performance Manual Applicator for 50 ml Cartridges (<https://amzn.to/3twpOWG>) *Note: when Andrew bought this applicator, it was still under patent and the only version available. It's worked reliably for 10+ years, but there are now several much less expensive clones on the market that all look perfectly fine.*
- **Glue gun:** Westcott Premium Safety Mini Hot Glue Pen (<https://amzn.to/3txfptR>)

Power Tools (these are all overpowered for the job, but are common in most machine shops, maker spaces, and maintenance facilities and will make housing assembly faster and more consistent):

- **Drill:** DEWALT 20V MAX Cordless Drill / Driver (<https://amzn.to/2Qajoy0>)
- **1-inch spade bit:** Irwin Tools 1764345 Short Spade Bit 1" (<https://amzn.to/3tuSdwn>)
- **Router:** PORTER-CABLE 690Lr Router 1-3/4Hp (<https://amzn.to/3eNjAO6>)
- **Chop saw:** The sliding compound miter saw Andrew uses is no longer in production, but pretty much any standard chop/miter saw will do and, in a pinch, a handsaw works just fine.

3D Printing (Pretty much any 3D printer will work, however, Andrew conducted a review of low-cost printers to determine the best 3D printer for field work, using the OpenCTD parts as one of the test standards. You may find the results of that project helpful in determining the best printer for you: <http://www.southernfriedscience.com/the-quest-for-the-best-tough-3d-printer-for-under-200-our-final-recommendations/>):

- **3D Printer:** Creality Ender-3 3D Printer (<https://amzn.to/2QdtIFr>)
- **Deburring tool:** General Tools 484 Swivel Head Deburring Tool Kit (<https://amzn.to/3eThJY8>)

Programming and Calibration:

- **Computer:** Any modern PC, Mac, or Chromebook can run the Arduino IDE. For users who may not be able to install software on their computer, Arduino also provides a web-based IDE.
- **Micro-USB cable:** Any standard microUSB cable works fine.
- **Calibration standard:** Atlas Scientific Conductivity Calibration K 1.0 Set (<https://atlas-scientific.com/calibration-solutions/conductivity-calibration-k-1-0-set/>)

Consumables:

- **PLA Filament:** HATCHBOX PLA 3D Printer Filament (<https://amzn.to/3tDF1oQ>)
- **Lead-free solder:** Mudder Lead Free Solder Wire (<https://amzn.to/38WomAl>)
- **Electrical tape:** 3M Scotch #35 Electrical Tape (<https://amzn.to/2QeA9rS>)
- **22-gauge solid core wire:** Solid Hook Up Wire Kit (<https://amzn.to/2Nx58hR>)
- **20-gauge copper hook-up wire:** 20 AWG Flexible 1007 Wire Electric Wire (<https://amzn.to/2OT17HH>)
- **Heat shrink tubing:** NTE Heat Shrink 2:1 (<https://amzn.to/3IEIUY4>)
- **Sandpaper:** 3M Garnet Sandpaper, 9-Inch by 11-Inch, Assorted-Grit (<https://amzn.to/3vCfaQ9>)
- **Hot glue:** Gorilla 3034518 Hot Glue Sticks (<https://amzn.to/3vEGtt0>)

- **2-part adhesive epoxy:** Loctite 219298 Gray E-90FL Hysol Epoxy (<https://amzn.to/315kQ6V>)
- **Venturi mixing nozzles for 50 mL epoxy cartridges:** 3M Static Mixing Nozzle, 38193, 50 mL (<https://amzn.to/30YJB4N>)
- **Five-minute epoxy:** Loctite Epoxy Five Minute Instant Mix (<https://amzn.to/3tuKM8q>)

2.2 Illustrated Parts List

3D-printed parts



Magnetic Switch Assembly



Sensor Baseplate for Sparkfun/DIY Pressure Sensor



Electronics Chassis for Protoboard Option



Sensor Baseplate for OpenROV IMU



Protective Cap



Sensor protector

Housing



Schedule 40 PVC Pipe



Oatey Plumber's Test Cap



46 to 70mm Hose Clamp



Short Length of Polypro Rope

OpenCTD Rev6 Custom Carrier Board



Custom Carrier Board



Adafruit Adalogger M0



SD Card



Real-time Clock Module



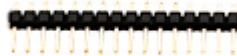
CR1220 Coin Cell Battery



Atlas EZO Conductivity Circuit



Female Headers



Male Headers



7-pin JST Header



MicroUSB Cable



10 kOhm resistor



4.7 kOhm Resistor

Adafruit Adalogger M0 and Real-time Clock Featherwing



Adafruit Adalogger M0



Adafruit Real-time Clock Featherwing



Male Headers



Female Headers



Extra Long Headers



MicroUSB Cable

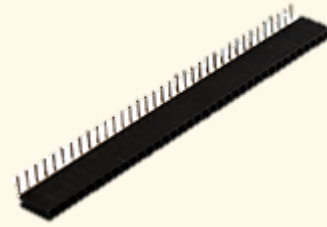
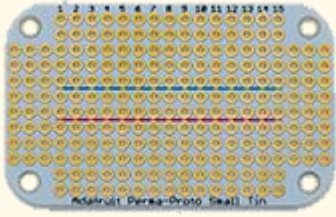


3V CR 1220 Coin Cell Battery



SD Card

Protoboard Control Unit



Atlas EZO Conductivity Circuit

Right angle headers



Solid core wire



10 kOhm resistor



4.7 kOhm Resistor

Sensors



Temperature Sensor



Conductivity Probe



*Sparkfun 14-Bar Pressure Sensor
Breakout Board*



JST Female Connector

Alternative Pressure Sensors



DIP to SOIC Adapter



Pressure Chip on DOIC to SOIC Adapter



10 kOhm Resistor



0.1 uF Capacitor

OpenROV



OpenROV 30bar IMU



Baseplate for 30 bar IMU

Power and Ballast



Battery



Board-mounted Switch



10 mm Magnet for External Switch



Assorted weights for ballasting CTD



String of weights to ballast the OpenCTD



10mm double-sided stickers for external switch

Accessories



Solderless Breadboard



Screw Terminals



Alligator Clip



Jumper wire



SD Card Reader

Please reference the full Bill of Materials in Appendix 1 for further details and sourcing.

3 3D-Printing

The main support structure as well as the sensor baseplate, magnetic switch and connector reinforcement are 3D-printable objects. While it is possible to construct a fully functional OpenCTD without using these parts, they are designed to make building and operating the device simpler and more intuitive. For the standard pathway, only the sensor baseplate is necessary.

- **Sensor Baseplate** which provides a foundation and template for arranging the sensors prior to potting in epoxy. There are several styles available depending on the pressure sensor set-up you're using.
- **3-part Magnetic Switch**, which allows users to turn the OpenCTD on and off without opening the housing. There is a large version and a slim version, both of which can work for any CTD layout. \

- **Electronics Chassis** which holds the Adalogger and other electronic components, allowing users to easily access the SD card while minimizing strain on the wiring. There are two separate versions depending on which control board you are using.
- **Connector**, which allows users to connect sensors to the control board. Only necessary if you are building the DIY version with a Perma-protoboard rather than using the custom control board.

All parts are designed to be printed in PLA at 100-micron resolution with 10 to 20% infill. Only the electronics chassis and magnetic switch (internal) requires support structures. We recommend PLA over other plastics as it is Generally Regarded As Safe, theoretically compostable (but only in a commercial composter), and has not been shown to harm the marine environment. ABS, in contrast, has been shown to cause developmental problems in fish exposed to ABS particulates (Oskui et al., 2016). Other materials, like PETG and Nylon may be preferable but require more 3D-printing experience and a more advanced printer.

Our personal preference is for Polymaker Polyterra PLA, which has been optimized for strength and compostability.

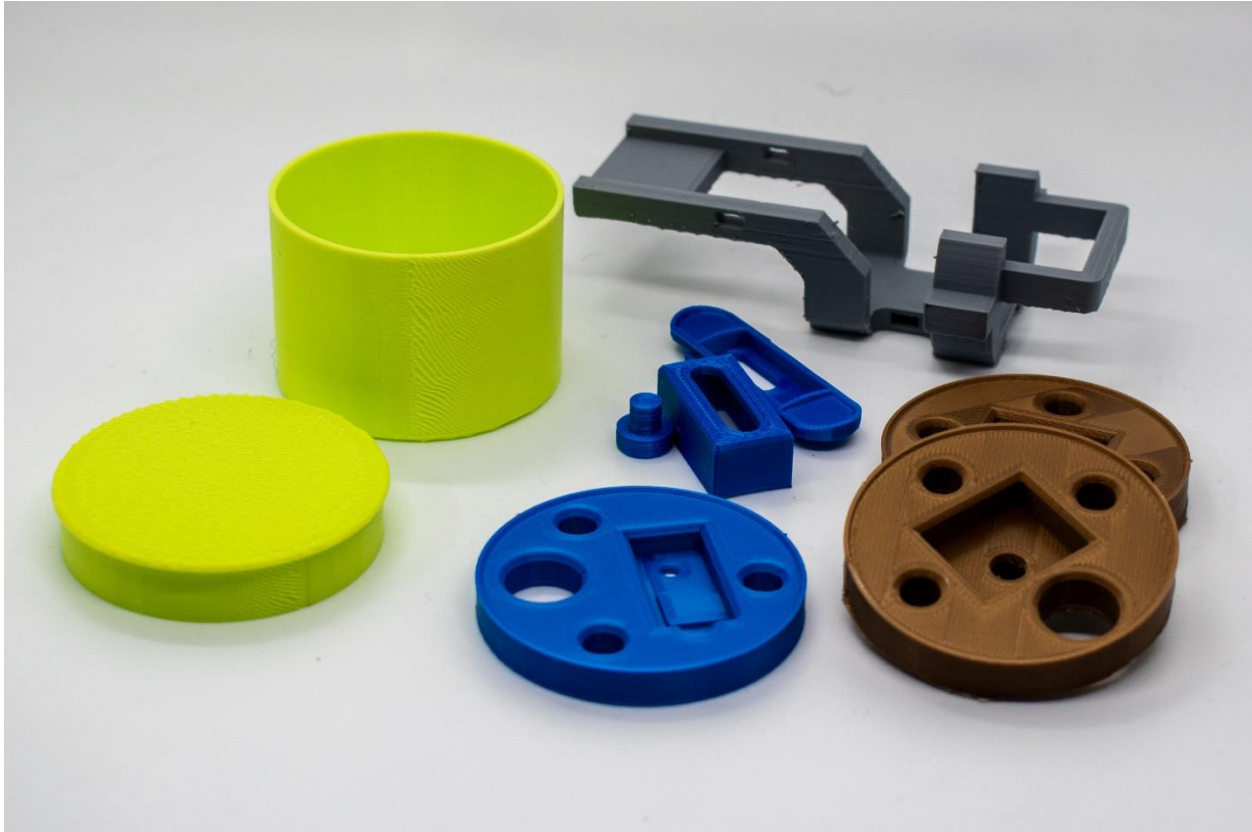


Figure 2: 3D-Printed parts for OpenCTD.

OpenCTD 3D Printed Structural Components:

- OpenCTD Baseplate
 - **Sparkfun Breakout Board:** OpenCTD_M0_Baseplate_SparkfunBreakout.3mf
 - **DIY Pressure Sensor:** OpenCTD_M0_Baseplate.3mf
 - **OpenROV IMU:** OpenCTD_M0_Baseplate_OpenROV_IMU.3mf
- External Magnetic Switch Assembly for all CTDs
 - **Magnetic Switch (external):** OpenCTD_MagneticSwitch_External_M0.stl
 - **Magnetic Switch (internal):** OpenCTD_MagneticSwitch_Internal_M0.stl
 - **Magnetic Switch Pin:** OpenCTD_MagneticSwitch_Pin_M0.stl
- OpenCTD with Protoboard
 - **Electronics Chassis:** OpenCTD_ElectronicsChassis_M0_Reinforced_Switch.stl
 - **Wiring Harness Connector:** OpenCTD_Connector_M0.stl

Once the sensor baseplate is complete, use a deburring tool to round off all the edges on both top and bottom as well as inside each hole. Sensors, especially the pressure sensor, should fit snugly inside their respective hole and the entire baseplate should fit tightly into a 2" PVC pipe.

There are two auxiliary caps to cover the sensors and top opening when painting the CTD. These components are not necessary to complete the build but will simplify some steps.

OpenCTD 3D Printed Auxiliary Components:

- **Cap for sealing the top during painting:** 2-inch_PVC_PaintCap_Top.stl
- **Cap for sealing the bottom and protecting sensors during painting and transport:** 2-inch_PVC_PaintCap_Bottom.stl

4 Preparing the Housing

The OpenCTD housing is comprised of a single piece of schedule-40 2" PVC pipe. Notches or holes cut in the bottom allow water to flow over the sensors while protecting them from impacts and entanglement. This end will be sealed with high strength epoxy, allowing the sensors to pass through the housing while preventing water from entering and damaging the electronics. The open end is sealed with an off-the-shelf pressure test cap designed for plumbers. When cutting and preparing the PVC, take care not to damage the inner surface where the test cap O-ring will seat against the PVC wall. (**Note: while we generally provide measurements in Metric, because US-based plumbing supplies are generally measured in Imperial units, where off-the-shelf parts are used, we will use the measurement standard provided by the manufacturer.**)

1. Cut a 12" section of PVC pipe. Deburr the edges around each cut with a knife, sandpaper, or deburring tool.
2. If you have access to a router table, cut 4 1-inch-deep notches into one end of the PVC using a ½-inch router bit. If you have access to a drill and bit, cut 3 1-inch holes centered ¾-inch from the bottom.
3. Lightly sand the inside bottom of the tube, just above where you cut the slots or holes. This will provide a good surface for the epoxy to adhere to.

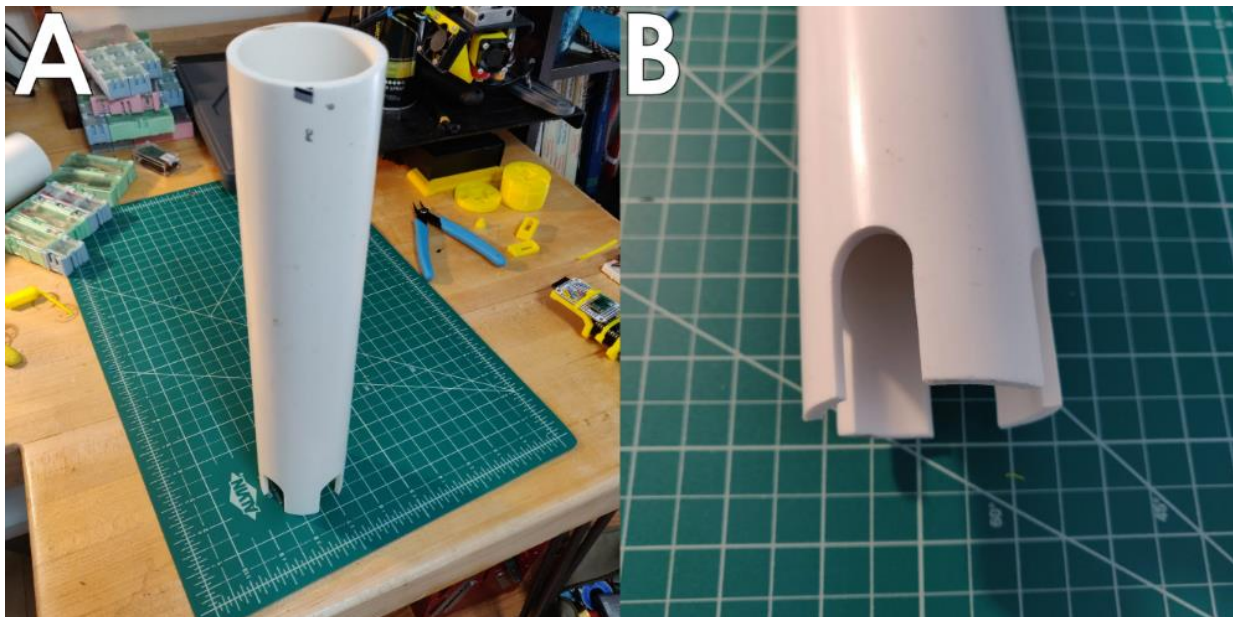


Figure 3. 12" PVC pipe (A) with notches cut at the base (B).

Though seemingly simple, this set-up has several advantages. Underwater housings are notoriously expensive. Using a PVC pipe instead of a conventional housing with sealed

mechanical pass-throughs is the single largest cost-saving step in the entire CTD construction process and allows us to keep costs down without compromising on sensor quality.

For instruments that are deployed in high-traffic areas long-term, there is also a benefit in having an oceanographic instrument that appears “cheap”. It will be far less likely to be interfered with by curious passersby.

If you have decided to use an internal power switch rather than the external magnetic switch, you can paint the housing now.

5 Arduino Preparation and Software

Before beginning the build, familiarize yourself with the electrical components as well as the Arduino Integrated Development Environment (IDE). By ensuring that the software is running and the base components have been assembled and tested, you can save yourself significant frustration down the road. Setting up Arduino from scratch can be a challenging and unintuitive process the first time you do it, so it is a good idea to invest an hour or two into making sure you have compatible, up-to-date libraries and that the OpenCTD code compiles on your computer.

Arduino provides an online development environment if you cannot install software on your personal computer: <https://create.arduino.cc/>.

The Adalogger M0 is a SAMD-based microcontroller with a built in SD card reader. When paired with a Real-Time Clock (RTC), this will allow the user to log timestamped data that can be extracted by any microSD card reader.

The Adalogger and RTC will need headers. Headers are the male or female pin assemblies that allow you to connect and disconnect electronic. You will be soldering components directly to a circuit board. Assembling the Adalogger and Real-Time Clock (RTC) is a good, low-risk, opportunity to practice your soldering skills.

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.

The carrier board will also require you to mount a 7-pin JST connector which will allow solid contact between the control unit and the sensor array. If you are unable to source a 7-pin JST connector, the board has an option to connect standard female headers.

5.1 (Carrier Board) Preparing the M0 Adalogger, Real-Time Clock, and carrier board

The custom carrier board allows for rapid and accurate assembly of the OpenCTD control unit.

