



Construction and Operation of the

# OpenCTD

(Rev8)



[oceanographyforeveryone.com](http://oceanographyforeveryone.com)

## **Construction and Operation of the OpenCTD**

Andrew Thaler, with contributions from Russell Neches, S. Kersey Sturdivant, Jeff Branson, Andrea Schmuttermair, Brian Marx, and Allie Wilkinson.

Sixth Edition: June 2026.

Cover Art by Andrew Thaler.

Photographs by Andrew Thaler and Allie Wilkinson unless otherwise noted.

Funding to produce this document was provided by a grant from the Open Science Hardware Foundation and support from a Patreon crowdfunding campaign.



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

## Contents

Introduction.....	5
Building Skills .....	7
Project Status and Compatibility .....	8
Acknowledgements.....	8
Materials, Tools, and Consumables .....	10
OpenCTD Rev8 Parts Checklist .....	11
Prepare the Adalogger M0 .....	11
Tools and Consumables .....	13
Andrew’s Toolkit .....	13
Safety .....	16
What tools will I encounter? .....	16
What chemicals will I encounter? .....	17
General Safety.....	17
Lithium Polymer Batteries .....	17
Electronics.....	18
Adalogger M0 Software and Libraries .....	18
Required Arduino Libraries .....	19
Flash the OpenCTD software to the Adalogger.....	20
Prepare the Adalogger M0 .....	20
Prepare the Carrier Board .....	22
Prepare the Real-Time Clock.....	24
Prepare the MAX31865 PT1000 Temperature Amplifier .....	25
Assemble the Control Unit.....	27
Powering the OpenCTD.....	28
Prepare the PT1000 Temperature Probe .....	29
Prepare the Sparkfun Pressure Sensor .....	30
Prepare the Atlas Conductivity Probe.....	30
Test All Components .....	32
Housing.....	34
3D-printing.....	34
Assemble the PowerSquid .....	35

Cutting the PVC Housing .....	37
Using the router jig .....	37
If you don't have a router table.....	41
Building the Sensor Package .....	43
Pot the Pressure Sensor.....	43
Pot the OpenCTD.....	44
Calibration and Data Management .....	46
Pressure (Depth).....	46
Temperature .....	47
Conductivity (Salinity).....	48
Data Management .....	54
Final Preparation and First Deployment.....	55
Casting Loop.....	55
Weighted Casting Line .....	56
Internal Ballast .....	57
External Ballast.....	58
The First Deployment .....	58
Stewardship.....	60
Works Cited .....	61
Resources and Datasheets .....	62
Bill of Materials .....	63
Appendix A. Alternate Pressure Sensor.....	70
Components for Board Mounted Pressure Sensor .....	70
DIY Pressure Sensor .....	71

## Introduction

**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.

There is something radical about making and using your own scientific instruments. The structure of scientific inquiry has coalesced around a model that is both expensive and exclusive. This centralizes knowledge production within a circle of individuals, organizations, and institutions that rarely reflect 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 this 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 inquiry. A researcher who can repair their own equipment is not dependent on the ever-changing winds of academic sentiment to decide what is and is not worthy of study. A community leader who has the tools to generate their own data does not have to wait for institutions to prioritize 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 that preclude the participation of ocean stakeholders further erodes our potential to understand, anticipate, and adapt to those changes.

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, knowledge seekers can unlock ocean patterns hidden beneath the sea's surface. To reveal these patterns, oceanographers use a CTD—an instrument that measures conductivity (used to calculate salinity), temperature, and absolute pressure (used to determine water depth; Thomson and Emery, 2014).

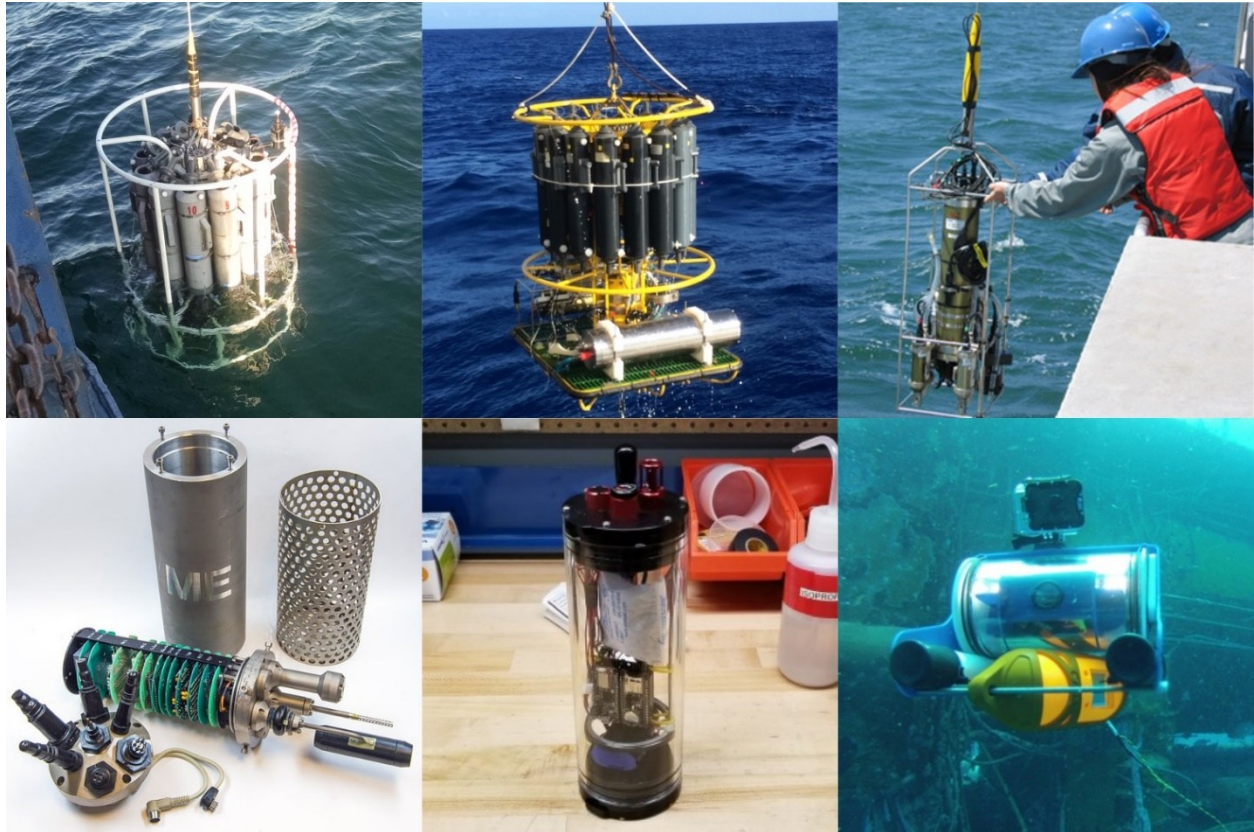


Figure 1. A selection of various CTDs used for field work. Top left: A CTD and rosette deployed from a NOAA boat (WHOI); top center: A CTD and rosette deployed by IFREMER; top right: a student deploys a CTD (MIT); bottom left: an 80s era ME GRISARD CTD; bottom center: the CTDizzle open-source CTD by 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 even the most affordable models costing several thousand dollars. Many contemporary instrumentation companies now use a subscription model, locking data analysis and support behind a paywall, adding yet another cost. 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 create 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 demand 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 (Thaler et al., 2024). 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.

The OpenCTD is built by the end-user, providing both access to the tools of oceanography as well as the skills to maintain, calibrate, repair, and replace OpenCTDs. For scientists working in remote settings, where commercial maintenance 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, hands-on STEM experience for advanced students, building an OpenCTD offers an introduction to coding, 3D-printing, hardware prototyping, and electronics. Construction of an OpenCTD provides a practical foundation for courses in oceanography and marine or environmental science.

## Skill Building

Though building an OpenCTD requires a variety of skills, you don't need any background in soldering, electronics, or coding. Constructing an OpenCTD provides the foundation for developing these hardware and software skills. With a few hours of practice, you can learn the techniques necessary to build and operate your own OpenCTD.

For absolute beginners, I 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](https://mightyohm.com/files/soldercomic/FullSolderComic_EN.pdf)

Arduino is a coding environment built on C/C++ and designed for flexibility and ease of use, but it can 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. 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>

## Project Status and Compatibility

OpenCTD is stable (as of 6/21/2023) on the ATSAM21G18 ARM Cortex M0 processor. Construction workflow, as well as pinouts, bills of material, and operation are designed around the Adafruit Adalogger M0. Adjustments may need to be made to adapt to other M0-based microcontrollers.

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 were designed around the Qduino Mini. Adjustments may need to be made to adapt to other ATMEGA32U4-based microcontrollers. This version is no longer supported, but is archived here: <https://github.com/OceanographyforEveryone/OpenCTD/tree/main/Retired>.

## Acknowledgements.

From its inception, the OpenCTD and Oceanography for Everyone has benefitted from the support and expertise of an incredible community. Dr. Kim Martini provided exceptional guidance throughout the development process. Dr. Harold Tay provided 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. J. Jacob Levenson has provided tremendous technical support and encouragement throughout the life of this project.

Hundreds of individuals have contributed to making the OpenCTD program a success. The authors would like to acknowledge Ian Black, Jeff Branson, David Lang, Eric Stackpole, Walt Holm, the OpenROV/Sofar Ocean Team, Miriam Goldstein, Alex Deghan and the Conservation X Labs Team, Cassie Stymiest of Educational Passages, the naturalists from Bar Harbor Whale Watch Company, Coastal Research & Education Society of Long Island, Boston Harbor City Cruises, Hyannis Whale Watch Cruises, Blue Ocean Society for Marine Conservation, and the college of the Atlantic, Henry Reiske and Katie Gavenus of the Center for Alaskan Coastal Studies, the students and teachers of Chapman and Homer Flex High School in Homer, Alaska, Tammy Silva and David Wiley of the Stellwagen Bank National Marine Sanctuary, Brian Marx,

Andrea Schmuttermair, Jamison Smith, Lisa Robbins, Ruth Stilwell, Allie Wilkinson, David Shiffman, and Amy Freitag.

Seed funding for the OpenCTD project was provided via a crowdfunding campaign to ADT and SKS via RocketHub. Sustaining funding for the OpenCTD was provided via an ongoing crowdfunding campaign to ADT via Patreon. The SEAoTech project was supported by a contract from the Bureau of Ocean Energy Management (140M0121D0004). Community Oceanography While Watching Whales was supported via a grant to ADT from the National Marine Sanctuaries Foundation and the Bureau of Ocean Energy Management (M17PG00019/P00002). A grant to improve the documentation for the project was provided by the Open Source Hardware Association to ADT.

## Materials, Tools, and Consumables

The approach to designing the OpenCTD focuses on accessibility—both accessibility of the components and the relative ease of construction. The 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. The number of non-standard parts are kept to a minimum and, when unavoidable, sourced from manufacturers with a long history of consistent quality who will continue to manufacture and support these components into the foreseeable future.

The OpenCTD can be assembled entirely from components available from large online electronics vendors such as Adafruit, Sparkfun, DigiKey, Amazon, or AliExpress as well as local hardware stores, with the exception of the OpenCTD Carrier Board. Many components can also be purchased directly through their supplier or sourced through auction sites at lower cost. Quality across suppliers for most components is 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. Pressure sensor breakout boards are available from a variety of manufacturers and instructions for building your own pressure sensor from principal components is provided in Appendix A.