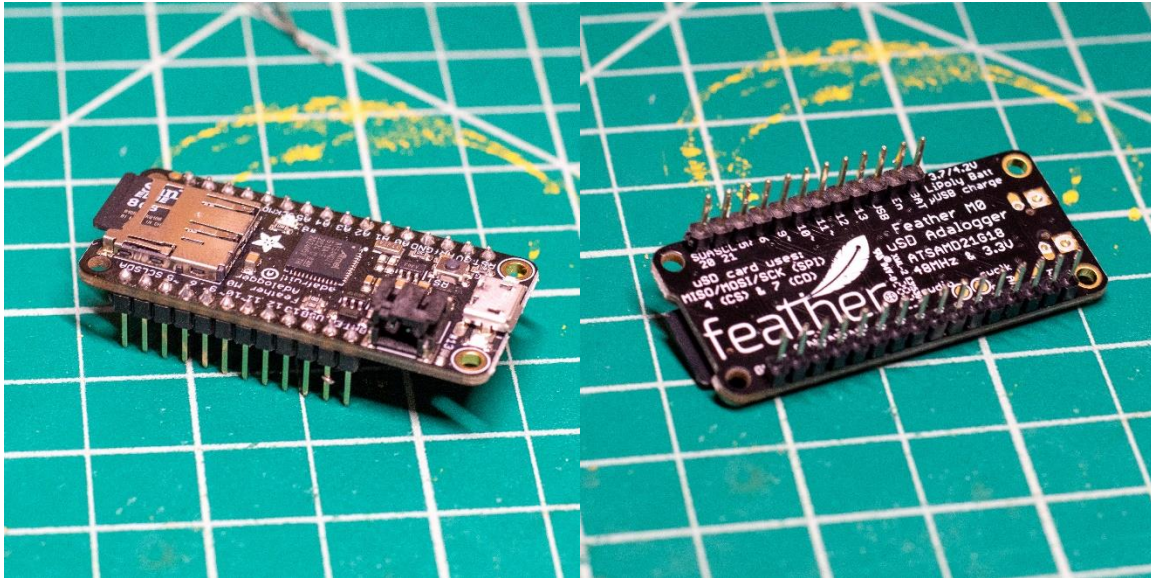


Figure 4. Adalogger M0 with male headers soldered to the pins. Top (left) and bottom (right)

1. Slot the male headers into the through-holes on the underside of the Adalogger M0, ensuring that the number of headers matches the number of available pins. The long header pins should point down. If you are using generic stacking headers, cut them to length with flush cutters so that the number of posts matches the number of pins on each side.
2. Ensure that the headers are square and perpendicular to the face of the Adalogger. On the top of the Adalogger, solder the four corner pins, double check that the headers are straight, and then solder the remaining pins.
3. Slot the standard female headers on the top of the carrier board in the pins for the Adalogger M0. You may need to trim the strip of female headers down to size using flush cutters so that the number of posts match the number of pins in the carrier board.
4. Solder the female headers to the custom carrier board.

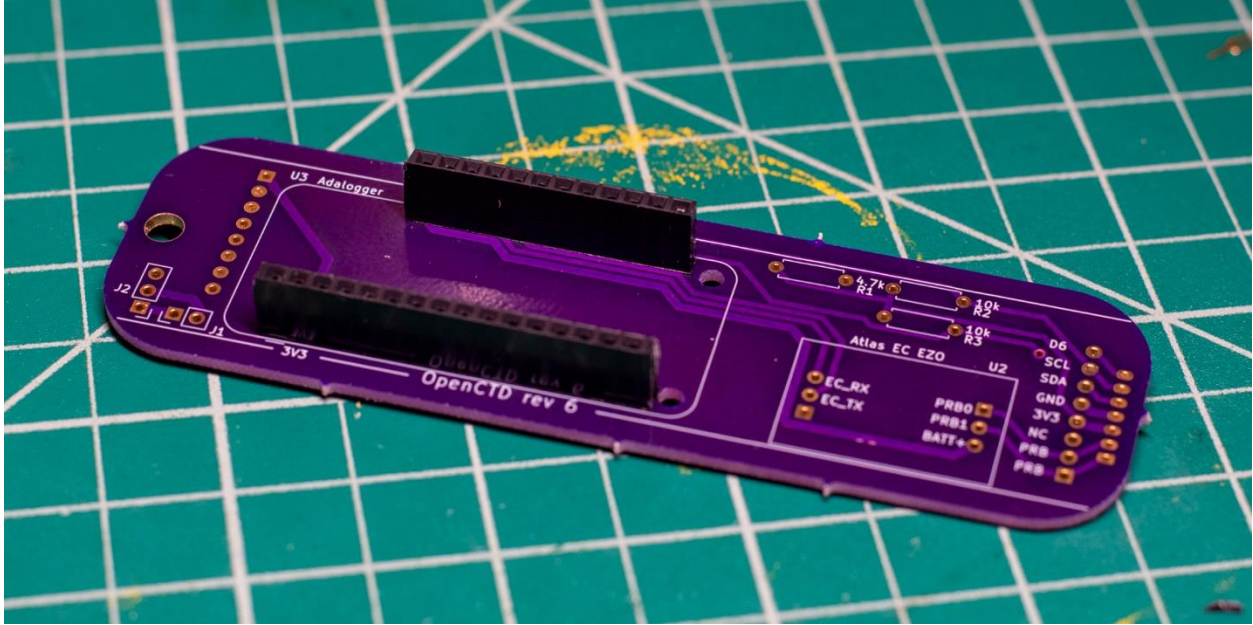


Figure 5. Female headers soldered to the custom carrier board.

5. Slot the male header pins into the RTC module such that the battery is facing up the header pins extend from the opposite side of the module.
6. Solder the RTC to the underside of the custom carrier board by first soldering the short end of the pins to the RTC and then soldering the long end of the pins through the control board. **(Optional: if you would like to keep the modules removable, you can solder female headers to the carrier board and then stack the RTC onto the female headers.)**

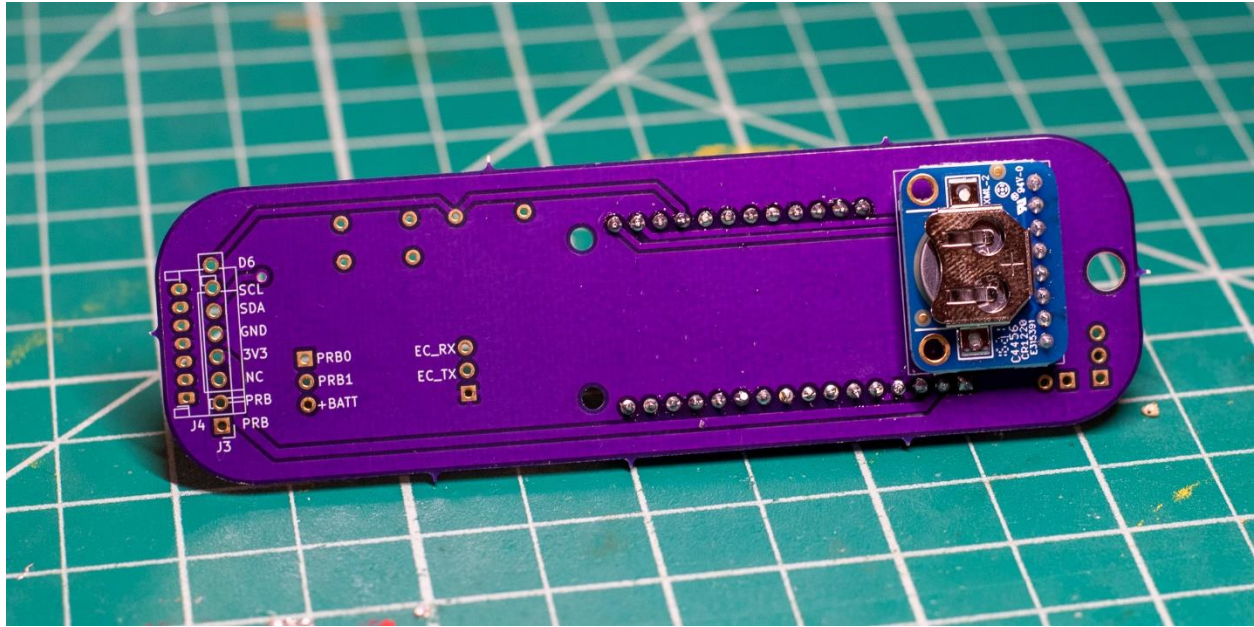


Figure 6. Custom carrier board with RTC module soldered to the underside.

7. Insert the Atlas EZO conductivity circuit into the EZO slots on the topside of the custom carrier board. Ensure that the pin labels on the board correspond to the pin labels on the module.
8. Solder the Atlas EZO conductivity circuit to the topside of the custom carrier board. **(Optional: if you would like to keep the modules removable, you can solder female headers to the carrier board and then stack the EZO circuit onto the female headers.)**
9. Solder the 10kOhm and 4.7 kOhm resistors to their corresponding slots on the topside of the custom carrier board.

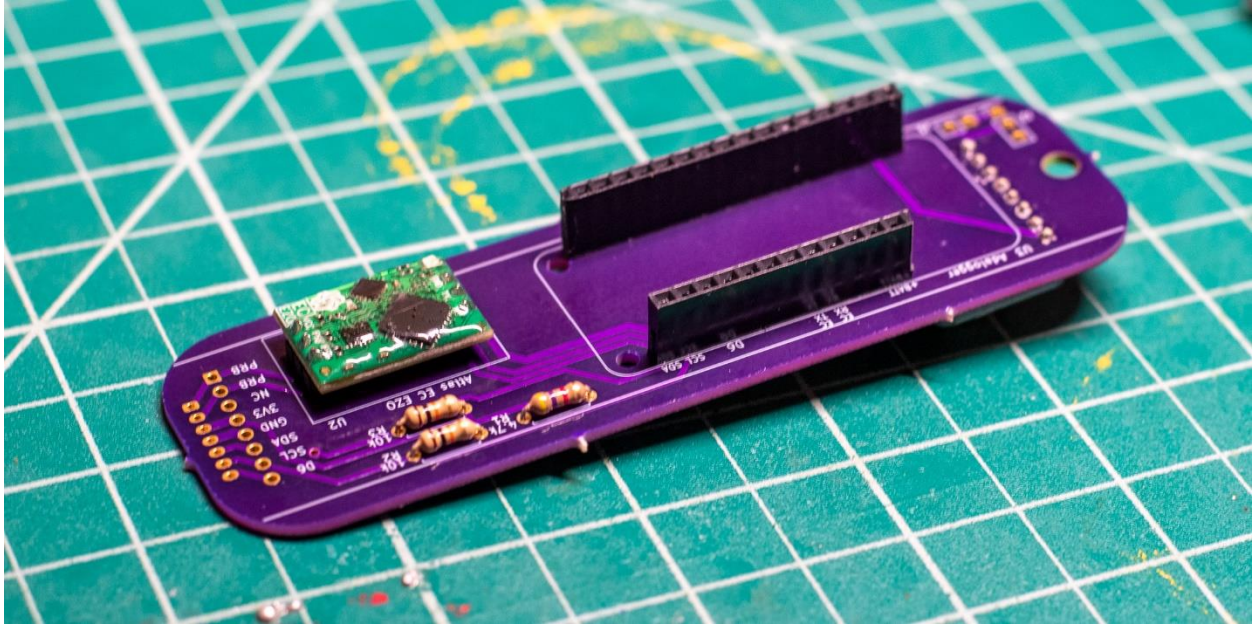


Figure 7. Custom carrier board with Atlas EZO circuit and resistors soldered to the topside.

10. Insert the JST 7-pin connector to the underside of the custom carrier board, ensure that the slot in the plastic connector housing is facing away from the board.
11. Solder the JST connector to the underside of the custom carrier board.
 - **ALTERNATIVE:** if you were unable to source a JST connector, you can solder straight- or right-angle female headers to the 8-pin strip directly adjacent to the JST connector pins. This will allow you to build a simpler, but less reliable connection between the sensors and the control board.

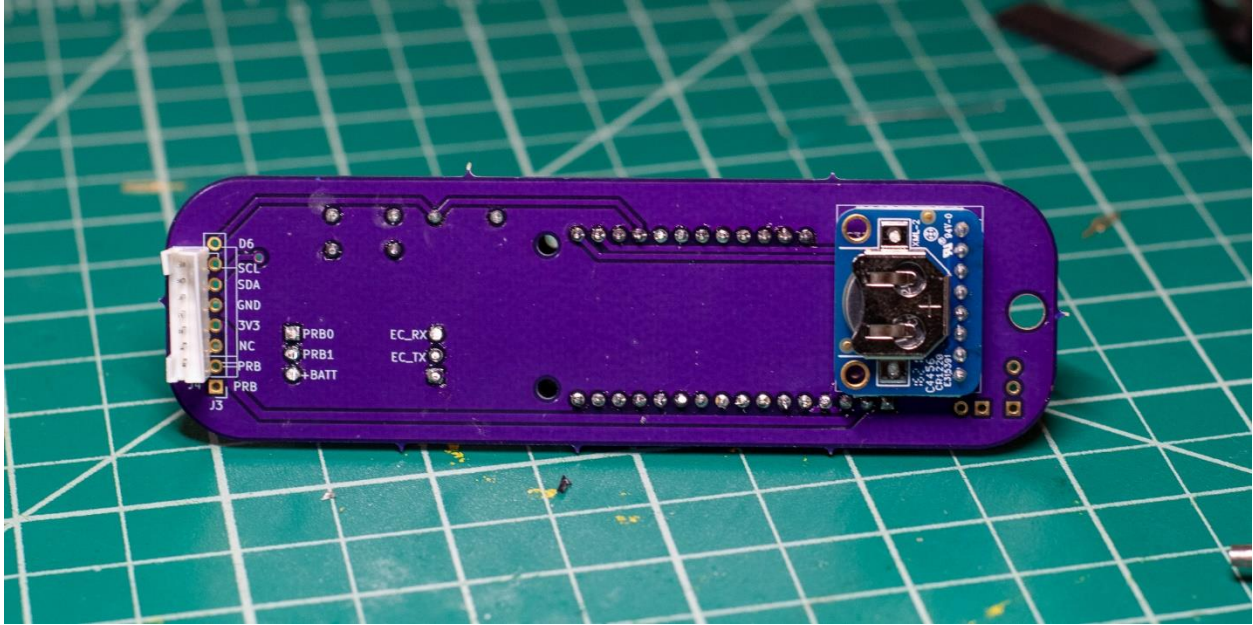


Figure 8. Custom carrier board with JST connector soldered to the underside.

12. Stack the Adalogger M0 on top of the custom carrier board by lining up the male, downward-facing header pins on the Adalogger with the female, upward facing header pins on the carrier board.
13. Insert the CR1220 battery into the RTC.

5.2 (Protoboard) Preparing the Adalogger and Featherwing RTC

The Adalogger will receive stacking headers—headers that contain both male and female ends so that the Adalogger can be connected to a breadboard while a second device (in this case the RTC) can be mounted on top. The RTC will receive male headers facing down so that they can be connected to the Adalogger.

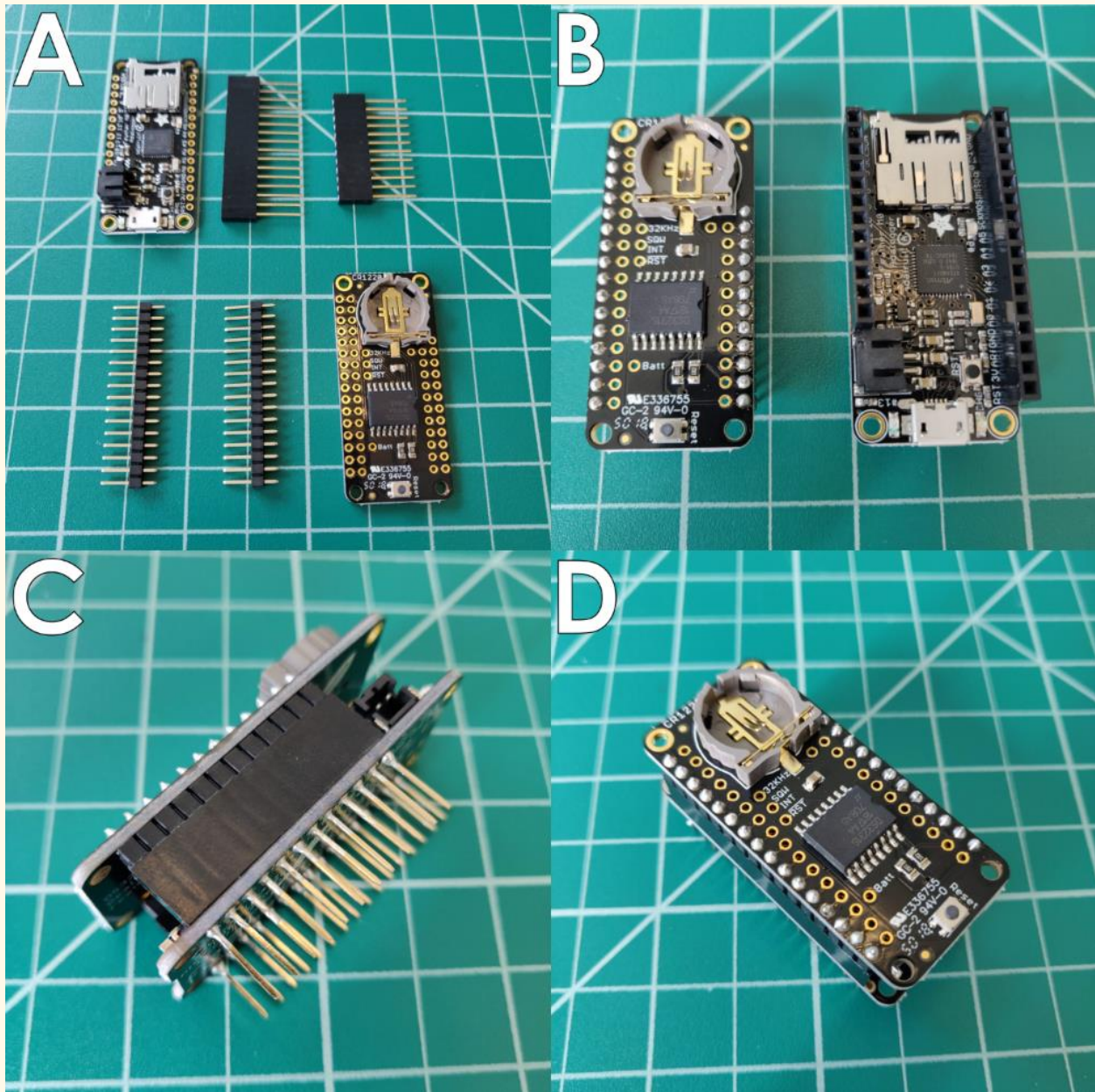


Figure 9. Adalogger and Real-time Clock with associated headers (A); after soldering (B); connected in profile (C); connected from above (D).