While a Bill of Materials with recommended parts and their suppliers is provided, these are not necessarily the least expensive options and prices can vary significantly depending on local availability and the 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 the OpenCTD team through [oceanographyforeveryone.com](http://oceanographyforeveryone.com).

When available, an Amazon Affiliate link is provided to purchase components. The OpenCTD project gets a small kickback when you buy a component through those links, which helps offset the cost of development and testing for the OpenCTD. This is, however, not an endorsement of Amazon and users are encouraged to seek out local suppliers and small businesses wherever possible.

## OpenCTD Rev8 Parts Checklist

Here you will find everything you need to build an OpenCTD. We've broken this checklist out into each section of the manual and list each component where it appears in the build process for the first time. We have intentionally put the Bill of Materials at the end of the manual to encourage you to read the whole document before buying parts. OpenCTD Rev8 kits are available through our online store at <https://oceanographyforeveryone.bigcartel.com/>.

### Prepare the Adalogger M0

- Adalogger M0 (with male headers)
- MicroSD Card
- USB (micro) cable

### Prepare the Carrier Board

- OpenROV Rev8 Carrier Board
- Female headers for Adalogger M0 (12-pin and 16-pin)
- Female headers for DS3231 RTC (8-pin)
- Male headers for Electrically Isolated EZO Carrier Board (5-pin)
- 2-position switch
- Female headers for MAX31865 PT1000 temperature amplifier (8-pin)
- 10 kOhm resistor (2)
- 4-pin JST header

### Prepare the Real-Time Clock

- DS3231 Real-Time Clock (with male headers)
- CR1220 coin cell battery

### Prepare the MAX31865 PT1000 Temperature Amplifier

- MAX31865 PT1000 temperature amplifier (with male headers)
- 3-pin JST SM connector (male)

### Assemble the Control Unit

- 10mm standoff (3)
- 2.5mm nylon screws (6)
- Atlas Ezo Conductivity Circuit
- Electrically Isolated EZO Carrier Board (with metal standoffs and screws)
- 5-pin female-to-female jumper wire

### Powering the OpenCTD

- 3.7v 400mAh lithium-ion battery

### Prepare the PT1000 Temperature Probe

- PT1000 temperature probe
- 3-pin JST SM connector (female)
- Solder seal wire connectors (3)

### **Prepare the Sparkfun Pressure Sensor**

- Sparkfun Pressure Sensor Breakout Board
- 20-cm long 22-gauge stranded wire (4)
- 4-pin JST connector
- Solder seal wire connectors (4)

### **Prepare the Atlas Conductivity Sensor**

- Atlas K1.0 Conductivity Probe

### **3D-printing (these are the parts you will 3D print yourself)**

- Sensor Baseplate
- Electronic Chassis Bottom
- Electronic Chassis Top
- Electronic Chassis Cap
- Router Jig 1
- Router Jig 2
- Test Cap Tightening Handle

### **Cutting the PVC Housing**

- 12" schedule 40 PVC Pipe

### **Pot the Pressure Sensor**

- 2-ton instant epoxy

### **Potting the OpenCTD**

- E/A120-HP 2-part epoxy
- Long Taper Tip Nozzle with Bayonet Connection for Two-Part Cartridge

### **Final Preparation and First Deployment**

- 2" hose clamp (2)
- Short length of Polypro rope (2)
- 2" plumbers test cap

### **Calibration**

- K 1.0 Calibration Standard

## Tools and Consumables

A basic soldering kit is sufficient for assembly. Several components 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, heat gun, precision screwdrivers.

**Consumables for general electronics work:** lead-free solder, electrical tape.

**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, Epoxy gun for 50ml cartridges.

**Consumables for housing and adhesion:** sandpaper.

**Tools for Arduino programming:** Computer capable of running Arduino IDE.

### A note on Epoxies and Glues:

**We recommend Loctite Hysol E-120HP.** Epoxies with similar characteristics may be acceptable replacements. We have tested several different epoxies and glues to ensure a generally wide variety of options for sourcing. For smaller applications, such as potting the pressure sensor, we use a standard 5-minute marine epoxy.

Hysol EA-90FL and EA E-120HP have been extensively tested in OpenCTDs as well as in 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 high viscosity, should be avoided.

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

## Andrew’s Toolkit

Sometimes it is nice to have a list of tried and tested tools rather sourcing 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 hundred of OpenCTDs at workshops around the world.

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. The PINECIL is the best soldering iron for most applications):

- **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:** PINECIL – Smart Mini Portable Soldering Iron (<https://pine64.com/product/pinecil-smart-mini-portable-soldering-iron/>). Requires a high wattage USB power supply like this one: Anker Nano USB C Charger (<https://amzn.to/4oIBm7d>).
- **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>)

**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>)
- **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.*

**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>)
- **Milwaukee M12 12-Volt Cordless PVC Shear:** (<https://amzn.to/44eDxFG>) If you're planning on making several dozen OpenCTDs, this pipe shear, while expensive, will save you significant time and reduce PVC waste.

- **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:

- **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.

### Consumables:

- **PLA Filament:** PolyTerra™ PLA (<https://us.polymaker.com/products/polyterra-pla>)
- **Lead-free solder:** Lead Free Solder Wire (<https://amzn.to/4oCo63K>)
- **Electrical tape:** 3M Scotch #35 Electrical Tape (<https://amzn.to/2QeA9rS>)
- **22-gauge silicone wire:** 22 Gauge Silicone Wire Kit (<https://amzn.to/3CN0F0z>)
- **Sandpaper:** 3M Garnet Sandpaper, 9-Inch by 11-Inch, Assorted-Grit (<https://amzn.to/3vCfaQ9>)
- **2-part adhesive epoxy:** Loctite E-120FL Hysol Epoxy (<https://amzn.to/315kQ6V>)
- **Five-minute epoxy:** Loctite Epoxy Five Minute Instant Mix (<https://amzn.to/3tuKM8q>)

## Safety

---

*“There is no more important rule than to wear safety glasses.”*

*Norm Abrams*

---

There are only a few potential hazards while working on the OpenCTD. Soldering involves both a hot soldering iron and fumes produced from burning solder. Potting the sensors involves the use of a 2-part chemical epoxy which can bond with skin. Trimming wires may produce small metal projectiles which can damage the eye. Working near water presents a drowning risk. All of these hazards can be mitigated with basic safety protocols and proper protective equipment.

### What tools will I encounter?

Construction of the OpenCTD requires only basic hand tools, as well as electronic tools associated with soldering. If you are constructing a PVC housing, you might also use power tools, including a miter saw, a jig saw, a drill, a PVC cutter, or a router. For workshops with primary school students, we recommend that the instructor prepare the PVC housing in advance, as this process is messy, requires the use of power tools, and adds additional time to the build process.

Whenever you are working with tools, you should wear safety glasses to protect your eyes. When working with power tools, what you aren't wearing is as important as what you are wearing. Avoid gloves or neckties, as well as anything with long cords, especially hoodies with drawstrings. These cords can become trapped in the tool and pull the user towards a spinning blade.

When soldering, be aware that the iron is extremely hot. Contact with a soldering iron will result in immediate 2<sup>nd</sup> degree burns. Only hold the soldering iron by the insulated grip and stow it in a secure soldering iron holder. Never leave a hot iron unattended. Always turn off and unplug the iron when not in use. Whenever hot irons are in use, other participants should take extra care not to jostle the table or the person holding the iron and avoid any distractions. Safety glasses should be worn when soldering.

The heat gun used for melting heat shrink tubing can also get very hot. Be aware of its location and never touch the exposed metal portion.

Soldering produces fumes. We recommend using a lead-free solder to dramatically reduce the presence of harmful fumes. In a well-ventilated environment using lead-free solder, the fumes produced from soldering should pose no health hazard. However, for added safety, a small fume extractor can be used to draw the fumes away from your person.

When trimming pins on the control board or other components, the snipped ends tend to fly off. This can be avoided by placing a finger over the end about to be snipped to prevent it from launching into the air. Some high-end flush cutters have a small magnet to prevent fly aways. When trimming pins, wear safety glasses and be aware of who is around you.

### What chemicals will I encounter?

The OpenCTD is potted using two different kinds of 2-stage epoxy. The epoxy used to seal the pressure sensor into the baseplate is an off-the-shelf 2-ton epoxy commonly found in hardware stores. The high-strength epoxy used to seal the sensors into the housing is a specialized epoxy available from most major adhesives distributors. Both epoxies are safe when handled properly. Always wear safety gloves and safety glasses when handling epoxy. Read the warning labels on the packaging before use. If epoxy gets on your skin or clothes, immediately flush with warm water.

### General Safety

You will use sharp tools during the build process. Always be aware of any blades or cutting implements and store them appropriately. Pliers and flush cutters can pinch and break the skin. Always treat your tools with respect.

When working on the water, wear a US Coast Guard (or other relevant agency) approved life jacket.

### Lithium Polymer Batteries

The lithium polymer pouch batteries used in the OpenCTD are generally safe and stable when stored and handled appropriately. In the event that a battery is punctured by a tool or inadvertently heated to extreme temperatures by a soldering iron, remove the battery from the classroom immediately and place outside on a concrete surface or fire-resistant container. The battery cannot be recovered and must be disposed of appropriately.

## Electronics

The control unit is the brain of the OpenCTD. At its core is an Adalogger M0, an 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 M0 is connected to a conductivity circuit and temperature amplifier through the custom carrier board.

Before beginning, 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/>.

### Adalogger M0 Software and Libraries

The Adafruit M0 Adalogger acts as both the main computer for the OpenCTD and writes data to the onboard SD Card drive. Before you begin soldering hardware, flash the software to the Adalogger M0 to confirm that the microcontroller is functioning and your Arduino IDE is set-up correctly. Ensuring that the microcontroller works with the OpenCTD firmware first will dramatically simplify any troubleshooting, later.

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, and 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.

All OpenCTD software and documentation are hosted on GitHub. GitHub can seem daunting to new users. 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 Apple Computers, 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](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. This may take a while.
5. Restart Arduino IDE.
6. Install the required libraries listed below (*Sketch > Include Library > Add .ZIP Library*) and update Arduino IDE as needed.
7. Restart Arduino IDE.

## Required Arduino Libraries

The following Arduino libraries are required for the OpenCTD Rev8 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.

**Adafruit\_MAX31865.h.** Allows you to communicate with the PT1000 temperature amplifier and probe.

**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/main/Software/Support>

## Flash the OpenCTD software to the Adalogger

Install the OpenCTD software on the Adafruit Adalogger M0. On some Adaloggers, Adafruit ships the microcontroller preloaded with both Arduino and Circuit Python, with Circuit Python being the default. When you plug your Adalogger in for the first time, it will open into the Circuit Python boot drive. You can ignore this. After you flash the OpenCTD software, the Adalogger will default to Arduino.

1. Restart Arduino IDE. You should see several new Adafruit options in the *Board* menu. The Adalogger is an Adafruit Feather M0 with an embedded SD card reader.
2. Select “Adafruit Feather M0” from the *Board* menu.
3. Connect your Adalogger to your laptop using a micro-USB cable. The necessary drivers should automatically install.
4. Open the `OpenCTD_M0_R8.ino` source code in Arduino IDE.
5. Under *Tools > Port*, select the COM port that reads “Adafruit Feather M0” (it should autodetect, but doesn’t always).
6. Click Upload (either the right pointing arrow on the main interface or under *Sketch > Upload*). The code will compile and upload to the Adalogger.

Once the OpenCTD firmware is flashed to the Adalogger M0, use the serial monitor in Arduino IDE to verify that it is working correctly. With the Adalogger connected to your computer via USB, 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.

The serial monitor should say “Card failed, or not present”. Insert an SD card. The Adalogger will initialize and then attempt to read the Real-Time Clock and sensors, which will fail. Unplug the control unit, remove the SD card, and confirm via SD card reader that an empty file was created on the SD card.

## Prepare the Adalogger M0

The Adalogger M0 needs headers. Headers are the male or female pin assemblies that allow you to connect and disconnect electronic components from a PCB. Headers make come pre-cut or in long strips that need to be cut to length to align with the correct pins on the carrier board. 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.

When placing headers for soldering, the short pins go through the holes in the printed circuit board and are soldered on the opposite side.



Figure 2. Adalogger M0 without headers. Left: topside of the board. Right: bottom of the board.

1. Slot 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 (16 on one side, 12 on the other). The long header pins should point down. If you are using generic headers, cut them to length with flush cutters so that the number of posts matches the number of pins on each side.
2. Confirm that the headers are straight and perpendicular to the face of the Adalogger.
3. On the top of the Adalogger, solder the four corner pins, double check that the headers are straight, and then solder the remaining pins.
4. Using the flush cutters to clean up any protruding solder joints.
5. Insert a microSD card into the SD card reader.

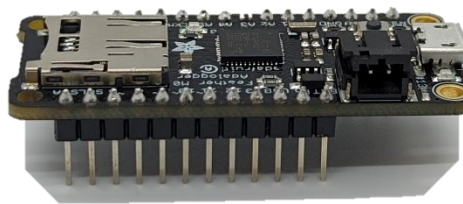


Figure 3: Adafruit M0 Adalogger with male headers (SD card not shown).

## Prepare the Carrier Board

The carrier board allows for rapid and accurate assembly of the OpenCTD control unit. Your carrier board should be labeled OpenCTD rev8. The color of the board may vary. Some early boards may be labeled “OpenCTD 8x”.

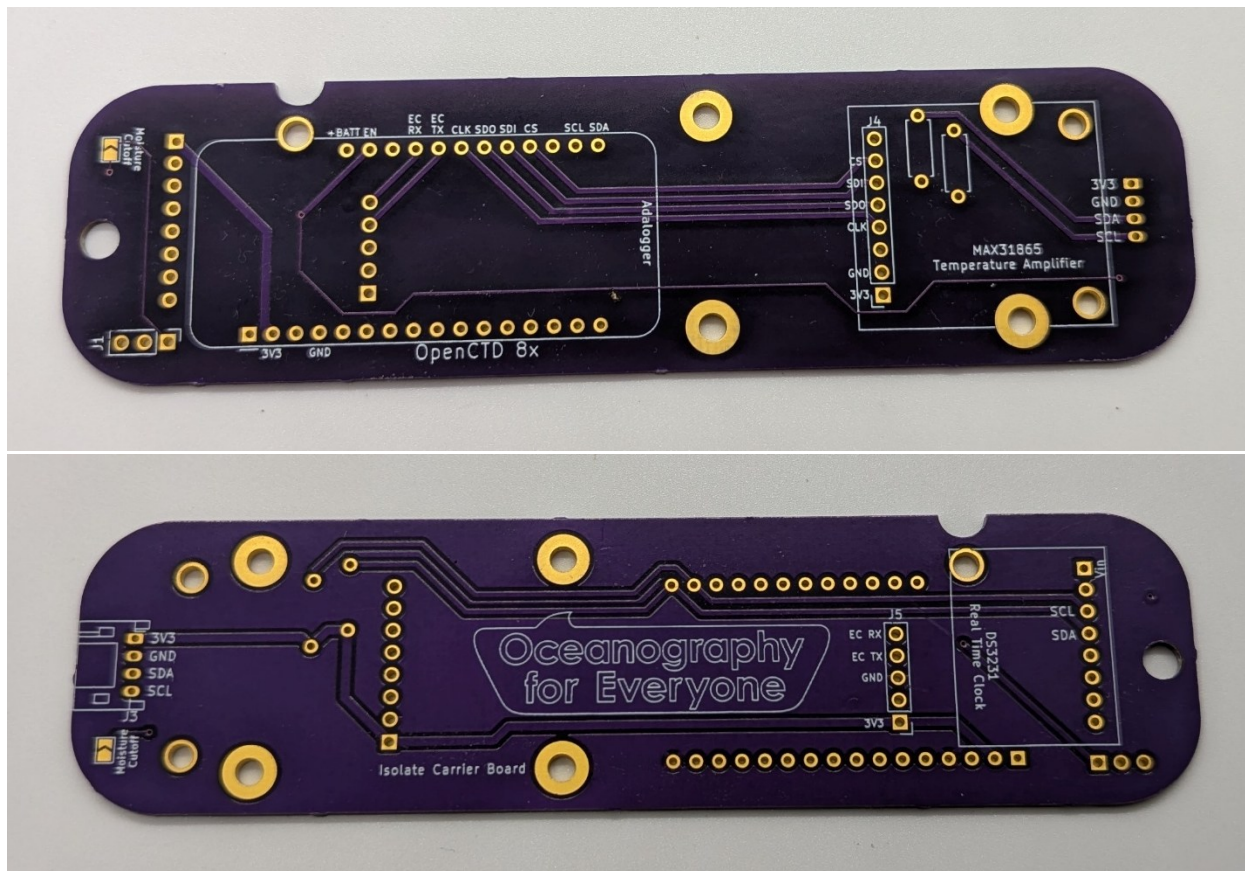


Figure 4: OpenCTD Carrier Board. Topside of the board above, underside of board below.

1. Slot female headers on the topside of the carrier board in the pins in the box labeled Adalogger. 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.
2. Solder female headers for the Adalogger M0 to the carrier board, starting with the four corners to hold them in place.
3. Slot female headers on the underside of the carrier board in in the box labeled DS3231 and solder them into place.
4. Slot male headers for the Atlas Electrically Isolated EZO Carrier Board on the underside of the carrier board in the box labeled J5 and solder them into place.

5. Slot the 2-position switch on the topside of the carrier board in the box labeled J1 and solder them into place. The pins will be extra long and need to be trimmed.
6. Slot female headers for the MAX31865 temperature amplifier on the topside of the carrier board in the box labeled J4 and solder them into place.
7. Solder 2 10kOhm resistors into the slots on the topside of the carrier board reserved for resistors inside the box labeled MAX31865. The pins will be extra long and need to be trimmed. The resistors are bidirectional and orientation does not matter.
8. Slot the 4 pin JST connectors on the topside of the carrier board in the box labeled J3 and solder into place.
- 9. Optional Step: To use the two moisture sensors embedded in the control board, lightly solder to small blobs onto each of the four pads labeled Moisture Cutoff, making sure the two adjacent pads do not connect. This will cause the device to power down if the pads are short-circuited by seawater, hopefully preserving the electronics in even of a housing failure.**
10. Using the flush cutters, snip down any protruding pins so that they sit flush with the solder joint.

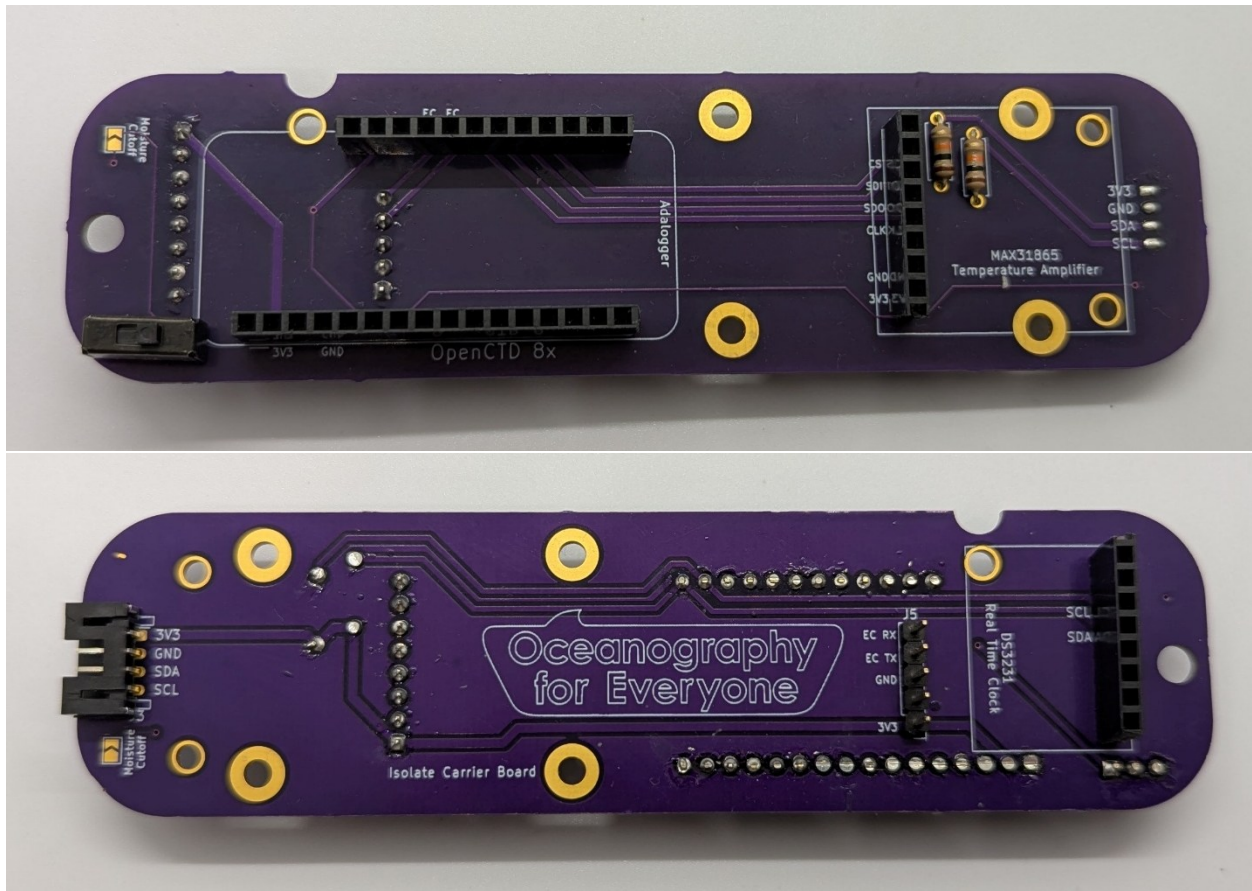


Figure 5: Fully populated OpenCTD control board. Topside of the board above, underside of board below

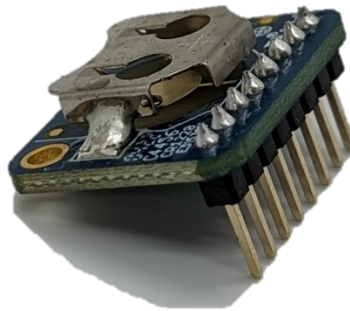
## Prepare the Real-Time Clock

The Real-Time Clock (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 use the program RTC\_Reset in the OpenCTD GitHub repository under Support to reset the clock to the system time on a connected PC. **Make sure you have the orientation of the pins correct before soldering or you're going to have a bad time!**



Figure 6. Real-Time Clock module without headers. Left: chip side of the board. Right: battery side of the board.