1. Slot the stacking headers into the through-holes on the Adalogger, ensuring that the number of headers matches the number of available pins. The female end of the stacking headers should face up. If you are using generic stacking headers, cut them to length with flush cutters so that the number of posts matches the number of pins on each side.
2. Flip the Adalogger over so that male headers are facing up. Ensure that the headers are square and perpendicular to the face of the Adalogger.
3. Solder the four corner pins, double check that the headers are straight, and then solder the remaining pins.
4. Slot the male headers into the outer pins of the RTC such that the short ends pass through the board and the long header pins face down. If you are using generic stacking headers, cut them to length with flush cutters so that the number of posts matches the number of pins on each side.
5. Solder the four corner pins, double check that the male headers are straight and aligned with female headers on the Adalogger, and then solder the remaining pins.
6. Stack the RTC on top of the Adalogger by lining up the male, downward-facing header pins on the RTC with the female, upward facing header pins on the Adalogger.
7. Insert the CR1220 battery into the RTC.

5.3 Software and Libraries

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.

We use GitHub to host various projects. GitHub can seem a bit daunting to people who aren't familiar with it. At its core, GitHub is a tool to organize and coordinate files and maintain version control (multiple people can edit the same file, while keeping everything in sync). Projects are organized into repositories ('repos') that contain the source code, supporting documents, data files, and other materials needed for each project. When you visit a repo, check the README.md, this will generally describe the project (including the current status), contain necessary guides, and let you know how to contribute.

1. Launch Arduino IDE.
2. Under *preferences*, in the *File* menu (on Apples, this is under the *Arduino* menu), add the following URL to the "Additional Boards Manager URLs" field: https://adafruit.github.io/arduino-board-index/package_adafruit_index.json
3. Click "Ok" and then navigate to *Tools > Board > Boards Manager*.
4. In *Boards Manager*, search for "Adafruit SAMD Boards" and install the library. Now search for "Arduino SAMD Boards" and install this library as well.
5. Restart Arduino IDE.
6. Install the required libraries listed below in section 5.3
7. Restart Arduino IDE.

5.4 Required Arduino Libraries

The following Arduino libraries are required for the OpenCTD and come packaged with Arduino IDE by default:

Wire. For communicating with I2C devices like the pressure sensor.

SPI. Serial Peripheral Interface for communicating with a standard SD card reader.
SD. For reading and writing to a standard SD card reader.

You will need to download and install the remaining libraries manually. Please refer to <https://www.arduino.cc/en/Guide/Libraries> for detailed instructions for installing libraries.

MS5803_14. For the MS5803 14-Bar pressure sensor.

Dallas Temperature Control. Allows you to communicate with the DS18B20 thermometer.

OneWire. For controlling the Dallas 1-wire family of devices, which includes the temperature sensors.

RTClib. For controlling and reading the Real-time Clock

SoftwareSerial. A modified version of the standard serial monitoring program that works with SAMD microcontrollers like the Adafruit Feather M0.

These libraries can be downloaded as compressed .zip files from the OpenCTD GitHub Repository:

https://github.com/OceanographyforEveryone/OpenCTD/tree/master/OpenCTD_Feather_Adallogger/Support%20Code

5.5 Flash the OpenCTD software to the Adallogger

We will now install the OpenCTD software on the Adafruit Adallogger M0. Please note that, on newer Adaloggers, Adafruit ships the microcontroller preloaded with both Arduino and Circuit Python, but Circuit Python will be the default. When you plug your Adallogger in for the first time, it will open into the Circuit Python boot drive. You can ignore this. The first time you flash the OpenCTD software, the Adallogger will default to Arduino.

1. Restart Arduino IDE. You should see several new Adafruit options in the *Board* menu. The Adallogger is an Adafruit Feather M0 with an embedded SD card reader.
2. Select “Adafruit Feather M0” from the *Board* menu.
3. Insert the SD card into the Adallogger’s SD card reader.
4. Connect your Adallogger to your laptop using a micro-USB cable. The necessary drivers should automatically install.
5. Open the OpenCTD_master_M0 source code in Arduino IDE.

6. Under *Tools > Port*, select the COM port that reads “Adafruit Feather M0” (if should autodetect, but doesn’t always).
7. Click Upload (either the right pointing arrow on the main interface or under *Sketch > Upload*). The code will compile and then upload to the Adalogger. Note: the RTC sets its time based on the time displayed on the computer when the code is compiled. Should the RTC ever lose time, you can reset it by pulling the battery, power cycling the device, then reinserting the battery and recompiling the OpenCTD source code onto the Adalogger.

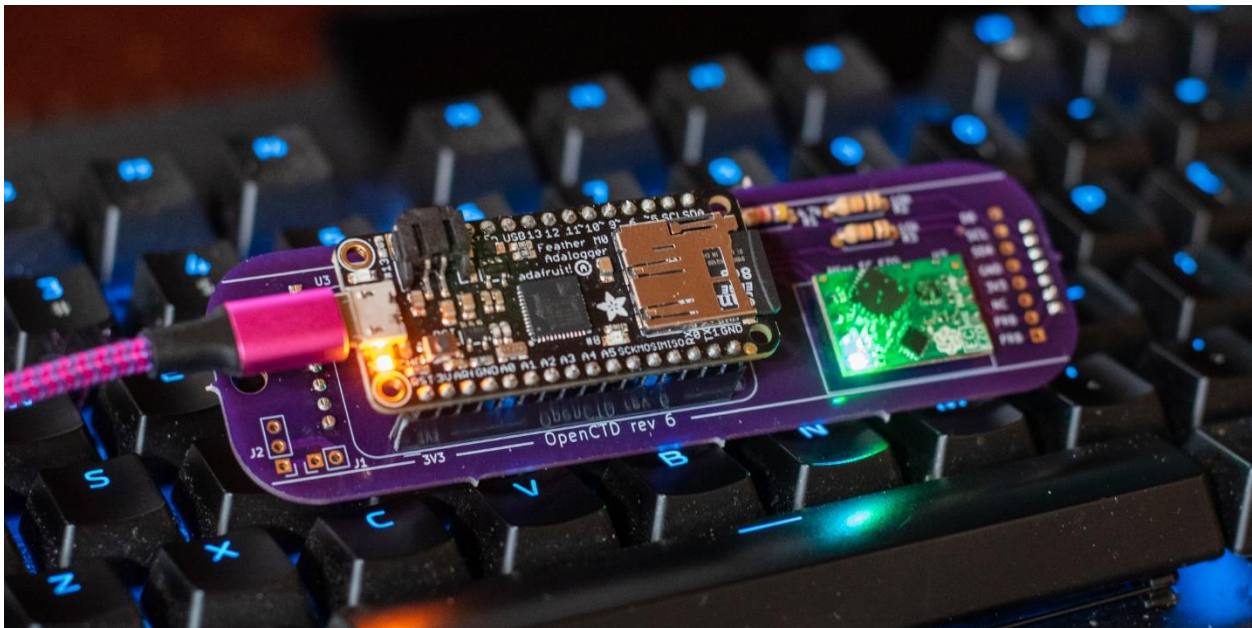


Figure 10. Control unit connected to PC for software validation.

5.6 Benchmark: Confirm Software Installation and Adalogger/RTC assembly

Once the control unit is assembled and the OpenCTD software is flashed to the Adalogger, use the serial monitor in Arduino IDE to verify that everything is working correctly. Open the Serial Monitor by clicking the magnifying glass in the upper righthand corner of Arduino IDE or through *Tools > Serial Monitor*. The software will initialize the SD card reader and RTC and then go through a sensor check, which will fail. You should then see the date and time, followed by a series of zeros or negative values. If you are using the control board, the EZO conductivity circuit should blink blue and green.

You can then unplug the control unit, remove the SD card, and confirm via SD card reader that data was recorded on the SD card.

Checklist:

- Adalogger connects to PC.
- OpenCTD software compiles and uploads to Adalogger.
- Serial monitor connects and can read data.
- Real-time clock shows the correct time.
- Data is recorded on the SD Card.

6 Sensor Assembly and Breadboard Testing

The OpenCTD uses three separate types of sensor to measure the marine environment accurately and precisely. The pressure sensor is a MS5803-14BA built by Measurement Specialties and is accurate to 1cm when properly calibrated. The pressure sensor also includes an on-chip thermistor; however, the onboard thermistor is very slow and is unused. For custom builds with extremely limited space that don't need the device to reach equilibrium quickly, users can opt to read temperature directly from the pressure sensor. The standard OpenCTD uses a battery of three DS18B20 thermistors sealed in a stainless-steel housing to take a temperature average of the ambient environment. Finally, the OpenCTD uses a commercially available Atlas EZO conductivity system consisting of a probe and conductivity circuit to determine salinity.

6.1 Sparkfun Pressure Sensor

If you're using the Sparkfun Pressure Sensor Breakout Board, strip $\frac{1}{2}$ cm off the end of 4 8-cm lengths of 22-gauge stranded wire and solder those wires to the GND, 3.3v (or VCC), SDA, and SCL pins on the back of the board. Keep track of which wire goes to which pin by color-coding each pin (we recommend red for VCC; black for GND; yellow for SCL; and white for SDA). Note that the wires project from the side opposite the pressure sensor.

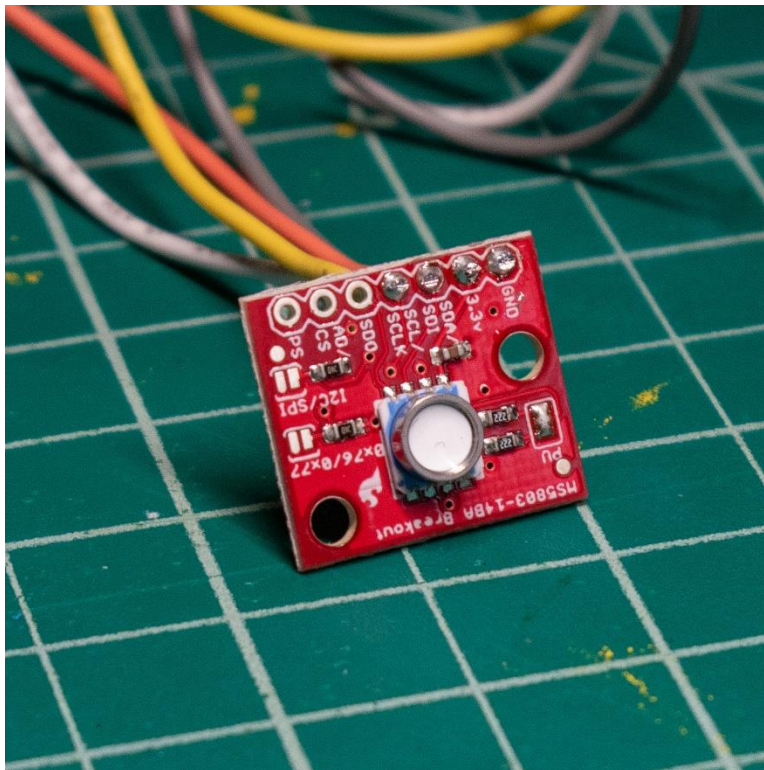


Figure 11. Sparkfun Pressure Sensor Breakout with soldered pins.

6.2 DIY Pressure Sensor

To build the pressure sensor, you will need to surface mount the pressure sensor chip onto an SOIC8-to-DIP adapter. This is the most challenging soldering step in the entire build and requires a steady hand. The pressure chip is able to withstand high heat for a short amount of time, as long as there is no direct contact. Be extremely careful not to touch the gel covering the pressure sensor.

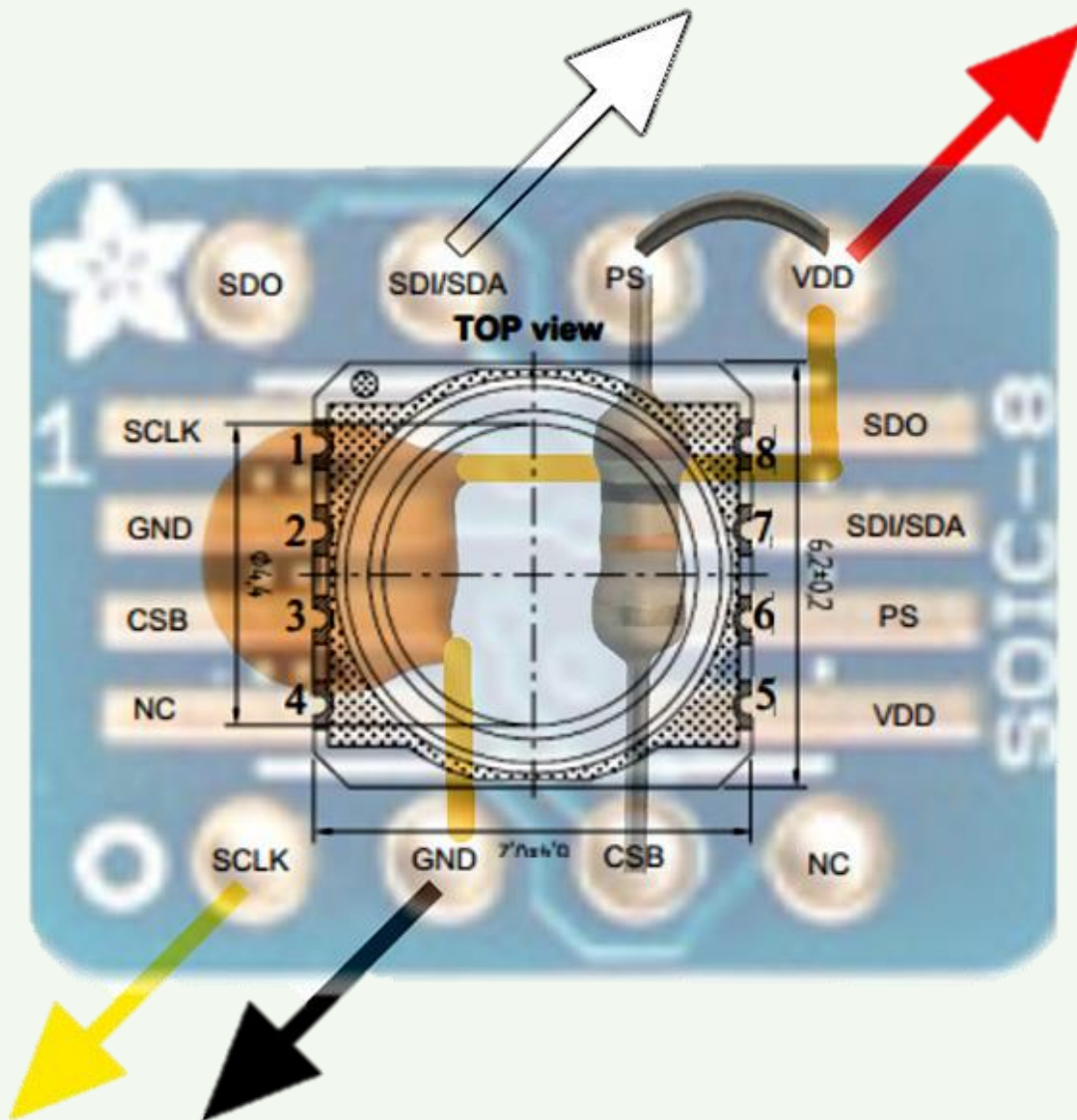


Figure 12. Diagram of pressure sensor chip and component alignment on SOIC-8 to DIP adapter. Note alignment of pin 1 on the chip to the alignment indicator (in this case, the Adafruit star, but on most boards simply a dot). Image courtesy of the Cave Pearl Project, who have an excellent guide to using and soldering these little chips here: <https://thecavepearlproject.org/2014/03/27/adding-a-ms5803-02-high-resolution-pressure-sensor/>.

1. Make sure that side of the SOIC8-to-DIP adapter with the larger pads is facing up. Add a small amount of solder to each pad. If you have solder rosin, add a dab of rosin to each pad to help the solder flow.
2. Ensure that the dot and notch on the pressure sensor line up with the dot on the SOIC-to-DIP adapter. Solder the sensor in place by applying a small amount of solder to the vertical grooves and drawing it down onto the solder pads. Be careful not to short any connections.
3. Bridge the pads for pin 5 (VDD/VCC) and pin 6 (PS) by drawing a small drop of solder between them.
4. Flip over the adapter and solder a 10kOhm pull-up resistor between pin 3 and 6.
5. Insert the 100nF (104) capacitor between pins 2 and 5 but don't solder yet. Bend the capacitor's legs to hold it in place.
6. Solder a 6 cm length of solid core wire to pin 1 (SCL) and pin 7 (SDA). Keep track of which wire goes to which pin by color-coding each pin (we recommend red for VCC; black for ground; yellow for SCL; and blue for SDA).
7. Solder a 6 cm length of solid core wire to pin 2 (GND) and pin 5 (VDD/VCC) such that the legs of the capacitor are captured in the solder joint. You may have to twist the wire a bit to get both the wire and the capacitor leg to fit in a single pin hole. Keep track of which wire goes to which pin by color-coding each pin (we recommend red for VCC; black for ground; yellow for SCL; and blue for SDA).
8. Strip ½ cm of insulation from the exposed ends of each wire.

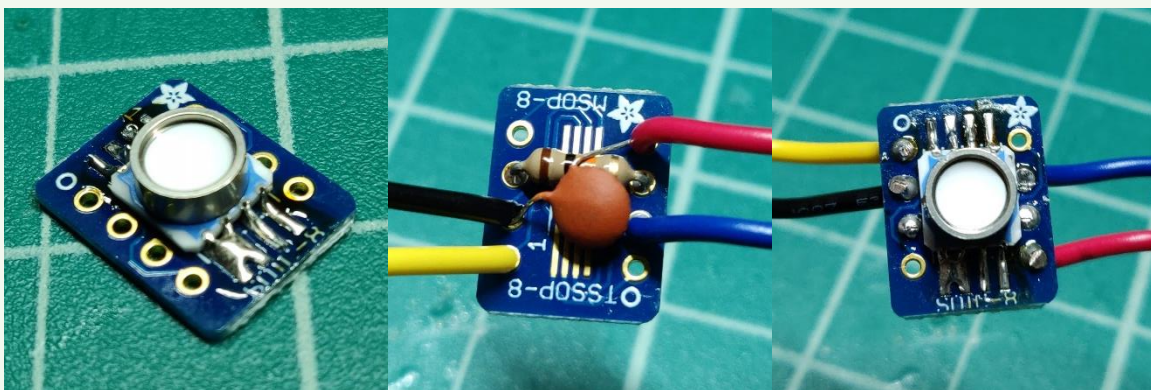


Figure 13. Building the Pressure Sensor unit. A: Components of the sensor. B: Surface mounting the sensor. C: Wiring up the components.

OpenROV IMU If you've received an OpenROV IMU with 30Bar pressure chip, you can use that as an alternative for a deeper operating CTD. Solder a 5 cm length of solid core wire to SCL, GND, VCC, and SDA. Keep track of which wire goes to which pin by color-coding each pin (we recommend red for VCC; black for GND; yellow for SCL; and blue for SDA).