1. Slot the male header pins into the RTC module on the chip side of the board such that the battery is facing up and the header pins extend from the opposite side of the module. Do not solder with the battery installed in the RTC.
2. Solder the male header pins into the RTC module.
3. Using the flush cutters, snip down any protruding pins so that they sit flush with the solder joint.
4. Insert the CR1220 battery into the RTC, positive side facing out.



*Figure 7. Real-Time Clock module with male headers soldered to the chip side of the board.*

## Prepare the MAX31865 PT1000 Temperature Amplifier

The PT1000 resistance temperature detector (RTD) amplifier is designed to read the low resistance of a platinum temperature resistor. This MAX31865 amplifier can read 2-, 3-, or 4-wire sensors. For the OpenCTD, we use a 3-wire platinum temperature sensor. You will have to set the amplifier to 3-wire by soldering two pads and breaking one jump pad, as described in the Adafruit instructions for 3-wire sensors here: <https://learn.adafruit.com/adafruit-max31865-rtd-pt100-amplifier/rtd-wiring-config> or in steps 1 – 3, below. **Please note, the temperature amplifier will be mounted upside-down on the control board, and the header pins will point up from the same side as the screw-block terminals. Make sure you have the orientation of the pins correct before soldering or you're going to have a bad time!**

1. Locate the pads labeled **2/3 Wire** and **24 3** on your MAX31865 amplifier.
2. Using a utility knife, break the small copper trace connecting the rectangular pads labeled 2 and 4. By carefully applying a small blob of solder, connect the middle rectangular pad labeled 4 to the rectangular pad labeled 3.

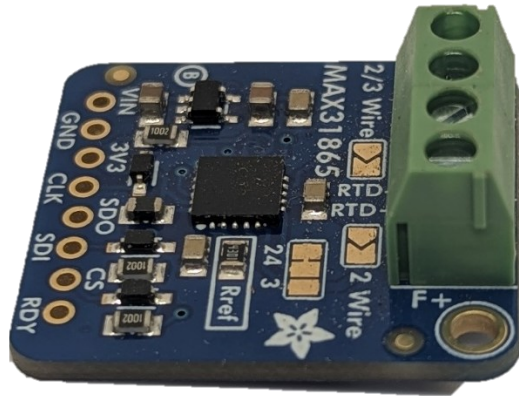


Figure 8. MAX31865 amplifier for PT1000 resistance temperature detector (RTD), as it arrives from the manufacturer.

3. By carefully applying a small blob of solder, connect the triangular pad labeled 2/3 wire to the chevron-shaped pad, also labeled 2/3 Wire. By creating these connections, you are telling the amplifier to look for 3 wires coming from the temperature probe.
4. Solder the male headers so that the long pins are pointing up, on the same side as the large screw terminals.

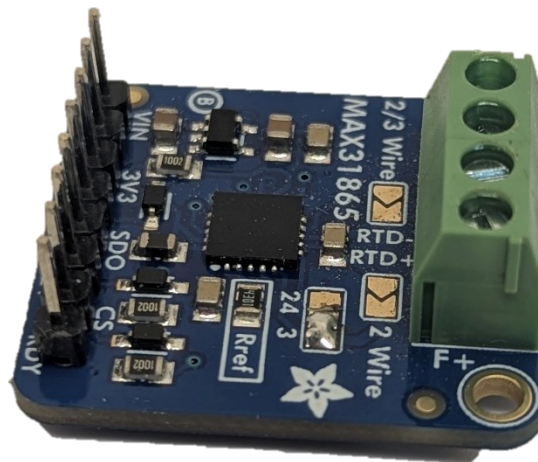


Figure 9. MAX31865 amplifier with pads 2 and 4 connected and the male headers installed. Note: pads labeled 2/3 Wire are not connected in this image.

5. Insert the male side of the 3-pin connector cable into the screw terminals labeled F+, RTD+, and RTD-. Tighten the screw terminals so that the bare wires make contact with the metal pad inside the screw terminals and do not pull out with a light tug. If necessary, you can shorten the wires on the 3-pin connector cable.



*Figure 10. Completed MAX31865 amplifier with 3-pin connector cable installed.*

## Assemble the Control Unit

You will not need to solder anything during this section.

1. Slot the RTC into the RTC headers on the underside of the carrier board and ensure that the RTC board is parallel with the carrier board. Secure the RTC to the board using the 10mm standoff and 2.5mm nylon screws.
2. Insert the Atlas Ezo Conductivity Circuit into the headers on the Electrically Isolated EZO Carrier Board. The pins labeled PRB and VCC should face towards the BNC connector. Using the metal standoffs provided, mount the EZO Carrier Board to the OpenCTD Control Board upside down, so that the EZO conductivity circuit is sandwiched between the two boards. You may need to use the flush cutters to trim down any sharp pins which might press against the EZO Conductivity Circuit.
3. Use the included 5-pin female-to-female jumper wire to connect the headers on the Electrically Isolated EZO Carrier Board to the headers labeled J5 on the OpenCTD Carrier Board, making sure that you match the header labels for each pin.
4. Slot 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.

5. Slot the MAX31865 amplifier into the MAX31865 amplifier headers on the topside of the carrier board and ensure that the MAX31865 amplifier board is parallel with the carrier board. Secure the MAX31865 amplifier to the board using 2 10mm standoffs and 2.5mm nylon screws. This board should also be upside down.

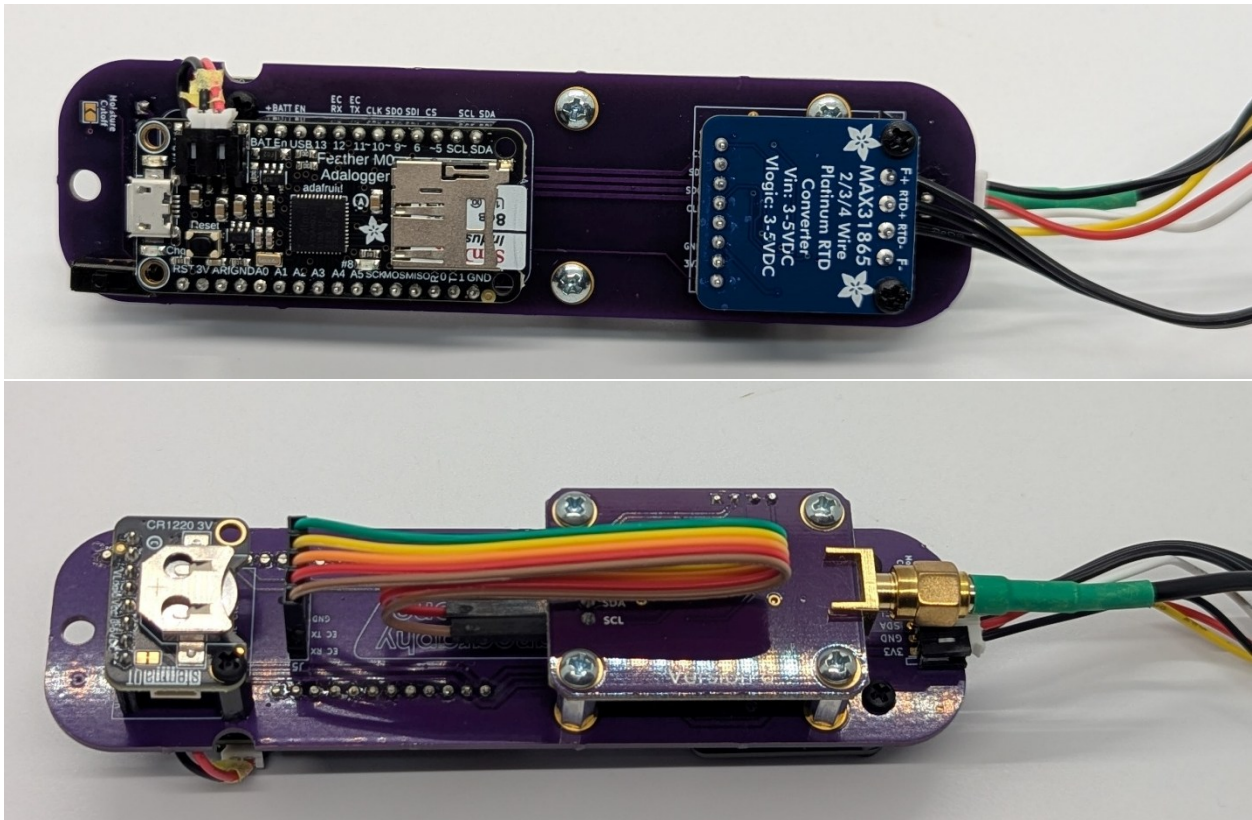


Figure 11. Topside and underside of completed OpenCTD Rev8 Control Unit.

## Powering the OpenCTD

The Adalogger has an internal 100mA charging unit, which means a standard 400mAh battery will take 4 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 400mAh battery down from 4 hours to less than 1 hour.

1. Plug in the battery and ensure that the switch functions correctly.
2. The 3.7v 400mAh is ideal for the OpenCTD. It fits snugly between the carrier board and the Adalogger M0 and lasts for roughly 18 hours of continuous use. If you need more

power, larger batteries that use a JST connector can be used. Make sure that the positive and negative leads on batteries from other supplier correspond to the positive and negative terminals on the Adalogger M0. There is a small notch in the carrier board that allows a larger battery to be mounted to the underside of the carrier board or anywhere else within the housing.

**Note:** Depending on the amperage of the USB power supply that you're connected to, there may not be enough power to drive the EZO EC chip from USB power. You may also need to plug the battery into the 2-pin battery JST connector.

If the EZO conductivity circuit blinks red, it is not receiving enough power.

### Prepare the PT1000 Temperature Probe

The PT1000 temperature probe connects to the 3-pin connector that you installed on the MAX31865 amplifier.

1. Remove the spade terminals and strip  $\frac{1}{4}$  cm off the end of each of the three wires.
2. Solder the female end of the 3-pin lead to the exposed wires, either using solder shrink tubing or solder and heat shrink. Make sure that the two red wires correspond to the pin labeled F+ and RTD+, and the clear wire aligns with the pin labeled RDT-.
3. Plug the temperature probe into the MAX31865 amplifier.