You will also need to 3D print the corresponding baseplate in order to pot the sensor.

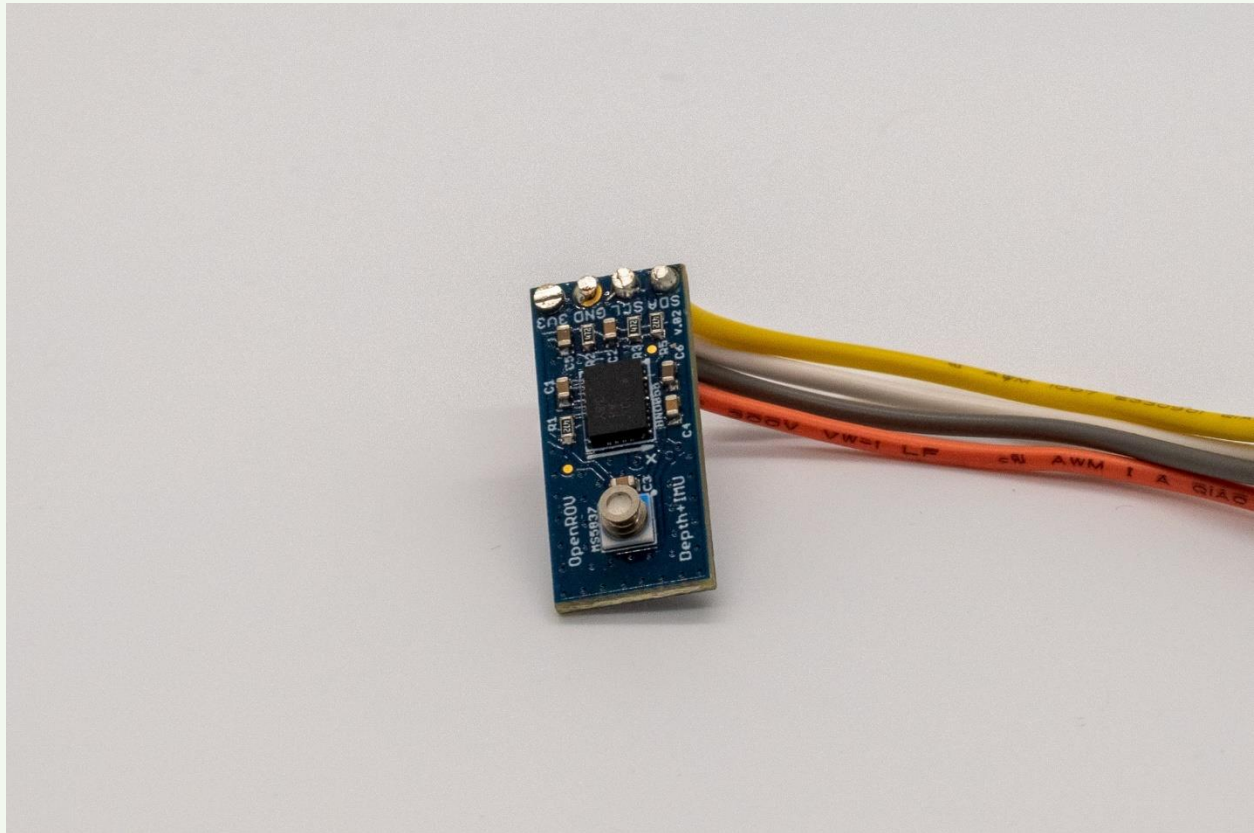


Figure 14. OpenROV IMU with 30-bar pressure sensor.

6.3 Temperature Sensors

The OneWire protocol allows multiple digital sensors to communicate through a single signal, so we can connect all the thermistors to a shared data pin in the Adalogger. In preparing these sensors, our objective is to join the three units together with as little excess wire as possible but with enough slack that moving and positioning the probes do not put any extra stress on the solder joints. Note, that while there may be variation in the color of the data wire, black will always be ground and red will always be VCC

1. Cut the wire on each temperature probe approximately 5 cm from the probe-end terminus.
2. Strip 2 cm of outer insulation, taking care not to break the inner wires.
3. Strip a centimeter of insulation from each inner wire, exposing the metal strands.
4. Twist all three black wire strands together and “tin” the wires by adding solder to the entire length. This will provide a base for more wire to be soldered to the temperature sensor packet. Do the same for the red and yellow wires, respectively (the colors may vary based on manufacturer, but the important thing is to ensure that matching colors are bundled in sets of three).

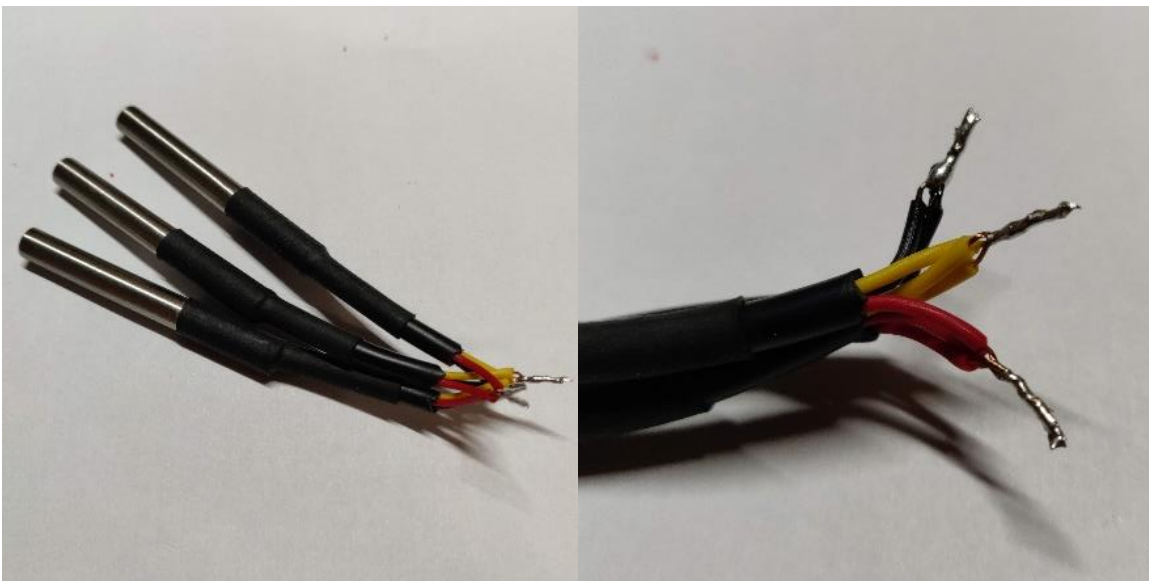


Figure 15. Three potted temperature sensors joined together with a close-up on the tinned tips.

6.4 Conductivity Sensor

The Atlas Scientific conductivity probe comes in both long and short varieties, with either paired leads that terminate at a large BNC connector or shrouded leads that terminate at a smaller SMA connector. Either version is acceptable for the OpenCTD and the connector will be removed during construction.

1. Cut the probe wire such that the full length of the probe and wire extends a little more than $2/3$ ds of the way up the PVC pipe.
2. Use a utility knife to strip and expose 2 cm of wire.
 - a. The wires may be co-axial (one wire is wrapped around the insulated core of the other) in which case you will have to separate the shroud from the central core and expose both. The shrouded wire will need to be twisted together once extracted from the insulation. Ensure that the insulation in the internal wire extends far enough that that wire will not make contact with the exposed external wire.
 - b. The wires may be embedded in extruded insulation rather than simply wrapped and will require a bit of care to extract without damaging the internal insulation.



Figure 16. Conductivity probe leads with coaxial wires stripped and soldered.

3. Strip half a centimeter of insulation from each inner wire.

4. Tin the exposed wire stands with a small amount of solder.

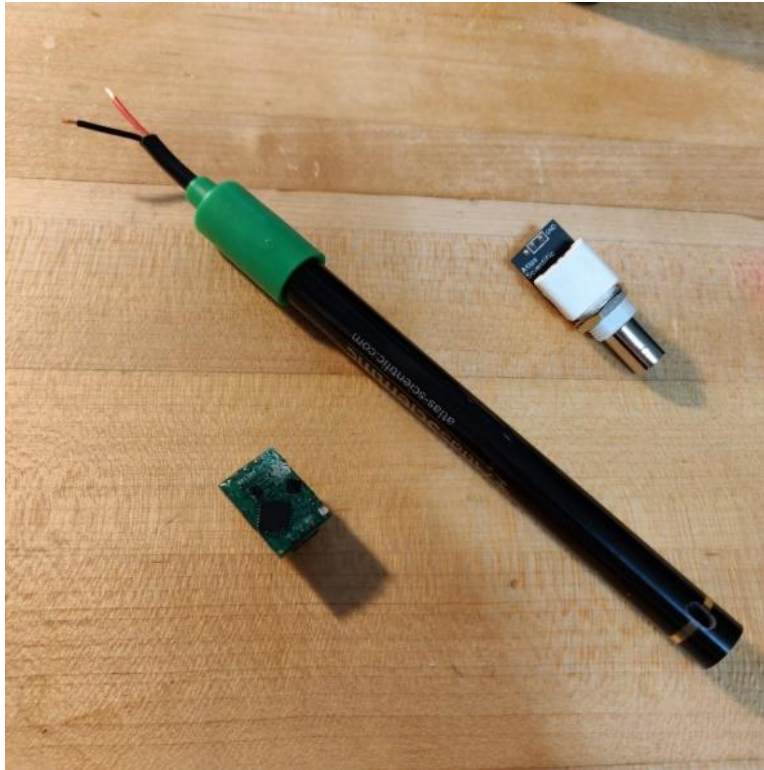


Figure 17. Long conductivity probe with stripped lead wires, Atlas EZO conductivity circuit, and BNC connector for simplified testing.

6.5 Benchmark: Breadboarding the Circuit (Protoboard)

A breadboard is a tool that allows you to prototype an electronic circuit without permanently affixing components. A breadboard is an electronic pegboard with three major regions. The two strips on either side of the board are the ground and positive rail. They are connected vertically. The rails in between are for affixing components. They are connected horizontally and usually clustered in groups of 5. In the center of the breadboard is a dividing groove. Horizontal rails are not connected across the groove. This allows you to mount large components with pins on either side of the central groove, like, for example, an Adalogger. By breadboarding the circuit first, you can ensure that the system is working properly before continuing.

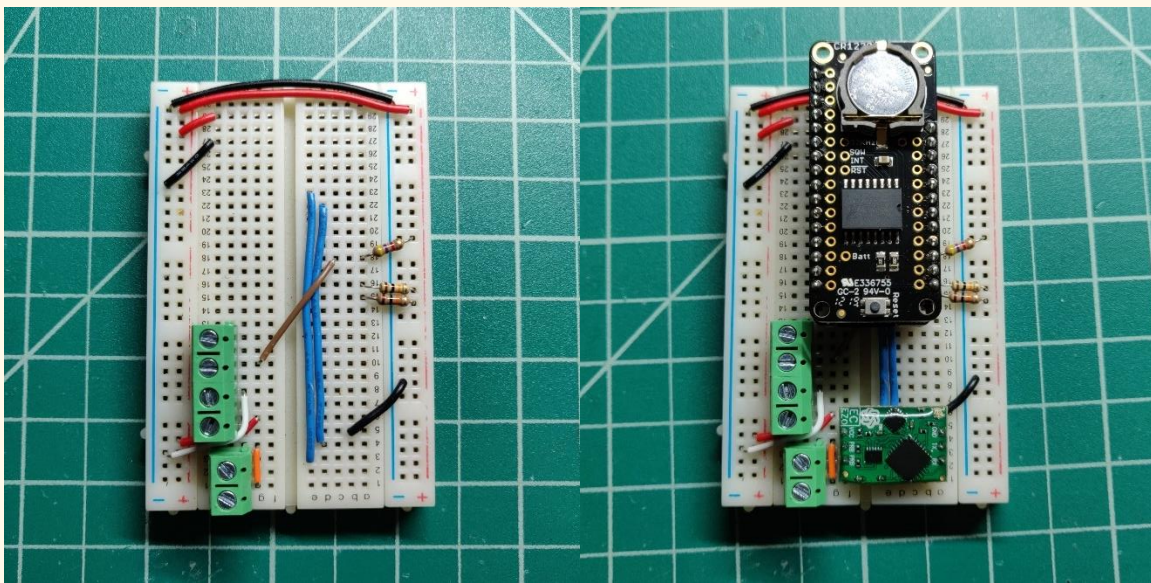


Figure 18. Breadboard with just jumper wires, screw terminals, and resistors attached and with Adalogger/RTC and EZO conductivity circuit.

1. Reference Figure 9 and the photo and pinout guide below. Place Atlas EZO circuit and connect jumper wires in accordance.
2. Connect a $4.7\text{ k}\Omega$ resistor between pin D6 and the positive voltage rail.
3. Connect a $10\text{ k}\Omega$ resistor between the SDA pin and the positive voltage rail and connect another $10\text{ k}\Omega$ resistor between SCL pins and the positive voltage rail.
4. Connect the temperature sensors to digital pin 6, positive, and ground using the screw terminal blocks.

5. Connect the pressure sensor to SCL, SDA, 3.3V, and GND. The solid core wire on the pressure sensor can be loosely slotted directly into the pin holes on the RTC.
6. Connect the temperature probe to the two probe pins on the Atlas EZO using the screw terminal blocks or BNC connector. It does not matter which wire goes to which probe pin. Make sure the conductivity probe itself is submerged or it will read zero.
7. Connect the Adalogger to your PC using a micro-USB cable and open the serial monitor in Arduino IDE.
8. Power the Adalogger down, remove the SD card, and read the data file.

Breadboard Diagram

Note: this diagram does not show the screw terminals. Refer to Figure 13 to see the position of the screw terminals.

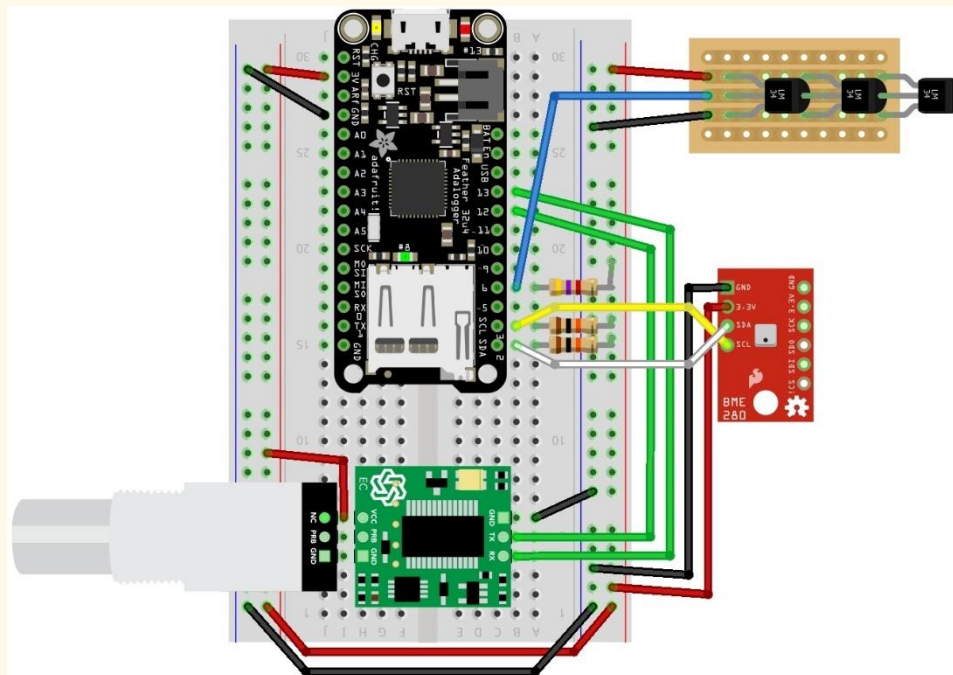


Figure 19. Diagram of breadboard assembly to test completed sensors. Note the presence of BNC connector for the conductivity probe and the use of a barometric pressure sensor to stand in for the OpenCTD pressure sensor.

Checklist

- Atlas EZO circuit is blinking with alternating between green and blue lights.
- Temperature probes are reading ambient temperature and are within 0.5°C of each other.
- Pressure sensor is reading atmospheric pressure (it should be around 1012 mbar if you're near sea level).
- Conductivity sensor is reading the conductivity of whatever fluid it is in.
- Data had been logged to SD Card.

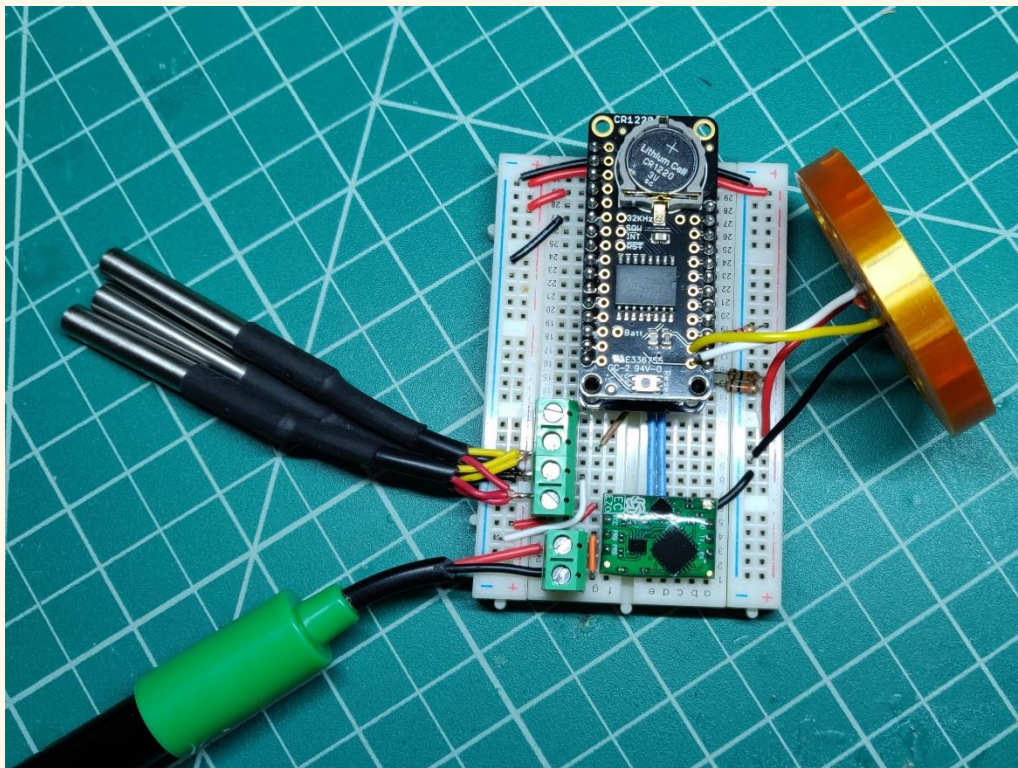


Figure 20. Completed breadboard set-up with all sensors ready for testing.

TESTBENCH

7 (Protoboard) Assembling the Protoboard Control Unit

The protoboard control unit replicates the traces embedded in the custom carrier board. It handles all the processing and signal routing between the sensors and the Adalogger. The protoboard control uses a mint-tin sized permanent breadboard, which is arranged similar to a standard breadboard, but with positive and negative rails in the center of the board and free, unattached pins at either end. This Perma-Protoboard allows you to make permanent electronic connections while mimicking the look and set-up of a breadboard.

7.1 Preparation

First, we will build the connector that interfaces with the sensors in the OpenCTD housing by attaching an 8-pin, right angle female connector to the end of the Perma-proto board and bridging it to the positive and negative rail with solder.

1. Once the Adalogger has been tested, it no longer needs the long pins that connected it to the breadboard. Snip these pins off at the base of the Adalogger. The Adalogger should now sit flush inside the Electronics Chassis.
2. Cut an 8-pin length from a right-angle female header strip.

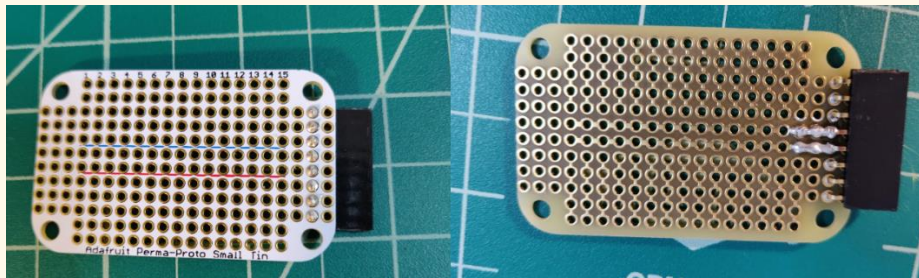


Figure 21. Perma-Protoboard with 8-pin connector top (A) and bottom, with solder bridges (B).

3. Solder the 8-pin connector to the second to last set of through-holes on the right side of the Perma-Protoboard, on the underside such that the pins emerge through the middle column of unconnected through-holes.
4. On the underside of the board, build a solder bridge between the positive and negative strips and their respective, adjacent connector pins. This works best with a soldering iron heated to 750°C. Hotter irons will cause the surrounding solder to melt.

7.2 Powering the Protoboard

Solid core wire will be connected to pass-through holes on the RTC and soldered underneath so that the wires pass between the RTC and the Adalogger and then contact the appropriate pins on the perma-proto board. It is easiest to start with long lengths of wire and cut down to appropriate sizes as you proceed.

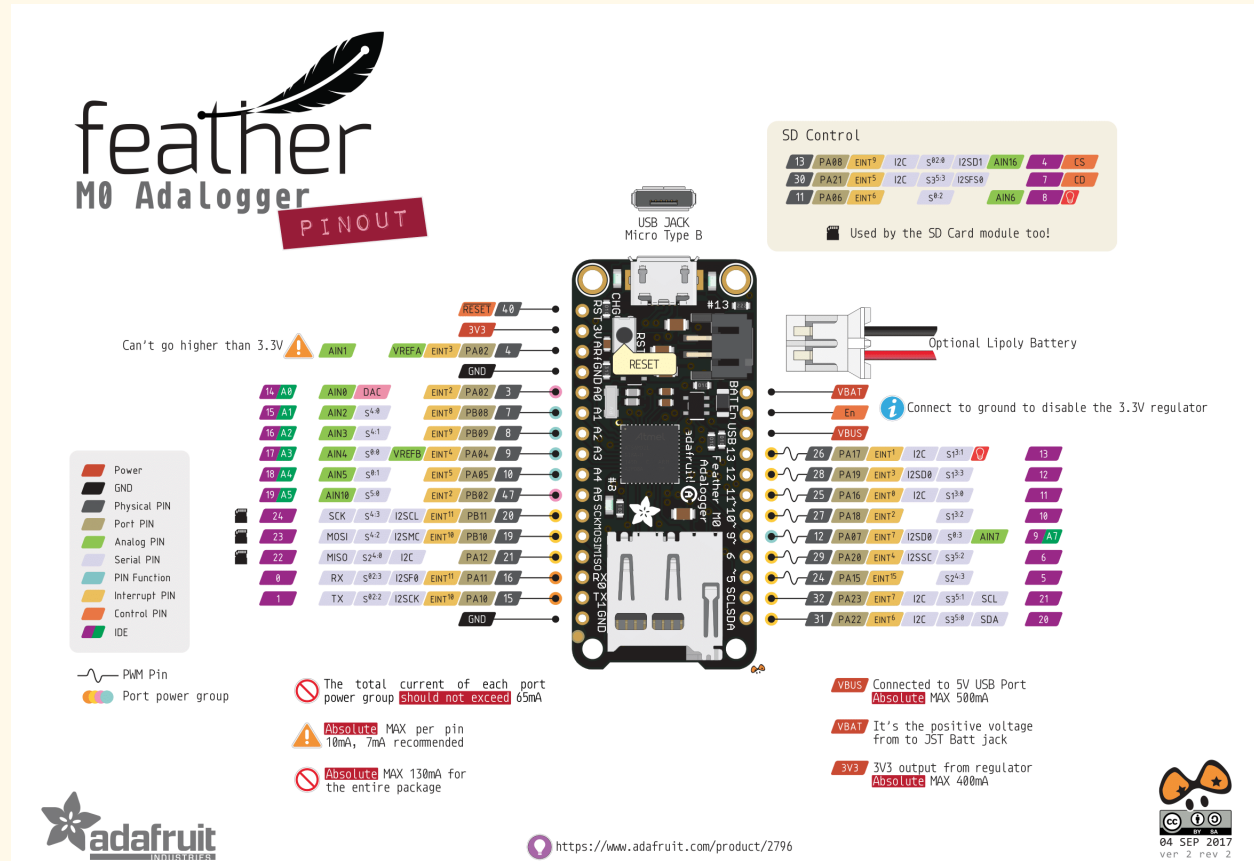


Figure 22. Pinouts for the Adalogger Feather M0. Note, these pinouts also correspond to those of the RTC.

1. Connect red and black solid core wire to the 3.3V and GND pins on the left side of the RTC.
2. Place the complete Adalogger/RTC assembly into the Electronics Chassis and stretch these two wires until they meet the end of the Chassis.
3. Cut the wires near the far end of the zip-tie hole in the e-chassis and strip off half a centimeter of insulation from the leads.
4. Solder these wires to the positive (3.3V) and negative (GND) rails of the Perma-Protoboard along columns 5 or 6.

5. Check the fit by placing the entire unit in the Electronics Chassis. The Adalogger and the Perma-Protoboard should sit flush against their respective support structures.

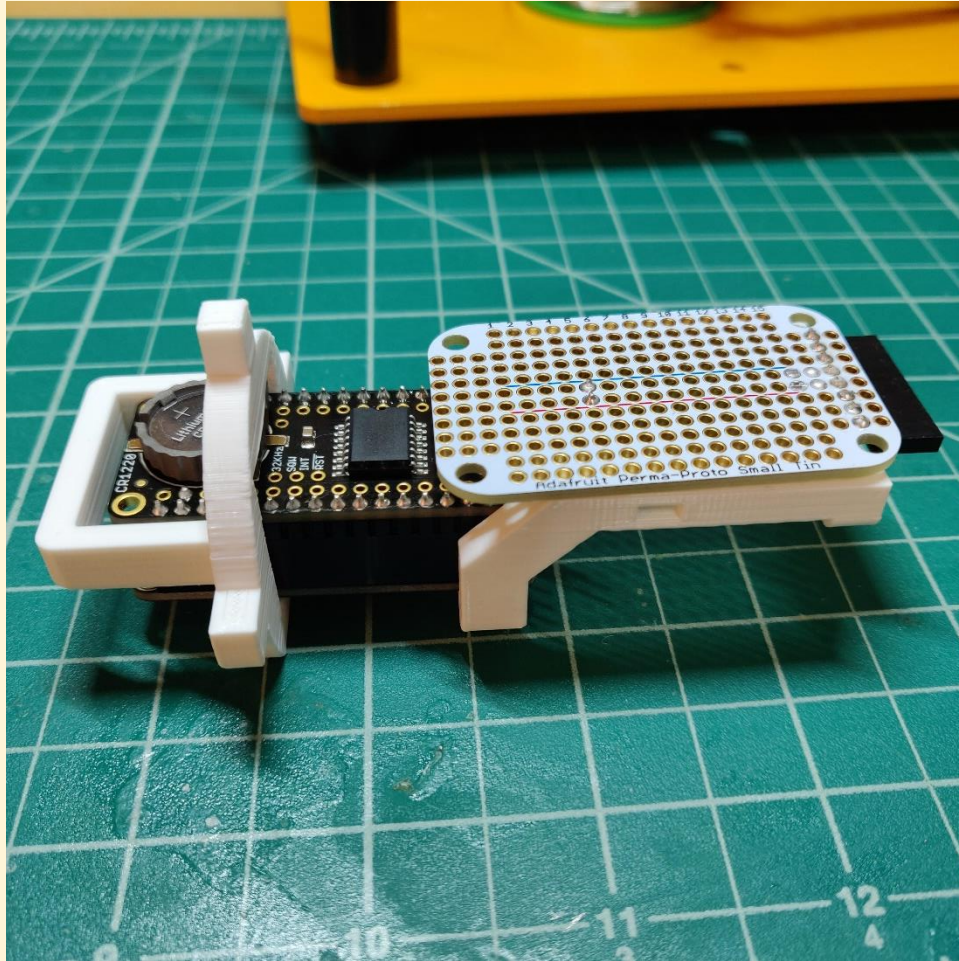


Figure 23. The Adalogger/RTC fit into the electronic chassis.

7.3 Populating the Perma-Protoboard

Skills: Soldering

Consumables: Solder, solid-core wire

Parts: Adalogger/RTC connected to Adafruit Perma-proto board, Atlas EZO circuit, 10 kOhm pull-up resistor (x2), 4.7 kOhm pull-up resistor

These steps will fully populate the protoboard control unit with all necessary components for the OpenCTD. We advise laying out all components first and double checking their placement before soldering. Reference Figures 23 and 24 frequently during assembly.

1. Solder the Atlas EZO circuit to columns 1 through 3 on the Perma-protoboard such that the GND and VCC pins correspond to the correct power rails. Holding the board so that “Adafruit Perma-Proto Small Tin” is correctly oriented, GND/TX/RX should be on top and VCC/PRB/PRB should be below.
2. On the underside of the Perma-proto board, bridge the GND pin of the Atlas EZO circuit to the negative rail with solder.
3. On the underside of the Perma-proto board, bridge the VCC pin of the Atlas EZO circuit to the positive rail with solder.
4. Using solid core wire, connect pin 12 on the RTC to TX on the Atlas EZO.
5. Using solid core wire, connect pin 13 on the RTC to RX on the Atlas EZO.
6. Connect each PRB pin to the through holes adjacent to pins 7 and 8 on the 8-pin right-angle header. It does not matter which PRB pin connects to which of the two header pins.
7. Connect the 4.7 kOhm pull-up resistor to column 8 and the positive (red) rail.
8. Connect a 10 kOhm pull-up resistor to column 10 and the positive (red) rail.
9. Connect a 10 kOhm pull-up resistor to column 11 and the positive (red) rail.
10. Using solid core wire, connect pin 6 on the RTC to upper column 8 on the Perma-proto board.
11. Using solid core wire, connect the SCL pin on the RTC to upper column 10 on the Perma-proto board.
12. Using solid core wire, connect the SDA pin on the RTC to upper column 11 on the Perma-proto board.
13. Using solid core wire, connect upper column 8 on the Perma-proto board to the through hole adjacent to pin 1 on the 8-pin right-angle header.
14. Using solid core wire, connect upper column 10 on the Perma-proto board to the through hole adjacent to pin 2 on the 8-pin right-angle header.
15. Using solid core wire, connect upper column 11 on the Perma-proto board to the through hole adjacent to pin 3 on the 8-pin right-angle header.

16. Starting from the inside pins closest to the positive and negative rails, bridge the 8-pin right-angle header pins to the respective adjacent solid core wires on the perma-proto board with solder.
17. Clip off the one unused pin from the male end of the sensor connector and slot it into the unused female header on the protoboard control unit. This will prevent the connector from being inserted into the protoboard control unit backwards, causing a short.
18. Using zip ties, attach the Adalogger/RTC and Perma-proto board to the 3D-printed electronics chassis.

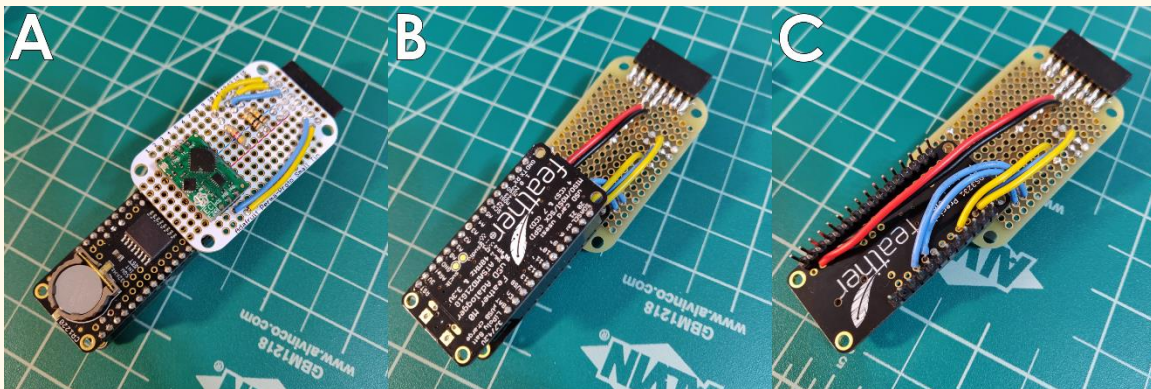
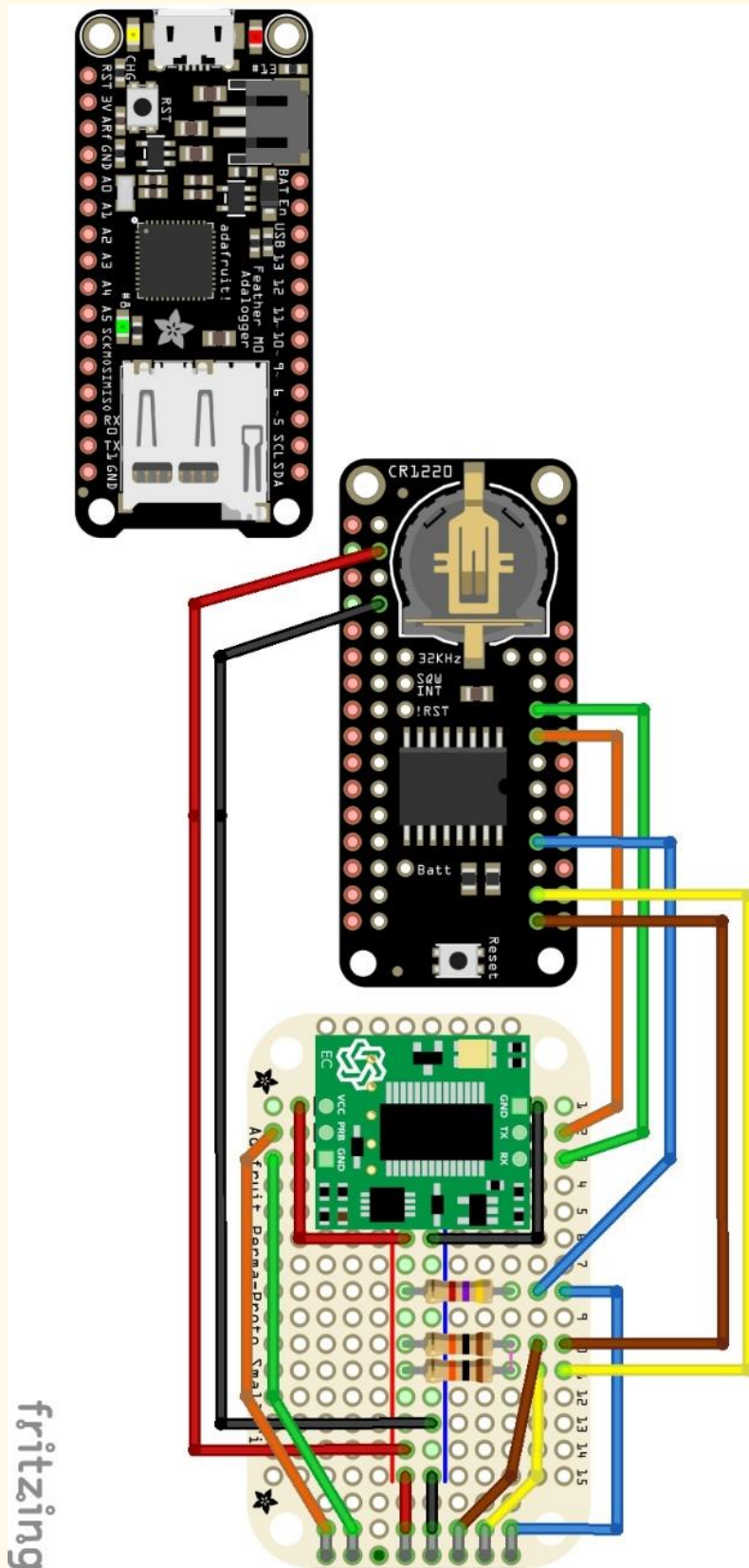


Figure 24. Fully connected CONTROL UNIT shown from the top (A), bottom (B), and bottom but without Adalogger attached (C).



fritzing

Figure 25. Layout of connections on the Perma-Protoboard.

7.4 Benchmark: Testing the Control Unit (Protoboard)

This test will ensure that the Control Unit has been assembled correctly and that all sensors are being read. Once you confirm that the protoboard control unit functions you will be ready to move on to the final phase of the OpenCTD build.

1. Holding the Perma-Protoboard face up with the connector on the right, the order of pinouts is, from top to bottom, D6 (Temperature), SCL (Pressure), SDA (Pressure), Ground, 3.3 Volt, UNUSED, Probe (-), Probe (+). Note that, as the conductivity probe functions like a resistor, it does not matter which line from the probe connects to which of the two pins.
2. Using alligator clips and jumper cables, or by soldering temporary headers onto each sensor lead, connect each sensor to its corresponding pin on the control unit .
3. Connect the Adalogger to your computer and check the serial monitor. If all sensors are reading accurately, you're ready to prepare and pot the sensors.