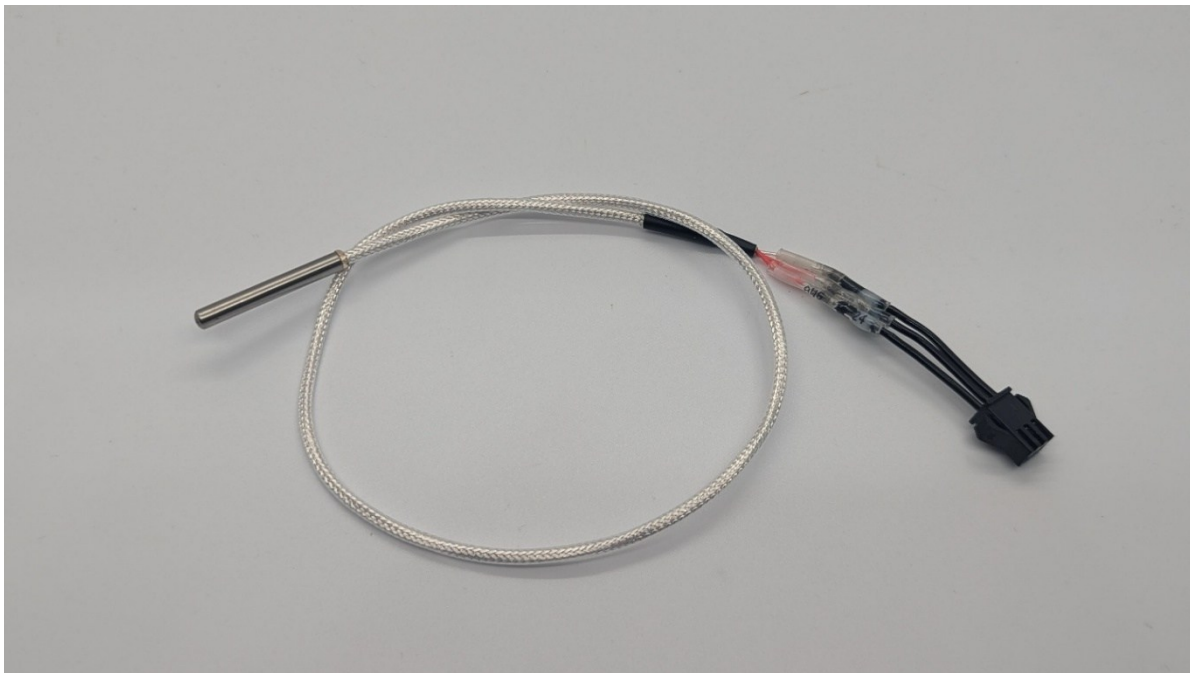


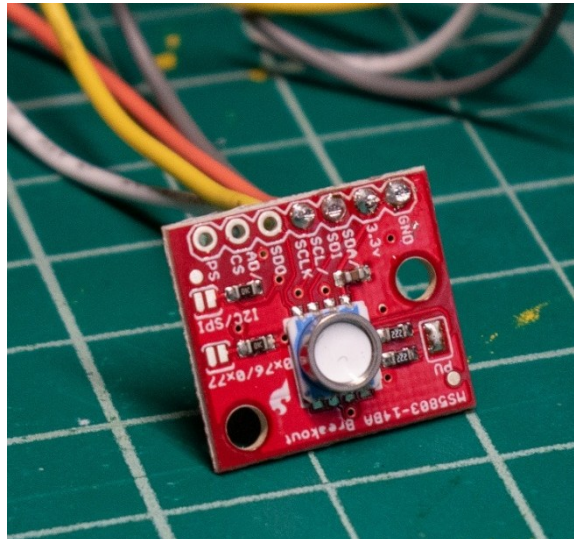
Figure 12. PT1000 Temperature Probe.

## Prepare the Sparkfun Pressure Sensor

**Please note: An alternative to the Sparkfun Pressure Sensor Breakout Board is available from Oceanography for Everyone, however, it requires surface mounted soldering and may be difficult for inexperienced builders. See Appendix A.**

For the Sparkfun Pressure Sensor Breakout Board:

1. Strip ½ cm off the end of 4 different colored 20-cm lengths of 22-gauge stranded wire. We recommend red for VCC; black for GND; yellow for SCL; and white for SDA.
2. Solder those wires to the GND, 3.3v (or VCC), SDA, and SCL pins on the back of the Sparkfun Pressure Sensor Breakout Board. Keep track of which wire goes to which pin. Note that the wires must extend from the side opposite the pressure sensor.



*Figure 13. Sparkfun Pressure Sensor Breakout with soldered pins.*

3. Solder the female end of the 4-pin JST connector to the exposed wires, either using solder shrink tubing or solder and heat shrink. Make sure that the wires from the breakout board correspond to the labels on the carrier board.
4. Connect the 4-wire JST connector to the 4-wire JST headers on the carrier board.

## Prepare the Atlas Conductivity Probe

The Atlas Scientific conductivity probe comes in both long and short varieties, with paired leads that terminate at a large BNC connector. You will not need to cut or modify this probe in any way.

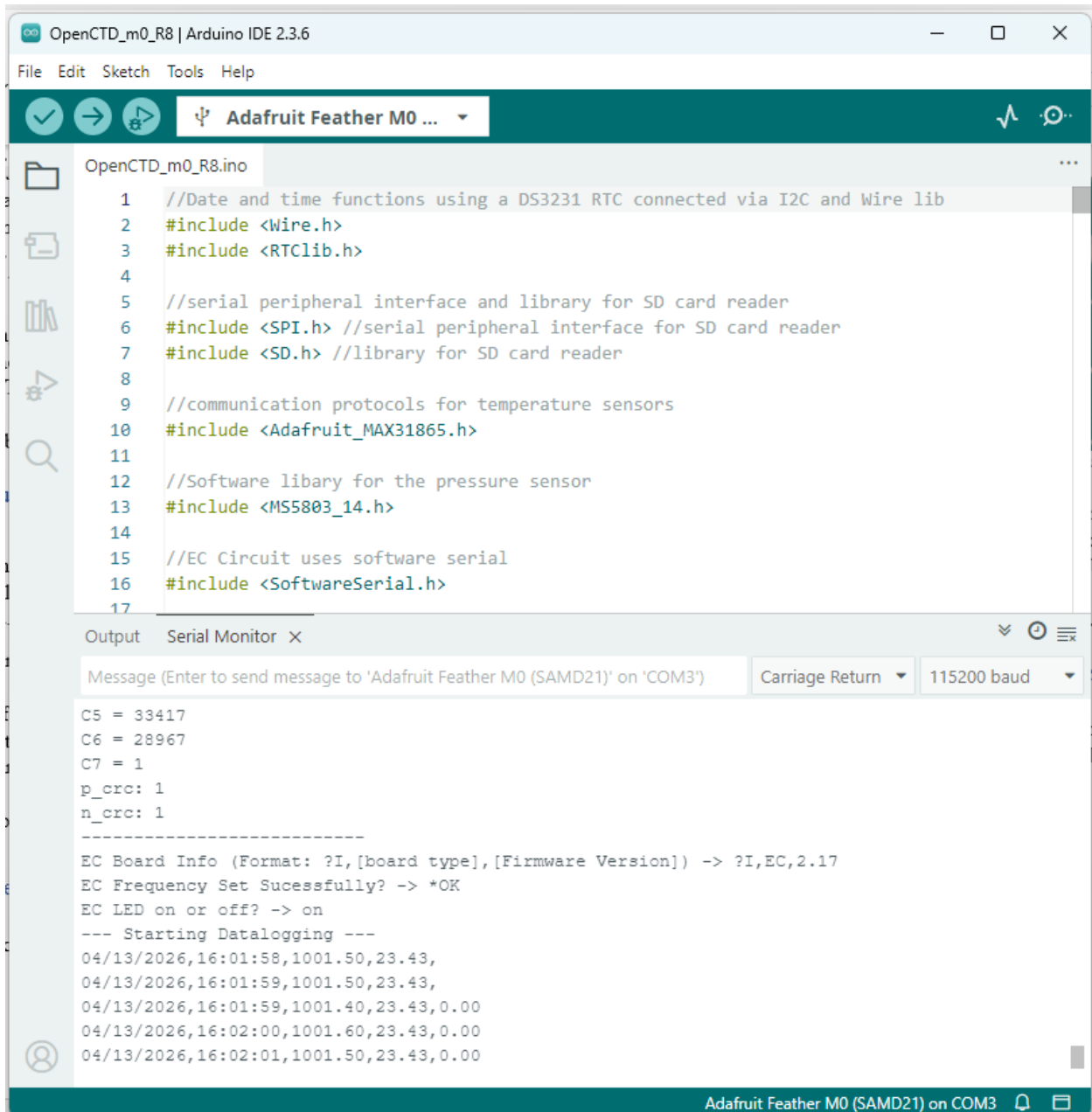
While you can cut this wire to size for ease of use, Atlas Scientific advises customers to test their conductivity probe before cutting any wire, as cutting the wire will void the warranty. Atlas also provides a much more detailed step-by-step guide to cleanly and accurately cutting the conductivity sensor wire here: <https://files.atlas-scientific.com/how-to-properly-cut-probe-cables.pdf>

If you do not have access to a coax cable stripper tool, you can carefully strip these cable using your flush cutters. It helps to cut them long and use the extra tail for practice before cutting to final length. Atlas also advise to clean any soldered leads in isopropyl alcohol to remove any flux residue which may cause resistance in the system.

1. Plug the BNC connector into the BNC threaded header on the Electrically Isolated EZO Carrier Board

## Test All Components

With all the components connected to your control board, it's time to test your soldering skills. Connect your control board to your computer using a USB cable, open Arduino IDE, and turn on the OpenCTD. Select the correct board and open the serial monitor. You should see a data stream that looks like this:



The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar shows a dropdown menu for the board, currently set to 'Adafruit Feather M0 ...'. The main editor displays the sketch 'OpenCTD\_m0\_R8.ino' with the following code:

```
1 //Date and time functions using a DS3231 RTC connected via I2C and Wire lib
2 #include <Wire.h>
3 #include <RTClib.h>
4
5 //serial peripheral interface and library for SD card reader
6 #include <SPI.h> //serial peripheral interface for SD card reader
7 #include <SD.h> //library for SD card reader
8
9 //communication protocols for temperature sensors
10 #include <Adafruit_MAX31865.h>
11
12 //Software library for the pressure sensor
13 #include <MS5803_14.h>
14
15 //EC Circuit uses software serial
16 #include <SoftwareSerial.h>
17
```

The Serial Monitor is open, showing the following output:

```
Message (Enter to send message to 'Adafruit Feather M0 (SAMD21)' on 'COM3') Carriage Return 115200 baud
C5 = 33417
C6 = 28967
C7 = 1
p_crc: 1
n_crc: 1
-----
EC Board Info (Format: ?I,[board type],[Firmware Version]) -> ?I,EC,2.17
EC Frequency Set Successfully? -> *OK
EC LED on or off? -> on
--- Starting Datalogging ---
04/13/2026,16:01:58,1001.50,23.43,
04/13/2026,16:01:59,1001.50,23.43,
04/13/2026,16:01:59,1001.40,23.43,0.00
04/13/2026,16:02:00,1001.60,23.43,0.00
04/13/2026,16:02:01,1001.50,23.43,0.00
```

The status bar at the bottom indicates 'Adafruit Feather M0 (SAMD21) on COM3'.

Figure 14. Arduino IDE and Serial Monitor readout of functioning OpenCTD.

After the bootup and diagnostic information, you should see a continuously updating line of numbers. These values are, from left to right, the date (in MM/DD/YYYY format), The time (in HH:MM:SS), the pressure (mbar), temperature (°C), and conductivity (mS). Outside of the water, at sea level, in a comfortable room, Pressure should read around 1012 mbar, temperature around 21°C, and conductivity at 0 mS.

Congratulations! If everything looks normal, you're CTD is working! If not, double check all your solder connections, make sure the wires have good contact, and check that nothing is reversed.

Now all you have to do is make you OpenCTD waterproof!

## Housing

### 3D-printing

It may seem counterintuitive to rely on a technology like 3D-printing in designing a low-cost and accessible oceanographic instrument but 3D-printing has matured to the point where the most affordable and widely available printers are more than sufficient for the production of these parts. This technology dramatically reduces the cost for manufacturing small and bespoke plastic components. 3D-printing plays a central role in the conservation technology movement.

The main chassis of the OpenCTD, which protects the control unit, and the sensor baseplate, which aligns the sensors and creates a bulkhead for the sensor passthrough, are 3D-printable objects. While it is possible to construct a fully functional OpenCTD without these parts, they are designed to make building and operating the device simpler and more intuitive.

- The **Sensor Baseplate** provides a foundation and template for arranging the sensors prior to potting in epoxy.
- The **Electronics Chassis** holds the Adalogger and other electronic components, allowing users to easily access the SD card while minimizing strain on the wiring and protecting and isolating the battery. The chassis is printed in three parts which fit snugly together and are held in place with electrical tape.
- The **Router Jig** is a set of jigs designed to fit inside a clean PVC pipe that allow you to cut 3 clean, consistent notches on the bottom of the CTD to protect the sensors while allowing water to flow through.
- The **Cap Wrench** is a small wrench that fits over the wingnut of the Plumber's Test Cap, making it easier to tighten and loosen.

All parts are designed to be printed in PLA at 200-micron resolution with 50% to 100% infill. 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 be harmful to the marine environment. ABS, in contrast, has been shown to cause developmental abnormalities in fish exposed to ABS particulates (Oskui et al., 2016). Other materials, like PETG and Nylon may be preferable but require more experience and a more advanced printer.

Andrew's personal preference is for Polymaker Panchroma PLA, which has been optimized for strength and compostability.

- Panchroma™ PLA (<https://shop.polymaker.com/>)

All 3D printer files are available in the OpenCTD GitHub repository:

<https://github.com/OceanographyforEveryone/OpenCTD/tree/main/Hardware/3DPrints>

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

The electronics chassis can be printed without supports. Test fit the two halves of the main body to ensure that they slot together firmly. Test fit the cap of the electronics chassis to ensure that it fits snugly.



*Figure 15: 3D-Printed chassis for OpenCTD.*

## Assemble the PowerSquid

PowerSquid is our name for the OpenCTD Rev8 control unit once it is housed in its protective chassis. The chassis comes in three parts, two long body pieces and one cap. This chassis will protect the control unit from drips, bumps, and knocks, but not from full submersion in seawater. A small notch in the cap allows you to access the power switch to turn the OpenCTD on and off.

1. Reconnect all cables leading to the sensor package.

2. Fold the extra wiring from the Atlas EZO Carrier Board and place the control unit in the bottom chassis piece so that the excess cable rest in the long, rectangular slot. The sensors should exit from the end with the large oval hole.
3. Place the top chassis piece on top of the control unit, making sure that the edges align. Use electrical tape or rubber bands to secure the two pieces.
4. Place the cap on the open end of the chassis assembly. It should fit snugly and remain in place.
5. Turn on the OpenCTD and confirm that it is reading all sensors.

## Cutting the PVC 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. The sensor end will be sealed with high shear-strength epoxy, allowing 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: 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.**)

## Using the router jig

The router jigs are designed to be used with a c-clamp and small bench router. We do not recommend attempting to route the housing notches freehand. We recommend using a ¼” compression cut template bit.

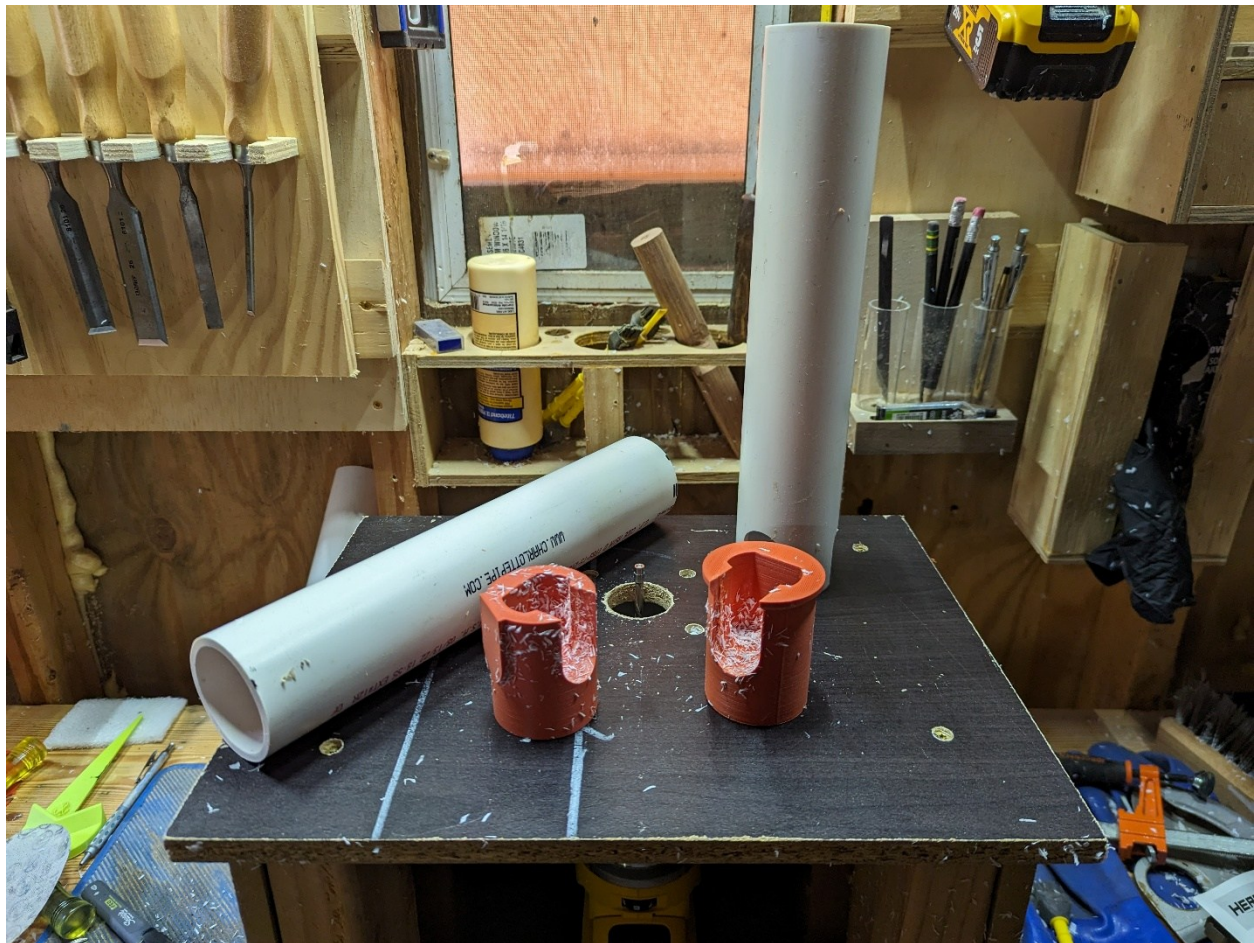


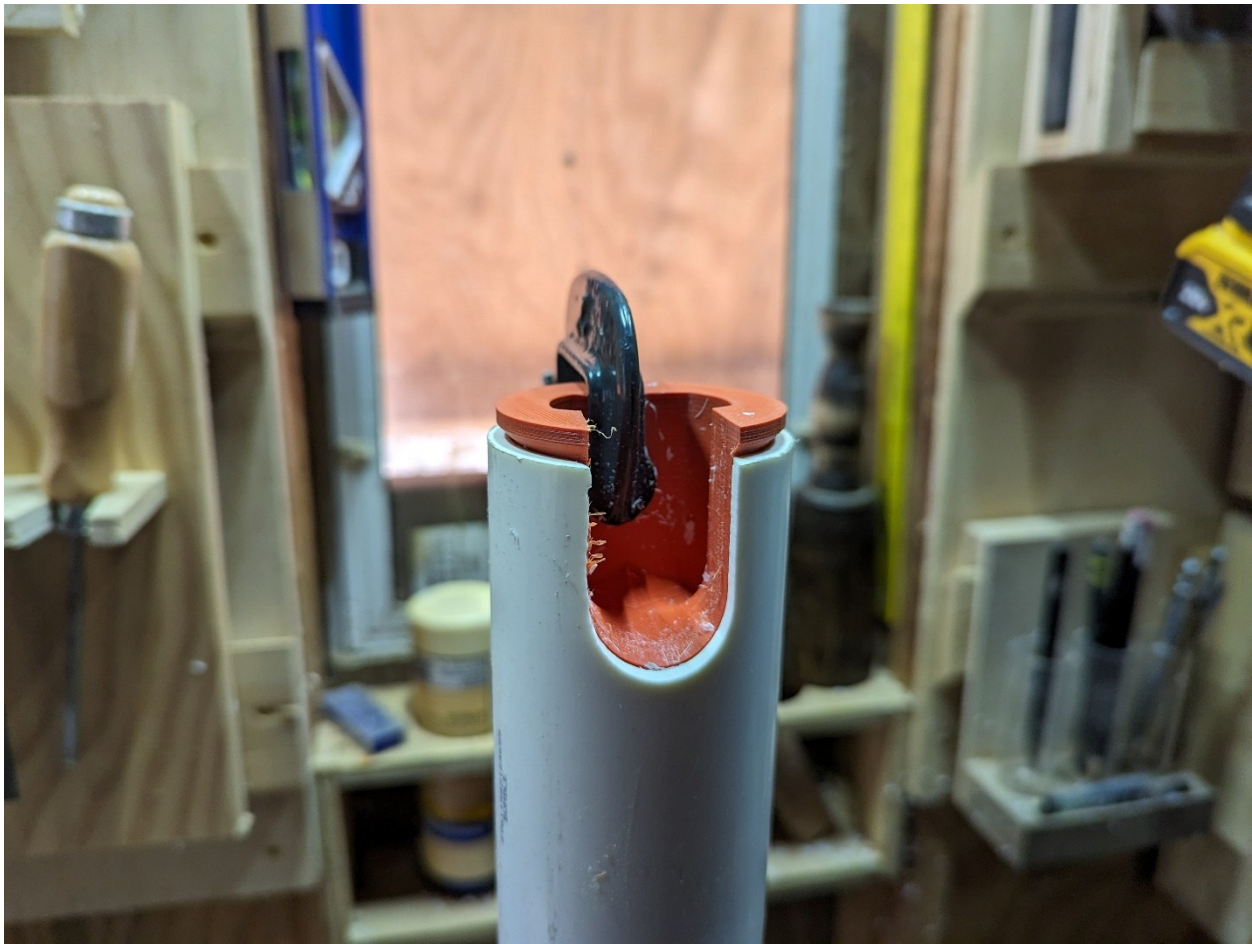
Figure 16. Bench router with PVC pipes and router jig.

1. Cut a 12” section of PVC pipe. Deburr the edges around each cut with a knife, sandpaper, or deburring tool.
2. Insert the router jig labeled RouterJigForHousing1.3mf into one end of the clean and evenly cut PVC pipe. Secure with a c-clamp. Make sure that c-clamp doesn't come in contact with the router bit during routing.



*Figure 17. Router jig inserted into PVC pipe and secured with c-clamp.*

3. Route the first of three notches in the PCV pipe, going slowly. Make sure that the router bit does not contact the c-clamp.