Checklist

- Protoboard control unit can read pressure sensor when connected.
- Protoboard control unit can read temperature sensors when connected.
- Protoboard control unit can read conductivity sensor when connected.

8 Building the Sensor Package and Potting Sensors

Potting the sensors is the point of no return. Once they are sealed in epoxy, they cannot be removed. It is essential that you confirm all sensors are operating correctly before final potting. You can test the pressure sensor before potting using alligator clips or jumper wires and touching them directly to their corresponding pins on the Adalogger.

8.1 Baseplate and Potting the Pressure Sensor

The 3D printed baseplate provides a template for accurately aligning all five sensors as well as an end-stop to prevent epoxy from leaking out during potting. The pressure sensor sits in the center of the template and fits snugly into the central hole. This sensor will be potted first with instant epoxy to protect the circuits before the full sensor assembly is potted. Be extremely careful not to get any epoxy on the gel of the pressure sensor.

1. Seat the pressure sensor into the central hole on the side with the large square indentation and ensure that the actual sensor sits flush with the bottom face of the baseplate. Take extra care not to apply too much pressure to the sensor housing. The metal ring that contains the gel cap can become detached if it experiences too much sideways force. If the sensor will not seat into the baseplate, use a knife, deburring tool, or other cutting instrument to slightly widen the opening until the sensor seats. Do not try to force it in, it should be a snug but comfortable fit.
2. Once the sensor is correctly seated, apply just enough instant epoxy to fill the entire central rectangular indentation. The epoxy will settle during pouring as it flows under the circuit board.
3. Carefully check that no epoxy is leaking out through the central hole.
4. Let the assembly sit for a 10 to 20 minutes to allow the epoxy to set.

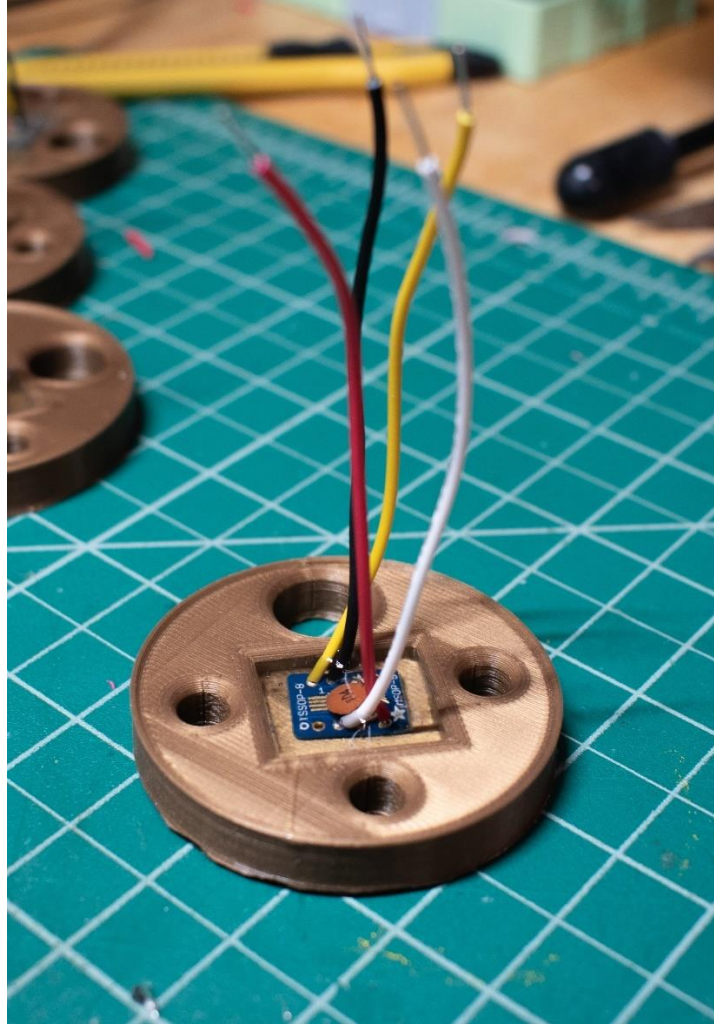


Figure 26. Pressure sensor seated and potted in baseplate.

5. Once the epoxy has set, insert each temperature probe into the three holes such that 1.5 to 2 cm extends out of the bottom of the baseplate.
6. The Pressure and temperature sensors share power. Solder the 3V3 (or VCC) wire from the pressure sensor to the positive (red) wire cluster of the temperature sensor assembly.
7. Solder the GND wire from the pressure sensor to the negative (black) wire cluster of the temperature sensor assembly.
8. Solder additional wire on to extend the GND, 3V3, DATA (the third wire on the temperature sensor cluster), SCL, and SCK wires to align with the wires on the conductivity probe. Seal each joint with heat shrink tubing.
9. Insert the conductivity probe into the large hole such that 1.5 to 2 cm extends out of the bottom of the baseplate. If the probe will not seat into the baseplate,

use a knife, deburring tool, or other cutting instrument to slightly widen the opening until the sensor seats. It will be a very snug fit.

8.2 Attaching JST connector

Depending on where you sourced the JST connector, the color-coded wires may vary from those shown in this guide. However, the connection process is simple. Remember, for joints sealed with heat shrink, you will want to slide the heat shrink tubing over the wires before closing the loop by soldering.

1. Examine the pin labels on the carrier board and write down which color wire on the JST female connector corresponds to which pin on the following table. Note that one pin is labeled NC for “no contact”. This pin is used when building the 8-pin header connector and should be ignored when using the 7-pin JST connector.

Pin	Color on Female JST	Color on Sensor Baseplate
D6		
SCL		
SDA		
GND		
3V3		
PRB		
PRB		

- a. **ALTERNATIVE:** If you are crimping your own JST female connector with wires, note which wire colors you use for each connection.
2. You will want the JST connector to stick out about 8 centimeters from the top of the PVC pipe.
 3. Solder the two wires from the conductivity probe to the wires on the JST connector that correspond to the pins marked PRB. As the conductivity probe is bidirectional, it does not matter which of the two PRB leads connects to which pin. Seal the soldered joint with heat shrink. You may need to use several layers of heat shrink to insulate the coaxial portion of the wire.
 4. Solder the GND and 3V3 wires to the leads for the positive and negative terminals. Seal the solder joint with heat shrink.

5. Solder the SDA, SCL, and D6 wires to the leads for SDA and SCL on the pressure sensor and DATA on the temperature sensors. Seal the solder joint with heat shrink.
6. Using a large piece of heat shrink, seal the entire area around the solder joints for added protection.

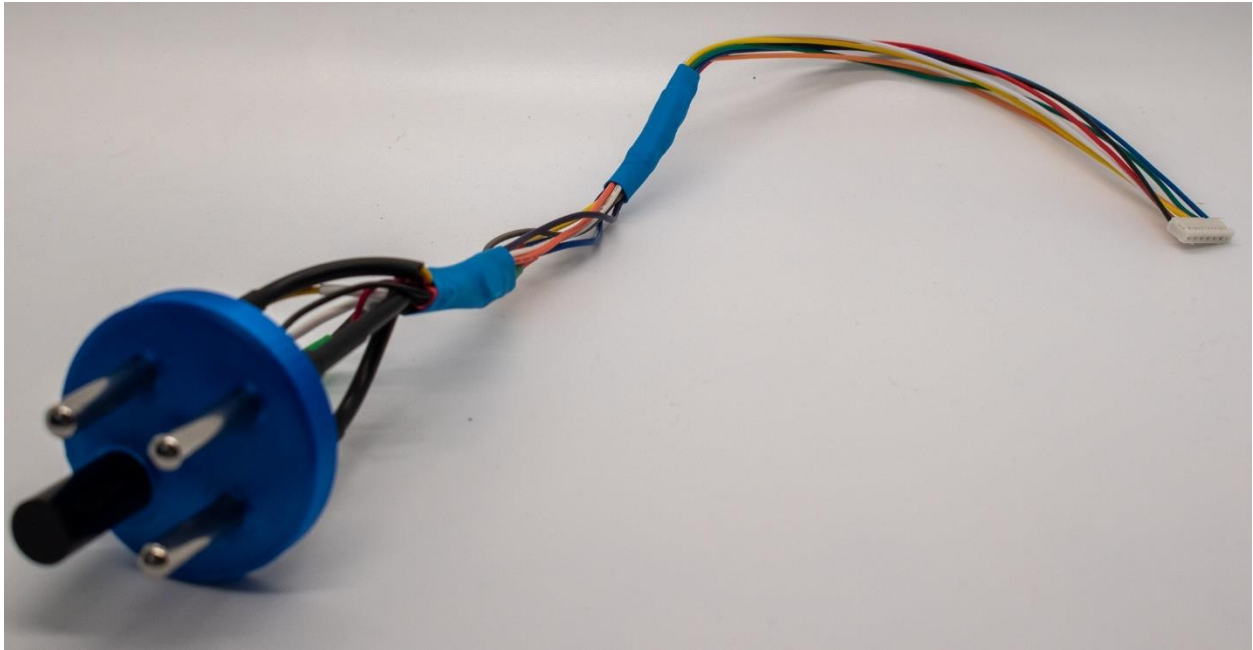


Figure 27. Completed sensor package.

8.3 Creating ad-hoc connector

The 8-pin male connector makes it easy to attach and detach the sensors from the Control Unit using standard 8-pin headers. While not ideal, this is a useful workaround when parts are limited. At the end, you will have 7 lines coming from the sensors, including a positive and ground line, SDA and SCL from the pressure sensor, data from the temperature probes, and two lines from the conductivity probe.

1. Solder a 5 cm length of 20-gauge wire to the positive (VCC) and ground (GND) wires of the pressure sensors and seal the connection with heat shrink tubing.
2. Solder the positive wires from the pressure and temperature sensors together, add a 15 cm length of wire to each joined end and seal with heat shrink.
3. Solder the ground wires from the pressure and temperature sensors together, add a 15 cm length of wire to each joined end and seal with heat shrink.
4. Attach a 25 cm length of 20-gauge wire to the SDA and SCL wires of the pressure sensor and seal with heat shrink.
5. Attach a 15 cm length of 20-gauge wire to each of the two wires of the conductivity probe and seal with heat shrink.
6. All wires should extend about 1 cm beyond the edge of PVC pipe when the baseplate is dry fit into position. Cut off any excess so that all wires are of equal length. Use electrical tape to tape the free wires to the side of the conductivity probe. This will relieve strain on the solder joins.
7. Strip and tin the ends of the sensor wires. Cut the braided sleeve to length and slide down over the sensor wires. This will provide additional protection to the wires and solder joins.
8. Prepare an 8-pin header by tinning the short ends of each pin.
9. Solder the sensor wires to an 8-pin header. They will align with their respective contacts on the Control Unit 8-pin female output. Take care not to apply too much heat to the header or the plastic strip will melt. Seal each pin with heat shrink tubing.

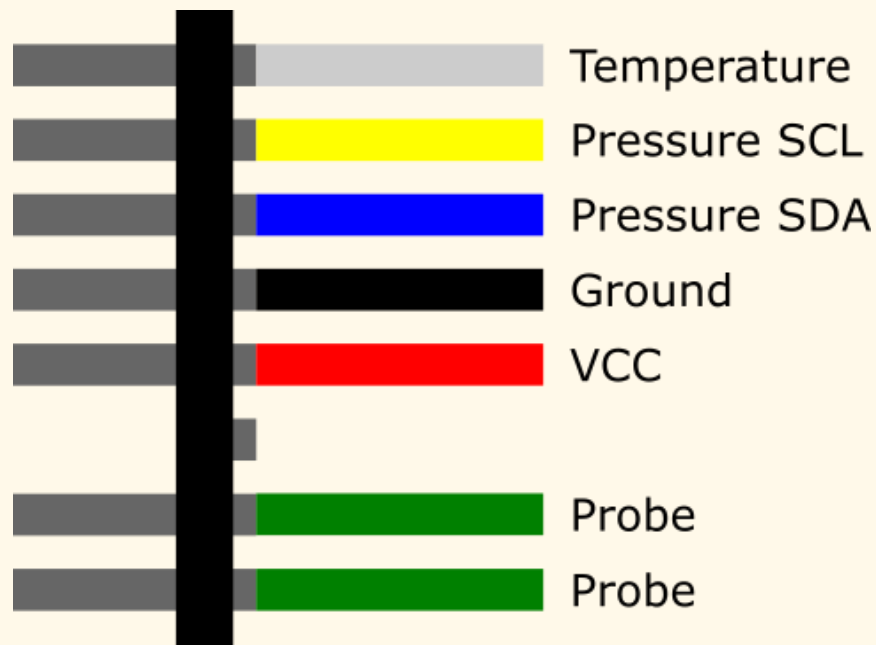


Figure 28. Illustration of connector with wires coming up from the housing. Wires leading to sensors on right, vertical black bar represents 8-pin headers. Pins leading to CONTROL UNIT are on the left.

10. Test the connector by plugging it into the breadboard and reading the sensors. You will have to rearrange the breadboard in order to connect to all seven leads. You can also test each sensor independently using jumper wires.
11. Slide the 3D-printed connector cover over the 8-pin connector.
12. Fill the entire void within the connector with 5-minute epoxy. This prevents strain on the header and solder joints as well as prevent moisture from entering the connector and prevents short circuits. Allow the glue to cure for 30 minutes.
13. Once cured, test the connection again by plugging it into the breadboard and reading the sensors. You will have to rearrange the breadboard in order to connect to all seven leads. You can also test each sensor independently using jumper wires.

8.4 Potting the OpenCTD

It is time to permanently seal the base of the OpenCTD. Once the epoxy has been mixed and injected, it is essential that you do not move or disturb the housing until the epoxy cures. Within the first 20 minutes, you may gently lift the housing vertically to check that no epoxy is leaking out the bottom. After about 2 hours of cure time, the epoxy should have set enough to allow you to continue the build process without disturbing it. Do not submerge the OpenCTD deeper than 10 meters until the epoxy has cured for at least 24 hours.

1. Ensure that you can slide the sensor baseplate up into the bottom of the CTD so that it sits just above the flow-through holes or notches with the sensors protruding out the bottom but protected by the housing. If the fit is too tight, use a deburring tool to knock down the edges of the 3D-printed parts.
2. Push the baseplate into the housing such that the sensors sit slightly back from the rim of the PVC pipe and baseplate clears any notches cut in the sides of the pipe.
3. Ensure that the inside of the PVC pipe is clean and free of obstruction.

**ENSURE EVERYTHING IS SET CORRECTLY
BEFORE PROCEEDING TO THE NEXT STEP.**

4. Using the 50mL cartridge gun and venturi mixing nozzle, inject the full 50mL of 2-part epoxy into the OpenCTD housing.
5. Check to ensure that nothing is leaking and let the epoxy cure undisturbed for up to 24 hours. We have generally found that an overnight cure is sufficient for testing in water less than 10 meters deep.



Figure 29. A clear OpenCTD showing the thick epoxy potting layer.

9 Powering the OpenCTD

The final step in assembly is building the battery switch and inserting the battery into the housing. The Adalogger has an internal 100mA charging unit, which means a standard 2000mAh battery will take 20 hours to charge. Using an external lithium polymer battery charger with higher amperage can dramatically speed up the charging process. For charging, we use this 500 mA charger (<https://www.adafruit.com/product/1904>) which cuts the charging time on a 2000mAh battery down from 20 hours to 4 hours.

There are two different options for power switches: a board mounted switch that sits inside the OpenCTD, and a magnetic switch that sits outside the OpenCTD. The magnetic switch is finicky and we recommend using the internal switch unless you have a specific need for an external power switch.

Be very careful with heat near a lithium-polymer battery.

9.1 Internal Power Switch

The power switch will be either mounted directly to the carrier board or glued into place on the 3D-pinted chassis for the perma-protoboard electronics assembly.

1. Solder the 3-pin toggle switch to the pins labeled J2 on the carrier board.
2. Snip the positive (red) lead from the battery about halfway up the connection, ensuring that there is enough space for the 2-pin JST connector to reach the battery connector on the Adalogger M0. Strip away a few millimeters of insulation.
3. Solder each end of the snipped positive lead to one of the two pins labeled J1 on the carrier board.
4. Plug in the battery and ensure that the switch functions correctly.
5. Tape the battery to the back of the carrier board so that it remains in place.

Protoboard Alternative

1. Adhere the 3-pin toggle switch to the associated slot in the back of the 3D-printed electronics chassis.
2. Snip the positive lead from the battery about halfway up the connection, ensuring that there is enough space for the 2-pin JST connector to reach the battery connector on the Adalogger M0.

3. Solder each end of the snipped positive lead to one of the two pins on the chassis, ensuring that one of the leads is soldered to the middle pin. And seal each connection with heat shrink.
4. Plug in the battery and ensure that the switch functions correctly.
5. Tape the battery to the back of the chassis in a convenient position for accessing the SD card.

9.2 External Magnetic Switch

The battery assembly is designed to seat independent of the rest of the electronics. This allows the battery leads to pass through a magnetic switch connected to the inner wall of the PVC tube and permits users to activate and deactivate the OpenCTD without opening the housing. The battery assembly is designed as a separate unit from the rest of the electronics to make it easier to replace the battery when necessary and to speed the rate of charging.

1. Assemble the inner switch assembly by running two 5 cm lengths of stripped solid core wire into the associated holes in the 3D-printed assembly.
2. Pinch the wires into place and superglue the outer, insulated wires to the edge of the assembly.
3. Clip off the battery leads about 2 cm from the base. Be careful not to let the leads contact each other.
4. Solder and seal enough wire to the battery leads to allow the battery to sit comfortably in the bottom of the CTD housing (about 10 cm). To reduce strain on the soldered battery leads, tape the wires to the body of the battery.
5. Now solder the wires with the JST connector to the switch assembly above the switch so that they emerge from the top of the CTD. Leave enough wire that you can easily attach and detach the JST connector from the Adalogger.

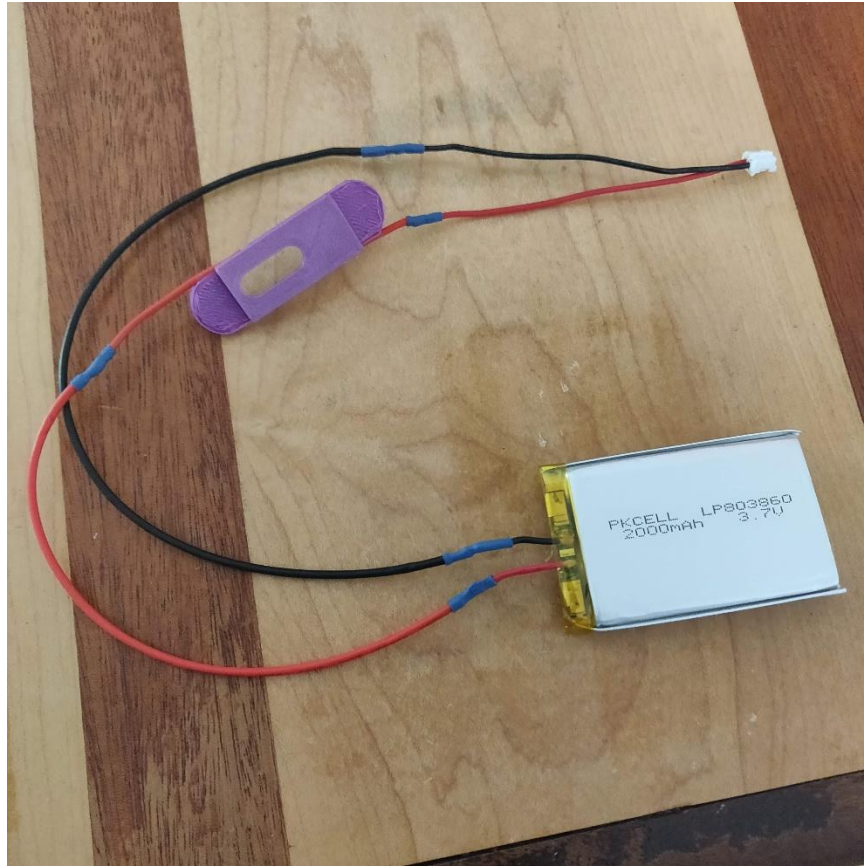


Figure 30. Inner battery switch assembly before battery is taped.

6. Place a 10 mm round magnet inside the housing (you can hold it in place by putting a magnet on the outside, too).
7. Using 10mm adhesive pads, adhere the internal switch component to the wall of the housing. Ensure that the internal magnet moves freely and contacts the two expose wires and that the switch assembly is seated low enough that it does not interfere with the test cap.
8. Stick the switch pin to another magnet using the 10mm double-sided pads.
9. Superglue the external switch assembly to the outside of the PVC pipe such that it traps the pin, is even and parallel with the internal assembly, and the pin can move freely. Line the switch up carefully and tape in place with electrical tape. Take extra caution not to allow the glue to leak into the inner walls of the external switch assembly or it could interfere with movement of the magnet.
10. Connect the control unit to the battery and ensure that the switch works consistently.

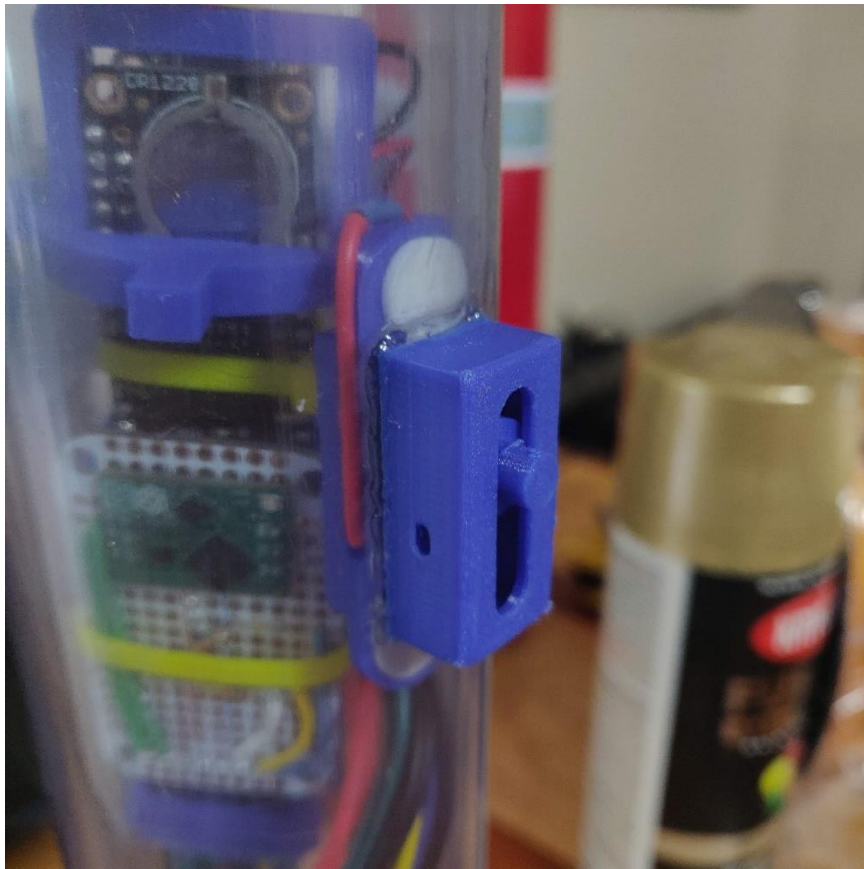


Figure 31. The complete switch assembly in a clear OpenCTD.

10 Casting and Ballast

Precisely how much weight a CTD needs for ballast and how it should be mounted to an instrument of line is entirely dependent on your specific needs. A CTD deployed in a brackish estuary to a relatively shallow depth can sink with no additional weights on something as simple as a handline. A CTD deployed to ocean depths in high current may need to be attached to a weighted anchor line with heavy-duty hose clamps.

Below, we provide a few possible mounting options for the OpenCTD. We recommend for most cases, using the casting loop and a medium-duty Polypro line.

10.1 Casting Loop

This is a basic, durable loop for clipping to a handline that can be constructed using a short length of Polypro rope and a 2" hose clamp, as picture below. For extra security and to avoid snags, we recommend that you also wrap the hose clamp in electrical tape.



Figure 32. Finished OpenCTD with optional casting loop.

The potted and sealed OpenCTD is positively buoyant. In order to make the CTD sink, you will need to ballast it with weights. 2 to 3-ounce lead fishing weights work well and can be carefully inserted into the CTD housing, taking care not to crush any wires or contacts. For coastal areas, we have found that 4 to 6-ounces is sufficient to make the OpenCTD slightly negatively buoyant, but the total amount of weight necessary will vary depending on salinity and water conditions.

10.2 Weighted Casting Line

For deeper deployments, or operations in high current, a weighted anchor line, with the anchor or weight held several feet below the CTD, will provide substantial negative buoyancy while keeping the unit horizontal. We recommend at least two hose clamps to ensure that the unit stay in position.



Figure 33. Example of OpenCTD attached to weighted anchor line using hose clamps.

10.3 Internal Ballast

The OpenCTD is positively buoyant in salt water. As a baseline, we recommend six ounces of standard fishing weights stored inside the housing. This will make the unit slightly negatively buoyant and able to sink in calm waters.

To avoid damaging the electronics, it is best practice to remove the weights between deployments so that they don't shake around during transportation. Wrapping them in rubber or electrical tape will also minimize movement.



Figure 34. Clear OpenCTD with fishing weights inside.

10.4 External Ballast

There are an infinite number of ways to affix weights to the outside of the OpenCTD. We've seen everything from lead flashing wrapped around the housing to small gym weights hung underneath. We recommend avoiding having external weights attached directly to the CTD. It is preferable to mount the entire OpenCTD to a heavier object for deployment.

11 Calibration and Data Management

Comprehensive validation of the OpenCTD requires access to professionally calibrated instruments. These can be commercial CTD, handheld systems, or bench-mounted laboratory systems. As the community grows, validated OpenCTDs can be used to benchmark newly built instruments.

It is not necessary to do a full validation test for every instrument, only those for whom a high degree of confidence and extreme precision are needed. For most use cases, an OpenCTD calibrated against salinity standards and general-purpose temperature sensor is adequate.

The template **OpenCTD_DataConversion_Template.xls** available in the OpenCTD GitHub repository will take raw data from the OpenCTD and make the necessary conversions for you.

11.1 Pressure (Depth)

The OpenCTD uses a 14-bar pressure sensitive chip from Measurement Specialties designed, initially, for SCUBA dive watches and depth gauges. This chip contains a pressure sensitive resistor embedded in a gel matrix. It is important that, while handling, not to damage the gel or allow anything to come in contact with it. Sand is particularly destructive, so take extra care when working from a beach. The pressure sensor is rated to 140 meters depth.

A 30-bar chip, which can double the operating depth of the OpenCTD, is also available, though requires additional customization.

PRESSURE ERROR VS PRESSURE AND TEMPERATURE

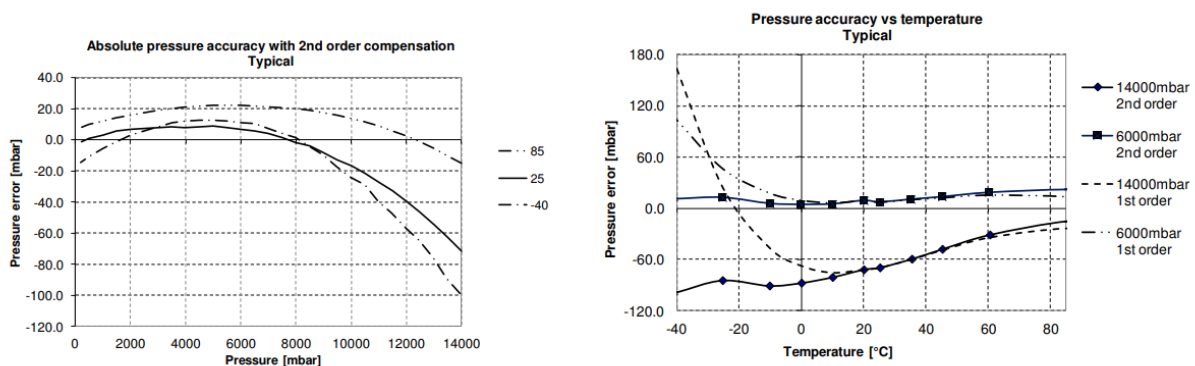


Figure 35. Pressure sensor error versus ambient pressure and temperature for the MS5803-14BA pressure sensor. Provided by manufacturer.

The MS5803 is a factory calibrated pressure sensor capable of outputting absolute pressure up to 0.2 millibar resolution and temperature with a resolution of 0.01 °C. The pressure sensor outputs

absolute pressure, which means that it is self-calibrating. At sea level, standard atmospheric pressure should be 1013.5 mbar, though local weather and small changes in altitude will affect this baseline. In order to minimize the amount of processing that happens onboard the OpenCTD, absolute pressure is measured in millibars (the default output for the current pressure sensor). To convert pressure to depth, use the following equation:

$$D = (P_{(DEPTH)} - P_{(SURFACE)} * 100) / (G * 1000)$$

Where **D** is depth in meters; **P_(DEPTH)** is the pressure (in millibars) at depth); **P_(SURFACE)** is the pressure at the surface in millibars; and **G** is acceleration due to gravity, which for most field purposes can be assumed as 9.81 m/s² (Fofonoff and Millard Jr, 1983). For high-accuracy and high-resolution commercial CTDs, the latitudinal variation in gravity is considered in the pressure to depth conversion as these sensors often offer sub-centimeter accuracy.

For extra accuracy, you can use the absolute pressure at the surface of the water at the time of deployment by taking the last pressure reading before the conductivity probe contacted the water for **P_(SURFACE)**.

11.2 Temperature

The OpenCTD uses a DS18B20 digital thermometer potted in a stainless-steel cladding. These sensors communicate over a 1-wire protocol, which allows us to connect multiple sensors to the same data pin. The sensors have an operating range of -55 to 125°C (with peak accuracy between -10 to 85°C). The advertised accuracy is ±0.5°C, however, in field trials using a battery of 3 sensors, we have seen accuracy as low as ±0.1°C.

DS18B20 TYPICAL ERROR CURVE

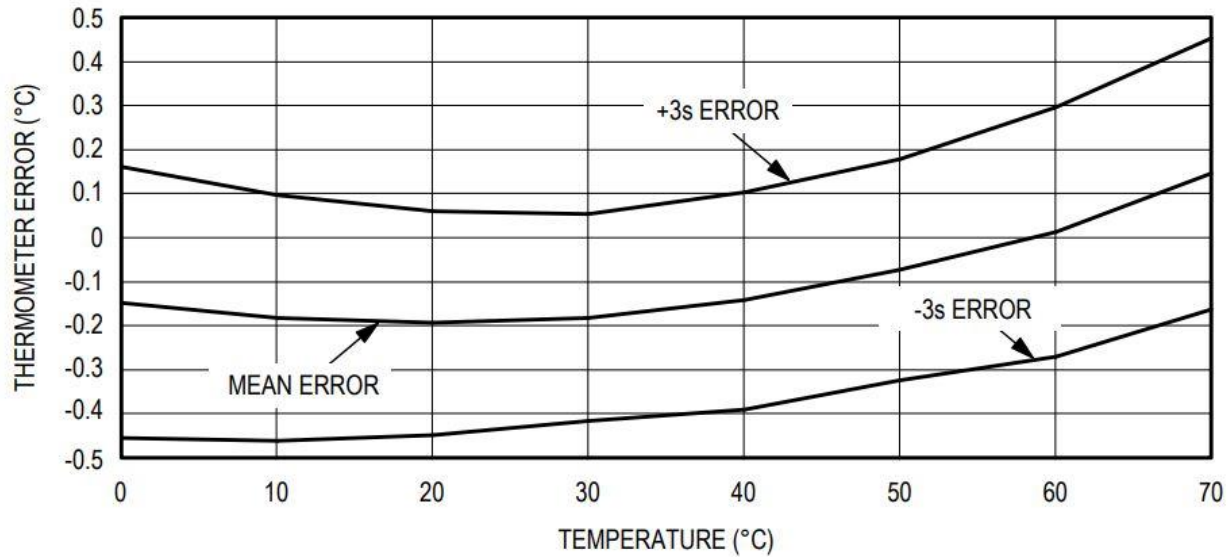


Figure 36. Typical error curve for DS18B20 digital thermometer. Provided by supplier.

Thermal time constant is a measure of how quickly the thermistor responds to changes in ambient temperature. It is expressed as the time it takes for a thermistor to cool to 63.2% of the total difference between a stable high temperature and a stable low temperature. We determined the thermal time constant for two different designs of DS18B20 sensors in stainless steel cladding: flat-topped and hemispherical. For sensors clad in a flat top housing, the time constant was 5.7 seconds. For those clad in a hemispherical housing, it was 8.5 seconds.

We also determined equilibration rate for DS18B20 sensors. While equilibration time is not a precise measurement, it is useful for making quick protocol adjustments in the field and gives a basic guideline for how quickly these thermometers react to temperature changes. For transitioning from low temperature to high temperature, the equilibration rate was 1.2 °C/s. For transitioning from high temperature to low temperature, the equilibration rate was 2.3 °C/s.

The DS18B20 temperature sensor is not factory calibrated and the user must perform a single-point or two-point calibration in order to achieve precision. The sensor offset is, however, linear, which means that a single calibration constant can be determined to correct the accuracy of the averaged measurement. A calibration constant can be established by plotting the average deviation between OpenCTD temperature sensors and an existing validated instrument or water of known temperature. The OpenCTD does not need to be submerged during this process.

Calibrated instruments should remain steady for many casts, but it is always good practice to check your temperature calibration whenever you suspect that the calibration may have changed. We recommend checking temperature calibration at the beginning of each expedition.

11.3 Conductivity (Salinity)

The OpenCTD uses a graphite conductivity probe manufactured by Atlas Scientific. Electrical conductivity probes are relatively simple devices consisting of two electrodes of known surface area and known distance from each other. As conductive fluid passes between the two electrodes, they measure the resistance of the liquid. The Atlas probe has a response time of 90% within 1 second, a 343-meter maximum operating depth, and a 1 to 110°C temperature range. The K 1.0 probes can measure conductivity from 5 to 200,000 $\mu\text{S}/\text{cm}$.

Salinity is derived following the formulas outlined by the 1978 Practical Salinity Scale (Lewis, 1980). Temperature and pressure are integral parts of this calculation, and as such it is recommended that the user use this method for determining salinity instead of using the salinity value derived by the Atlas-Scientific conductivity circuit. The accuracy of the calculated salinity value ultimately depends on the accuracy of each individual sensor.



Figure 37. Accuracy of Atlas Scientific K 1.0 Conductivity Probe and EZO circuit. Data provided by Atlas Scientific.

The conductivity circuit comes pre-calibrated from Atlas Scientific and in the vast majority of cases, it does not need additional calibration before deployment. Before initiating the calibration process, check the factory calibration against a known standard to determine if the unit needs calibration. Make sure to check the temperature specifications, most calibration standards are designed to be used at 25°C.

The conductivity circuit requires a two-point calibration using solutions with a known and precise conductivity. Calibration solutions can be pre-mixed or ordered from a scientific supplier. Atlas provides a set of calibration standards for their probes.

Two-point calibration is facilitated using the Serial_for_EC_Calibration_m0.ino software found in the OpenCTD GitHub repository. Calibration should be performed at 25°C.

1. Flash Serial_for_EC_Calibration_m0.ino to the Adalogger.
2. Open the serial monitor through the Arduino IDE. You should see a stream of unprocessed data. Make sure the dropdown menus in the serial monitor are set to “Carriage return” and “9600 baud”.
3. Enter **c,0** in the command line and hit enter. This will turn off continuous monitoring.
4. Enter **k,?** in the command line and hit enter. This will tell you what the K-value of your probe is set to. It should be set to K=1.0 for most use cases.
 - a. To change the K-value, enter **k,x** (where x is the new K-value of the probe) in the command line and hit enter.
5. Clear the previous calibration data by entering **cal,clear**.
6. With the probe dry (be sure there is absolutely no water on the electrodes) enter **cal,dry** in the command line and hit enter. This will dry calibrate the probe.
7. Re-enable continuous monitoring by entering **c,1**.
8. For two-point calibration, suspend the probe in the less conductive solution (ideally 12,880 μS). Leave the probe suspended in the solution until readings stabilize. Enter **cal,low,y** (where y is the known value of the standard) in the command line and hit enter.
9. Rinse the probe thoroughly in fresh water. Suspend the probe in the more conductive solution (ideally 80,000 μS). Leave the probe suspended in the solution until readings stabilize. Enter **cal,high,z** (where z is the known value of the solution) in the command line and hit enter.

Alternative: For 1-point calibration, suspend the probe in known calibration solution. Leave the probe suspended in the solution until readings stabilize. Enter **cal,one,y** (where y is the known value of the standard) in the command line and hit enter. Only do this if you do not have two calibration standards.