*Figure 18. The first notch routed into the PCV pipe.*

4. Clean up your work surface. Remove the first router jig and insert the jig labelled RouterJigForHousing2.3mf into the base of the PVC pipe such that the notch projection slides into the notch that you just routed. Secure with a c-clamp and route the notch, taking care to go slowly and not let the router bit contact the c-clamp.



*Figure 19. The second router jig inserted into the PVC pipe and secured with a c-clamp.*

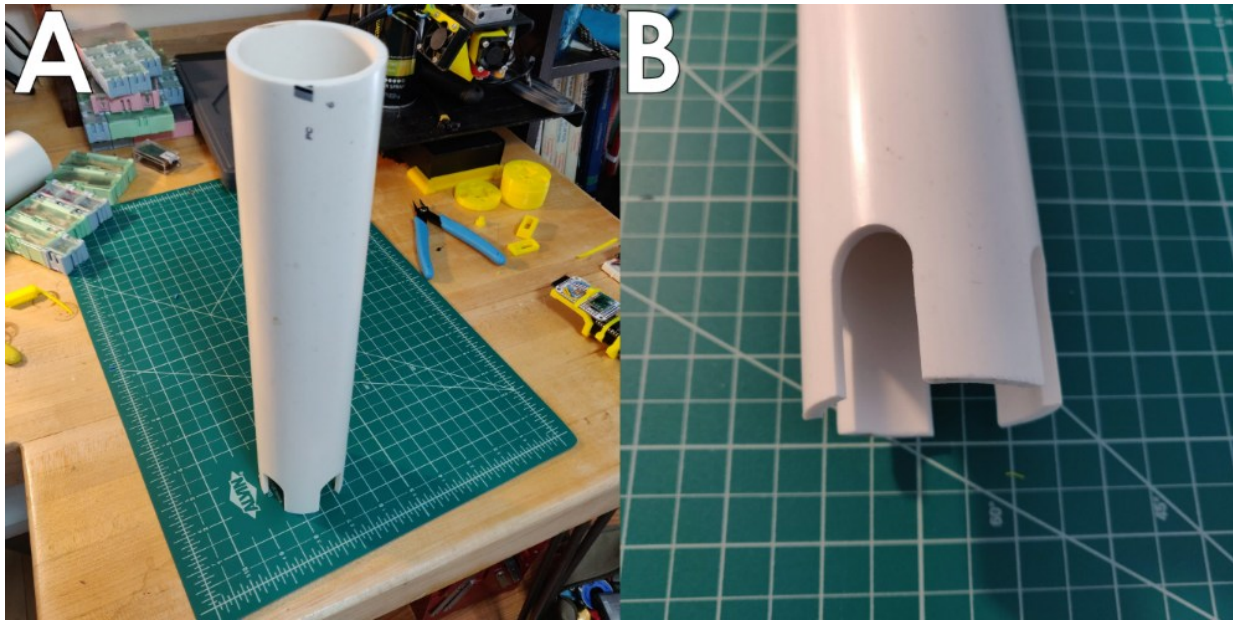
5. Route the third notch using the same jig with the notch projection slotted into the second notch. Secure with a c-clamp and route the notch, taking care to go slowly and not let the route bit contact the c-clamp.
6. Remove the jig and clean up your workspace. You should now have an OpenCTD housing with three clean notches routed into the base. With a vacuum or blower, collect any stray pieces of PVC and dispose of safely.
7. 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 20. Completed PVC housing made on a router table.*

### If you don't have a router table

If you don't have access to a router table, you can cut notches using a drill with a 1" bit and a jigsaw or handsaw. It won't look as pretty, but it will work just fine. Depending on how you intend to use your OpenCTD, you may not need notches at all, or you can simply drill holes in the base to allow water flow without cutting full notches.



*Figure 21. 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. This 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.

## Building the Sensor Package

It's time to create a waterproof pass through for all of your OpenCTD probes. To build the sensor package, first disconnect the temperature, pressure, and conductivity sensors from the carrier board.

### Pot 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 2-ton instant epoxy to protect the circuits before the full sensor assembly is sealed. 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. **Do not apply too much pressure to the sensor housing or the metal ring that contains the gel cap can become detached.** If this happens, the sensor cannot be repaired. 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 force it in, it should be a snug but comfortable fit that will hold in the epoxy applied in step 2.
2. Once the sensor is correctly seated, apply 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.

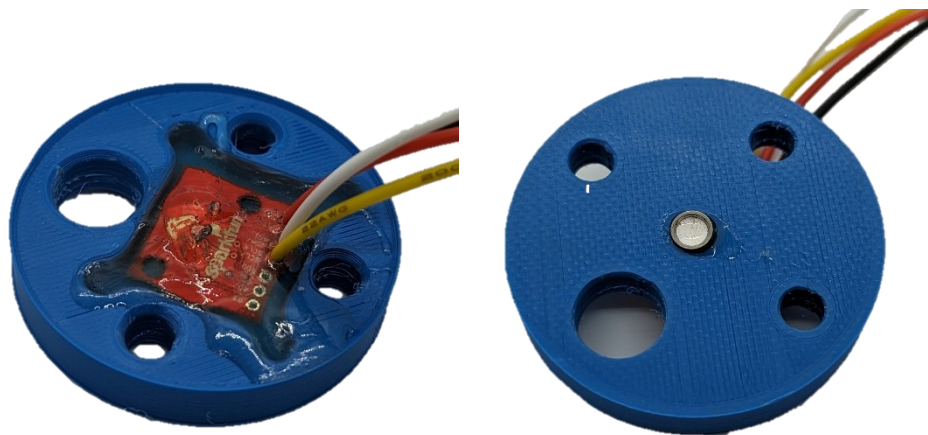


Figure 22: SparkFun MS5803-14BA Pressure Sensor Breakout potted in Epoxy. Topside view on right, underside on left. Note, image from Rev7 OpenCTD, baseplate for Rev8 OpenCTD only has 3 through-holes.

4. Let the assembly sit for a 10 to 20 minutes to allow the epoxy to set.
5. Once the epoxy has set, insert the PT1000 temperature probe into the smaller of the two holes such that 1.5 to 2 cm extends out of the bottom of the baseplate.
6. 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.

### Pot 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 E/A120-HP epoxy cures. Within the first 20 minutes, you may gently lift the housing vertically to check that no epoxy is leaking out the bottom. Do not submerge the OpenCTD 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. Reconnect all the sensors to the carrier board and double check that all three probes are still reading.
4. 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.**

5. Using the 50mL cartridge gun and venturi mixing nozzle, inject the full 50mL of 2-part epoxy into the OpenCTD housing.
6. 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 at the surface.



*Figure 23. A clear OpenCTD showing the thick epoxy potting layer.*

## Calibration and Data Management

One of the most significant challenges with any open-source science hardware is calibration. Commercial CTDs often require expensive service contracts in order to maintain data quality. The OpenCTD is designed to ensure that the end user is capable of calibrating and maintaining their own equipment, independent of any manufacturer.

While comprehensive validation of the OpenCTD requires access to professionally calibrated instruments maintained in a controlled environment, it is not necessary to do a full validation test for every instrument. This level of validation is for those for whom a high degree of confidence and extreme precision are needed. For environmental monitoring, conservation, ecology, management, and most other use cases, an OpenCTD calibrated against salinity standards and general-purpose temperature sensors is adequate.

As the community grows, validated OpenCTDs can be used to benchmark newly built instruments.

A Microsoft Excel and Google Sheet template is available in the OpenCTD GitHub repository. This template will take raw data from the OpenCTD and make the necessary conversions to create human-readable water column profiles.

### Pressure (Depth)

The OpenCTD uses the MS5803 14-bar pressure sensitive chip from Measurement Specialties designed for SCUBA dive watches and depth gauges. This chip contains a pressure sensitive resistor embedded in a gel matrix. 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

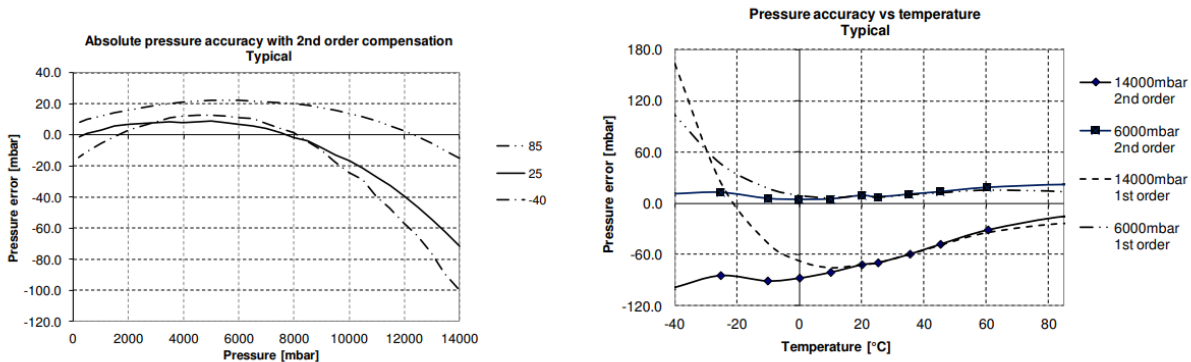


Figure 24. Pressure sensor error versus ambient pressure and temperature for the MS5803-14BA pressure sensor. Provided by manufacturer.

The MS5803 is a factory-calibrated pressure sensor capable of outputting absolute pressure up to 0.2 millibar resolution. No additional calibration is needed by the user. The pressure sensor outputs absolute pressure. At sea level, the average standard atmospheric pressure is 1013.25 mbar, though local weather and small changes in altitude will affect this baseline. To convert pressure in millibars to depth in meters, use the following equation:

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

Where **D** is depth in meters; **P<sub>(DEPTH)</sub>** is the pressure (in millibars) at depth); **P<sub>(SURFACE)</sub>** 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/s<sup>2</sup> (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, 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<sub>(SURFACE)</sub>**. For deployments in alpine lakes and other high-altitude environments, be aware that surface pressure may be significantly less than 1013.25 mbar.

## Temperature

The OpenCTD Rev8 uses a platinum PT1000 temperature sensor potted in a stainless-steel cladding. The sensor has an operating range of -50 to 280°C. It is a Class B temperature sensor designed for industrial use. The advertised accuracy is ±0.3°C, however, in field trials, we have seen accuracy consistently as low as ±0.1°C.

Thermal time constant is a measurement 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 these PT1000 temperature sensors at less than 1 second, though it varies depending on the length of exposed stainless steel cladding.

The PT1000 temperature sensor and MAX31865 is self-calibrating based on the resistance between the two positive leads and the internal resistance on the amplifier. Calibration can drift as the instrument ages. We recommend checking temperature calibration at the beginning of each expedition or once per year. When calibration drifts, a linear offset can generally be applied to correct for changes.

## 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 25. Accuracy of Atlas Scientific K 1.0 Conductivity Probe and EZO circuit. Data provided by Atlas Scientific.

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 Scientific sells a set of calibration standards for their probes. Most calibration standards are designed to be used at 25°C but the Atlas EZO-EC conductivity circuit also allows for temperature compensation if maintaining a stable 25°C temperature is not possible. While any clean glass or plastic container can be used to hold the conductivity solution standards, we recommend a small probe storage bottle to contain the fluid and minimize waste.

Do not return the conductivity solution to the original bottle once it has been used.