10. Reflash the OpenCTD software to the Adalogger after calibration is done.

Conductivity should not need to be calibrated again once the calibration protocol is complete, however it is good habit to check calibration at least once per year or whenever components are changed. For more detailed explanation of the calibration process and for an explanation of temperature compensation, refer to the Atlas EZO-EC Datasheet (https://atlas-scientific.com/files/EC_EZO_Datasheet.pdf).

11.4 Data Management

The OpenCTD outputs data as a tab delimited text file which can be read by any standard spreadsheet program. It outputs the date and time followed by the pressure sensor in mbar, each individual temperature sensor in °C, and the conductivity sensor in microSiemens. In order to convert those values into human-readable oceanographic data, we have produced an Excel template that takes the raw data and outputs depth in meters, average temperature, and practical salinity units. It will also average readings in batches of 60 to simplify data presentation. It has fields for calibration constants so that users can input the absolute pressure at sea level at the time of deployment and the temperature offset of the CTDs temperature probes determined experimentally during the calibration phase.

Although the SD card reader is built into the Adalogger, Arduino M0 microcontrollers do not provide data passthrough, which means the SD card cannot be read directly from the Adalogger without flashing a secondary program to the microcontroller. Data is accessed post deployment by removing the SD card from the Control Unit and connecting it directly to a computer.

Please note that due to noise in the system, the conductivity circuit can create a bad data entry with additional, non-numerical characters. These are normal and can either be cleaned-up during data processing or discarded.

Importing into Excel: Open the Excel template and navigate to the Data Input sheet in the lower left corner. You can then either use the Get Data function in the Data menu to directly import the entire text file to the sheet or copy-paste the desired data points directly onto the sheet. If you use the Get Data function, Excel will create a new sheet for the data. Regardless of how you import the data, Excel will import it all as a single column. Highlight that column and then, under the Data tab, use the Text to Columns feature to split the data into individual columns by selecting Delimited, clicking Next, checking the boxes for Space and Treat consecutive delimiters as one, and clicking Finish.

The data is now in Excel-readable format and can be pasted into the Paste RAW data from OpenCTD columns on the Calculations sheet.

A Note about Macs: Depending on the version of MacOS, the default text editor may not be able to open the datafile produced by OpenCTD without modifying the editor's settings. We recommend downloading a robust, 3rd-party text editor if you plan to use a Mac to handle OpenCTD data.

11.5 Stewardship

Whenever you introduce a new tool into the marine environment, you run the risk of introducing additional harms to the ecosystem. Though OpenCTDs can serve as a valuable tool for ocean conservation, they also have the potential to negatively impact marine animals and ecosystems. We have developed a set of guidelines for minimizing potential negative impact when introducing new technologies into marine ecosystems (Thaler et al., 2019, 2015). Though these guidelines specifically address small underwater robots, they are equally applicable to OpenCTDs and other instrumentation platforms.

We ask that you read the following two short papers and approach your project with a stewardship ethic and an eye towards ensuring that your work will not contribute additional harms to the ecosystems that you are working in.

- Robots as Vectors for Marine Invasions: Best Practices for Minimizing Transmission of Invasive Species Via Observation-Class ROVs
(<https://journals.sagepub.com/doi/full/10.1177/194008291500800308>)
- Bot Meets Whale: Best Practices for Mitigating Negative Interactions Between Marine Mammals and MicroROVs
(<https://www.frontiersin.org/articles/10.3389/fmars.2019.00506/full>)

12 The First Deployment

It is finally time to make your first cast!

The OpenCTD's maximum operating depth was established both through field trials and tests in a pressure chamber. The maximum operating depth of the sensor system is 140 meters; however, the housing has only been tested consistently to 100 meters depth without leaks or catastrophic failure. Because each OpenCTD will be slightly different, using the 100-meter benchmarks provides a good margin of error for operation. If you are unsure of the robustness of your own build, you can test your CTD by submerging just the housing without the control unit to the desired depth and confirm that the unit did not flood before deploying with the battery and control unit.

If the CTD housing does flood, immediately flush the interior and JST connector with distilled water and dry thoroughly. As long as there is no corrosion on the connector and it is dry when under load, there should be no damage to the electronics.

A brand new OpenCTD will be positively buoyant. Additional weight can be added into the housing or attached to the outside. Lead shot or fishing weights are particularly useful for this. If you are doing a line cast, you will need to secure the CTD to a sturdy line using hose clamps, waterproof tape, or other adhesive solutions. You will want to experiment with different weights to find the ideal ballast for the conditions that you will be deploying in.

Check that the inside surface of the PVC pipe is clean and free of debris. To ensure no leaks, the test cap used to seal the unit should be fully seated with no obstructions between the pipe wall and the rubber gasket and be hand tight. Seat the test cap onto the housing and tighten the wingnut. The wingnut should be as tight as possible without tool assistance and the cap should not move or turn. Over time, salt may build up on the wingnut, causing the crew to bind. WD-40 or other water displacing lubricants can help loosen up the screw.

Your OpenCTD is now ready to take the plunge! Gently lower into the water and descend no more than 1 meter per second. The slower the better. For the first cast, descend to 5 meters without the control unit, hold for several minutes, then recover and check that no leakage has occurred. Once it passes the first soak test, you're ready to insert the control unit and collect ocean data.

While calibration plays a significant role in affecting data quality, response times and the method of deployment also have an influence. The DS18B20 clad in stainless steel has a time constant of 5.7s. The conductivity sensor maintains a response time of approximately one second for a 90% value (Atlas-Scientific, 2017). The pressure sensor response time is variable between 0.5 to 8.22 m/s depending on the resolution selected by the user (Measurement Specialties, 2012). In order to ensure that the device has reached relative equilibrium with the surrounding water, it is necessary to leave it at station for several minutes before beginning an in-situ deployment. When

faster response time is required, the sensors used in the most recent version of the OpenCTD can easily be replaced by more accurate, but also more expensive alternatives.

OpenCTD Deployment Checklist:

Preparation

- Battery is charged (note, CTD switch must be in the on position for the device to charge)
- Sensors are calibrated

Pre-deployment

- SD card is inserted in the Adalogger
- CTD is on and LEDs on the control unit and conductivity circuit are blinking
- Test cap is seated and seal to hand tight
- CTD has had 5 minutes to reach equilibrium at the water surface

Deployment

- CTD is lowered into the water at a speed of no more the 1 meter per second, the slower the better

Post-deployment

- CTD housing is dried and inspected for damage
- CTD is turned off and test cap is reinstalled to protect internals
- At the end of the day: SD card is removed and data is backed up to appropriate database

13 Works Cited

- Fofonoff, N.P., Millard Jr, R.C., 1983. Algorithms for the computation of fundamental properties of seawater.
- Hooker, S.K., Boyd, I.L., 2003. Salinity sensors on seals: use of marine predators to carry CTD data loggers. *Deep Sea Research Part I: Oceanographic Research Papers* 50, 927–939. [https://doi.org/10.1016/S0967-0637\(03\)00055-4](https://doi.org/10.1016/S0967-0637(03)00055-4)
- Lewis, E., 1980. The practical salinity scale 1978 and its antecedents. *IEEE Journal of Oceanic Engineering* 5, 3–8. <https://doi.org/10.1109/JOE.1980.1145448>
- Oskui, S.M., Diamante, G., Liao, C., Shi, W., Gan, J., Schlenk, D., Grover, W.H., 2016. Assessing and Reducing the Toxicity of 3D-Printed Parts. *Environ. Sci. Technol. Lett.* 3, 1–6. <https://doi.org/10.1021/acs.estlett.5b00249>
- Stammer, D., Balmaseda, M., Heimbach, P., Köhl, A., Weaver, A., 2016. Ocean Data Assimilation in Support of Climate Applications: Status and Perspectives. *Annual Review of Marine Science* 8, 491–518. <https://doi.org/10.1146/annurev-marine-122414-034113>
- Stephen, R.C., Howard, F.J., Dean, R., Susan, W., Ariel, T., Mathieu, B., Denis, G., Jianping, X., Sylvie, P., Thresher, A., Traon Pierre-Yves, L., Guillaume, M., Birgit, K., Kjell-Arne, M., Velez-Belch Pedro, J., Isabelle, A., Brian, K., Jon, T., Molly, B., Steven, J.R., 2016. Fifteen years of ocean observations with the global Argo array. *Nature Climate Change Marine Res Inst* 6. <https://doi.org/10.1038/NCLIMATE2872>
- Sverdrup, H.U., Johnson, M.W., Fleming, R.H., 1942. *The Oceans: Their Physics, Chemistry, and General Biology*. Asia Publishing House.
- Thaler, A., Parsons, E.C.M., de Vos, A., Rose, N.A., Smith, C., Fretz, D., 2019. Bot Meets Whale: Best Practices for Mitigating Negative Interactions Between Marine Mammals and MicroROVs. *Front. Mar. Sci.* 6. <https://doi.org/10.3389/fmars.2019.00506>
- Thaler, A.D., Freitag, A., Bergman, E., Fretz, D., Saleu, W., 2015. Robots as vectors for marine invasions: Best practices for minimizing transmission of invasive species via observation-class ROVs. *Tropical Conservation Science* 8.
- Thomson, R.E., Emery, W.J., 2014. *Data Analysis Methods in Physical Oceanography*. Newnes.

14 Resources and Datasheets

General Guides

Soldering is Easy: Here's How To Do It:

mightyohm.com/files/soldercomic/FullSolderComic_EN.pdf

Environmental Monitoring with Arduino: Building Simple Devices to Collect Data About the World Around Us: <https://amzn.to/2M17yUi>

Pressure Sensor

MS5803-14BA Miniature 14 bar Module:

https://cdn.sparkfun.com/datasheets/Sensors/Weather/ms5803_14ba.pdf

Temperature Sensor

DS18B20 1-Wire Digital Thermometer:

<https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

Conductivity Sensor

EZO-EC Conductivity Circuit Datasheet – https://www.atlas-scientific.com/files/datasheets/circuit/EC_EZO_Datasheet.pdf

EC K 1.0 Conductivity Probe Datasheet – https://www.atlas-scientific.com/files/datasheets/probe/EC_K_1.0_probe.pdf

Conductivity Accuracy Graph – https://www.atlas-scientific.com/files/instructions/conductivity_accuracy_graph.pdf

15 Bill of Materials

	Item	Links	Price	Price per unit
Housing	2" Schedule 40 PVC	https://amzn.to/2JZnNAm Check local hardware store	\$10.50	\$5.25
	Oatey/Cherne 270229 Mechanical Test Plug	https://amzn.to/2WLCcS7 Check local hardware store	\$8.80	\$8.80
	Hose clamp (AB-36/46mm to 70mm)	https://amzn.to/3t7G1UX Check local hardware store	\$12.49	\$2.50
	1/4" PolyPro Rope	https://amzn.to/3w7su1G Check local hardware store	\$5.19	\$0.52
OpenCTD Custom Carrier Board	Custom Carrier Board	Available from Oceanography for Everyone	\$9.37	\$9.37
	Adafruit Feather M0 Adalogger	https://www.adafruit.com/product/2796 https://amzn.to/36DrCm4	\$19.95	\$19.95
	SanDisk MicroSD Memory Card	https://amzn.to/2FTkfKW	\$6.95	\$6.95
	Adafruit DS3231 Precision RTC Breakout	https://www.adafruit.com/product/3013	\$13.95	\$13.95
	CR1220 3V Coin Cell Battery	https://amzn.to/2NYFQpf https://www.adafruit.com/product/380	\$4.75	\$0.48
	Atlas Scientific EZO Conductivity Circuit	https://www.atlas-scientific.com/product_pages/circuits/ezo_ec.html https://amzn.to/2K0Xk5z	\$67.99	\$67.99
	Short Feather Female Headers	https://www.adafruit.com/product/2940	\$1.50	\$1.50
	Short Feather Male Headers	Can be sourced from any electronics supplier https://www.adafruit.com/product/3002	\$0.50	\$0.50
	7-pin JST Header/Cable Bundle	Can be sourced from any electronics supplier https://amzn.to/35Uvh3i	\$13.99	\$1.75
	MicroUSB Cable	https://amzn.to/3CEE1p0 Can be sourced from any electronics supplier	\$6.81	\$6.81
10 kΩ Resistor (2)	https://amzn.to/3JcUFjk Can be sourced from any electronics supplier	\$6.18	\$0.12	
4.7 kΩ Resistor (1)	https://amzn.to/3JdZhpt Can be sourced from any electronics supplier	\$6.17	\$0.06	

Adalogger + RTC Featherwing	Adafruit Feather M0 Adalogger	https://www.adafruit.com/product/2796 https://amzn.to/36DrCm4	\$19.95	\$19.95
	Adafruit DS3231 Precision RTC FeatherWing	https://www.adafruit.com/product/3028 https://amzn.to/2PRHmvS	\$13.95	\$13.95
	Short Feather Female Headers	https://www.adafruit.com/product/2940	\$1.50	\$1.50
	Short Feather Male Headers	Can be sourced from any electronics supplier https://www.adafruit.com/product/3002	\$0.50	\$0.50
	Adafruit Stacking Headers	Can be sourced from any electronics supplier https://www.adafruit.com/product/2830	\$1.25	\$1.25
	MicroUSB Cable	https://amzn.to/3CEEIp0	\$6.81	\$6.81
	CR1220 3V Coin Cell Battery	Can be sourced from any electronics supplier https://amzn.to/2NYFQpf https://www.adafruit.com/product/380 https://amzn.to/2FTkfKW	\$4.75	\$0.48
	SanDisk MicroSD Memory Card	https://www.adafruit.com/product/1214	\$6.95	\$6.95
Protoboard Control Unit	Adafruit Perma-Proto Small Breadboard PCB	https://www.adafruit.com/product/1214	\$7.95	\$2.65
	Atlas Scientific EZO Conductivity Circuit	https://www.atlas-scientific.com/product_pages/circuits/ezo_ec.html https://amzn.to/2K0Xk5z	\$67.99	\$67.99
	Right Angle Female PCB Header	https://amzn.to/2ICUOKi	\$9.99	\$1.00
	Solid Core Wire	Can be sourced from any electronics supplier https://www.adafruit.com/product/290	\$2.95	\$0.12
	10 kΩ Resistor (2)	Can be sourced from any electronics supplier https://amzn.to/3JcUFjk	\$6.18	\$0.12
	4.7 kΩ Resistor (1)	Can be sourced from any electronics supplier https://amzn.to/3JdZhpt	\$6.17	\$0.06
Sensors	DS18B20 Temperature Sensor in Stainless Steel Housing (Preferred)	https://amzn.to/2YOevug https://www.atlas-scientific.com/product_pages/probes/ec_k1-0-mini.html	\$11.88	\$7.13
	(Alternative) Atlas Scientific K 1.0 Mini Conductivity Probe	https://www.atlas-scientific.com/product_pages/probes/ec_k1-0.html https://amzn.to/2K0Xk5z	\$123.99	\$123.99
	(Alternative) Atlas Scientific K 1.0 Conductivity Probe	https://www.atlas-scientific.com/product_pages/probes/ec_k1-0.html https://amzn.to/2K0Xk5z	\$157.99	\$157.99
	MS5803-14BA Pressure Sensor Breakout	https://www.sparkfun.com/products/12909 https://amzn.to/2rfyEgA	\$64.50	\$64.50

DIY Pressure Sensor	MS5803-14BA Pressure Sensor SMC	https://www.digikey.com/en/products/detail/te-connectivity-measurement-specialties/MS580314BA01-00/5277631 https://www.mouser.com/ProductDetail/Measurement-Specialties/MS580314BA01-00 https://www.tti.com/content/ttiinc/en/apps/part-detail.html?partsNumber=MS580314BA01-00	\$34.29	\$34.29
	SOIC to DIP Adapter	https://www.adafruit.com/product/1212	\$2.95	\$0.41
	10 kΩ Resistor	https://amzn.to/3JcUFjk	\$6.18	\$0.06
	0.1 uF Capacitor	Can be sourced from any electronics supplier https://amzn.to/32tMBoI Can be sourced from any electronics supplier	\$5.92	\$0.24
Power and Ballast	Lithium Ion Battery Pack 3.7V 2000 mAh	https://www.adafruit.com/product/2011 https://amzn.to/2UOLmkq	\$12.50	\$12.50
	(Internal Switch) Board-mounted switch	https://www.adafruit.com/product/805	\$0.95	\$0.95
	(External Switch) 10mm X 3mm Round Rare Earth Magnet (2)	https://amzn.to/3Ifpbri	\$5.99	\$0.40
	(External Switch) 10mm round double-sided adhesive mounting dots (2)	https://amzn.to/2OdU85B	\$8.98	\$0.02
	3-ounce fishing weights	https://amzn.to/3CI46Vy	\$12.99	\$2.60
Useful Accessories	Half-size breadboard	https://www.adafruit.com/product/64 https://amzn.to/2CmjyIB	\$5.00	\$5.00
	Screw Terminal Block Extra Long Pins for Breadboarding (3)	Can be sourced from any electronics supplier https://amzn.to/34nqB0x	\$5.99	\$0.60
	Alligator Clip	Can be sourced from any electronics supplier https://www.adafruit.com/product/3448	\$3.95	\$3.95
	Jumper Wire	Can be sourced from any electronics supplier https://www.adafruit.com/product/1956	\$1.95	\$1.95
	SD Card Reader	Can be sourced from any electronics supplier https://amzn.to/3I8mjNb Can be sourced from any electronics supplier	\$16.99	\$16.99

16 Consumables

	Item	Links	Price	Price per unit
Consumables	Polymaker Polyterra Matte PLA Filament 1.75mm	https://amzn.to/3Js8XMU	\$19.99	\$19.99
	Lead Free Solder Wire	https://amzn.to/3tSUHWY	\$6.95	\$6.95
	3M Scotch #35 Electrical Tape	https://amzn.to/2QeA9rS	\$15.76	\$3.15
	Hook-up Wire Spool Set - 22AWG Solid Core	https://www.adafruit.com/product/1311	\$15.95	\$15.95
	Hook-up Wire Spool Set - 22AWG Stranded-Core	https://www.adafruit.com/product/3111	\$19.95	\$19.95
	Multi-Colored Heat Shrink Pack	https://www.adafruit.com/product/1649	\$4.95	\$4.95
	3M Garnet Sandpaper, 9-Inch by 11-Inch	https://amzn.to/3vCfaQ9	\$5.19	\$5.19
	Loctite 219298 Gray E-90FL Hysol Epoxy	https://amzn.to/315kQ6V	\$28.43	\$28.43
	3M Static Mixing Nozzle, 38193, 50 mL	https://amzn.to/30YJB4N	\$21.99	\$1.83
	Loctite Epoxy Five Minute Instant Mix	https://amzn.to/3tuKM8q	\$11.95	\$5.96