In order to clean the conductivity probe and whatever containers you are using to hold the conductivity solution between steps in the calibration process, you will need a source of either distilled water or water filtered through reverse osmosis. Distilled water is readily available in

most grocery stores. Reverse osmosis is often used to purify drinking water from seawater and RO machines are often found on oceangoing vessels as well as ground wells. Aquafina brand bottled water is produced using reverse osmosis and has no added minerals, making this brand a cheap source of RO water.

We have developed a simple system for establishing a controlled, thermally stable environment for the calibration process that used the hot bed of a low-cost 3D printer, coupled with a Styrofoam cooler, and a large ceramic thermal mass. A few hours before calibration, set the heated bed of the 3D printer to 25°C.

Find the largest thermal mass available that the OpenCTD housing can fit inside (a heavy coffee mug works well) and place it on the heated bed of the printer. If you cannot maintain a thermostable environment, you can create a cheap isolation chamber by cutting a hole in the bottom of a Styrofoam cooler large enough for the OpenCTD housing to fit snugly inside. Place the inverted cooler on top of the 3D printer's heated bed such that the OpenCTD can be inserted through the hole and sit inside the thermal mass.

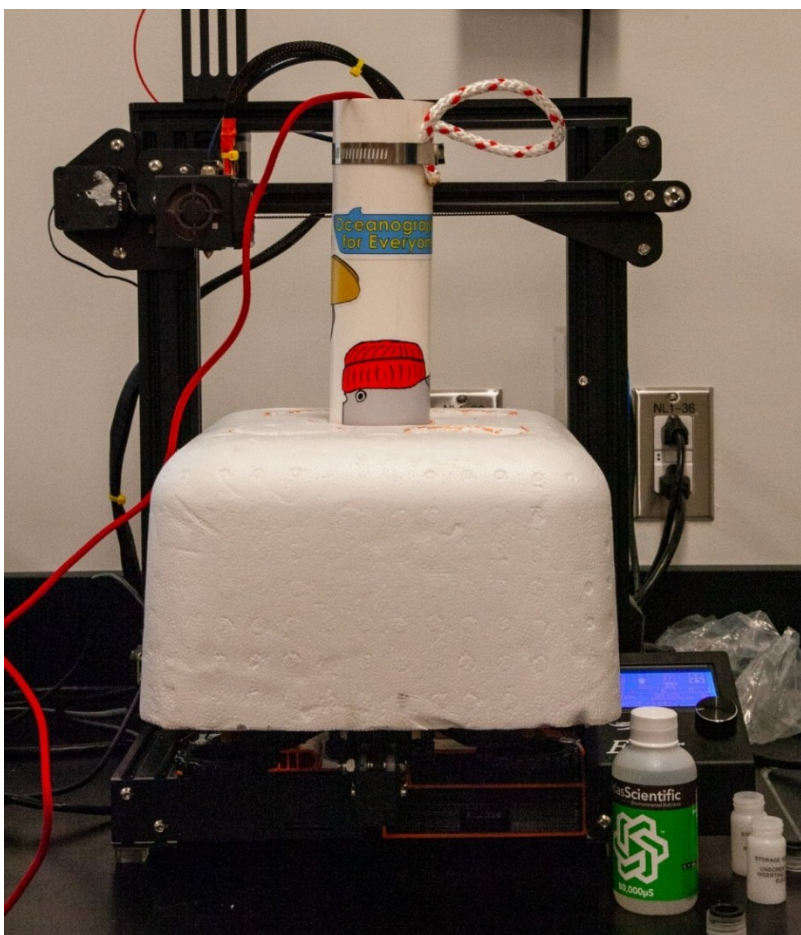


Figure 26. OpenCTD mounted inside a Styrofoam cooler on top of a 3D printer heat bed in order to maintain an ambient 25°C temperature. Photo by Allie Wilkinson.

When you are ready to calibrate your OpenCTD, insert it into a coffee mug or other thermal mass, and allow it to reach equilibrium.

Two-point calibration is facilitated using `Serial_for_EC_Calibration_m0.ino` ([https://github.com/OceanographyforEveryone/OpenCTD/tree/main/Software/Support/Serial\\_for\\_EC\\_Calibration\\_m0](https://github.com/OceanographyforEveryone/OpenCTD/tree/main/Software/Support/Serial_for_EC_Calibration_m0)) found in the OpenCTD GitHub repository. If only a single calibration standard is available, one-point calibration is possible, but not recommended. You can monitor the temperature probe through the calibration software.

1. With the control unit powered on and connected to the sensors, upload `Serial_for_EC_Calibration_m0.ino` to the Adalogger M0. Make sure that the battery is on, as the voltage differential between USB and battery power may affect calibration.
2. Open the serial monitor through the Arduino IDE. Make sure the dropdown menus in the serial monitor are set to “Carriage return” and “9600 baud”. You should see a message with EC Board Info, a note to review this guide, and a reference list of useful commands.
3. Type `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. If the K-value is something other than 1.0 type `k,x` (where x is the new K-value of the probe) in the command line and hit enter. This is useful if your EZO-EC conductivity circuit reads the incorrect K-value or if you are building an OpenCTD for freshwater deployments using a K 0.1 probe. If there is an error transmitting the command, the serial monitor will report `*ER` and you should send the command again.
4. To check whether the probe has been previously calibrated type `cal,?` in the command line and hit enter. If it reads `?CAL,0` the probe has not been calibrated.
5. Type `cal,clear` and hit enter to clear the previous calibration. It is good practice to send this command even if the circuit reports back that it has not been calibrated. If there is an error transmitting the command, the serial monitor will report `*ER` and you should send the command again.
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. If there is an error transmitting the command, the serial monitor will report `*ER` and you should send the command again.

7. Enable continuous monitoring by entering **c,1**. You will begin to see a steady stream of readings from the conductivity probe updating once per second. If the dry calibration was successful, those values should read 0.00.
8. Clean the probe with distilled or RO water and dry with a clean paper towel. Make sure there is no water trapped in the small hole between electrodes. Clean and dry the container used to hold calibration solution with distilled or RO water and dry with a clean paper towel.
9. Fill the container used to hold calibration solution (we recommend using a beaker or standard ceramic coffee mug) with the less conductive of the two conductivity solutions (12,880  $\mu\text{S}$  if you're using the Atlas standard solutions). Tap the probe gently to shake out any bubbles.
10. Place the submerged probe in a thermally stable environment. Leave the probe suspended in the calibration solution until readings stabilize (15 minutes is usually sufficient). The reading should be anywhere from 1% to 40% off from the calibration standard. Even once they stabilize, there will be small fluctuations in value. If you leave the probe in for too long, it will ionize the water and the conductivity readings will begin to creep up. A gentle shake will return the solution to its baseline.
11. Type **cal,low,y** (where y is the known value of the standard. If you are using the Atlas conductivity standard, type **cal,low,12880**) in the command line and hit enter. The conductivity readings in the serial monitor will not update.
12. Remove the probe from the solution, clean the probe with distilled or RO water, and dry with a clean paper towel. Make sure there is no water trapped in the small hole between electrodes.
13. Fill the container used to hold calibration solution with the more conductive of the two conductivity solutions (80,000  $\mu\text{S}$  if you're using the Atlas standard solutions). Tap the probe gently to shake out any bubbles.
14. Place the submerged probe in a thermally stable environment. Leave the probe suspended in the calibration solution until readings stabilize (15 minutes is usually sufficient). The reading should be anywhere from 1% to 40% off from the calibration standard. Even once they stabilize, there will be small fluctuations in value. If you leave the probe in for too long, it will ionize the water and the conductivity readings will begin to creep up. A gentle shake will return the solution to its baseline.

- Type **cal,high,y** (where y is the known value of the standard. If you are using the Atlas conductivity standard, type **cal,high,80000**) in the command line and hit enter. The conductivity readings in the serial monitor will update and should read very close to 80,000.

**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.

- After calibration is complete, upload the OpenCTD software to the Adalogger M0.

If you do not have access to a system that allows you to maintain a temperature at a stable 25°C, there are two alternatives for calibration. You can use the temperature compensation feature of the Atlas EZO-EC conductivity circuit to set a different stable temperature for calibration, provided that the environment around the probe is stable at that temperature. You can derive the compensation temperature from the OpenCTDs temperature probes, rounded up to the nearest tenth of a °C. Set the compensation temperature on the EZO-EC by sending the command **t,x** through the serial monitor, where **x** is the compensation temperature rounded to the nearest tenth of a °C. Once temperature compensation is entered, continue through the above two-point calibration protocol. Temperature compensation will reset to 25°C whenever the unit is powered down.

Review the Atlas EZO-EC Datasheet ([https://atlas-scientific.com/files/EC\\_EZO\\_Datasheet.pdf](https://atlas-scientific.com/files/EC_EZO_Datasheet.pdf)) for more thorough instructions.

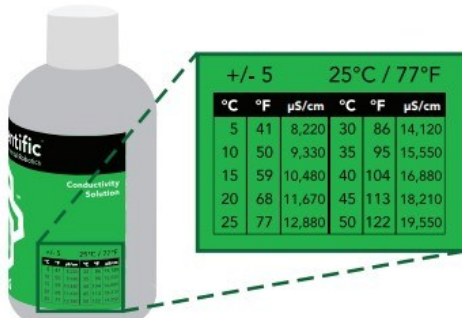


Figure 27. Location of conductivity temperature offsets. Image from Atlas Scientific.

If you cannot maintain a consistent thermal environment, there is one potential alternative that will result in a rough calibration of the probe acceptable for education or for applications where precise salinity measurements are not essential. On the side of bottles of salinity standards provided by Atlas Scientific and others, there is usually a table of compensation values in intervals of 5°C. You can calibrate to the value closest to ambient temperature. This method should only be used if no other alternatives are available.

Conductivity should not need to be calibrated again once the calibration protocol is complete, however it is practice 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](https://atlas-scientific.com/files/EC_EZO_Datasheet.pdf)).

Each control unit is calibrated to a specific sensor package. Control units are not interchangeable between OpenCTDs without recalibration.

## Data Management

The OpenCTD outputs data as a comma 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 and generates a basic water column profile.

The template 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 OpenCTD 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. Data is accessed post-deployment by removing the SD card from the Control Unit and connecting it directly to a computer.

**Importing into Excel:** Open the Excel or Google Sheets template and navigate to the Data Input sheet in the lower left corner. You can copy-paste the desired data points directly into the sheet. If you use the Get Data function, Excel will create a new sheet for the data.

**Data clean-up:** Depending on when you started and stopped the OpenCTD, you may have dozens to thousands of rows of data from the surface that you don't need. Using the reading from the pressure sensor and conductivity probe, remove any data that appears to be from the surface (pressure readings around 1010 and conductivity readings at or near 0).

**Water column profile:** Select the data that you want to analyze and copy/paste it into the "Paste RAW data from OpenCTD" columns on the Calculations sheet. The water column profile graph should update but you will have to edit the source data fields in order to cover all of your data. You can save the water column profile as an image to embed in presentations or download the cleaned and processed data for more analysis. For long deployments, the final column included the data averaged in 1-minute intervals.

## Final Preparation and First Deployment

Precisely how much weight a CTD needs for ballast and how it should be mounted to an instrument or 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.

### 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 28. 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. 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. We recommend installing a second casting loop at the bottom of the OpenCTD housing from which you can connect weights below the device.

### 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 stays in position.



*Figure 29. Example of OpenCTD attached to weighted anchor line using hose clamps.*

## 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.



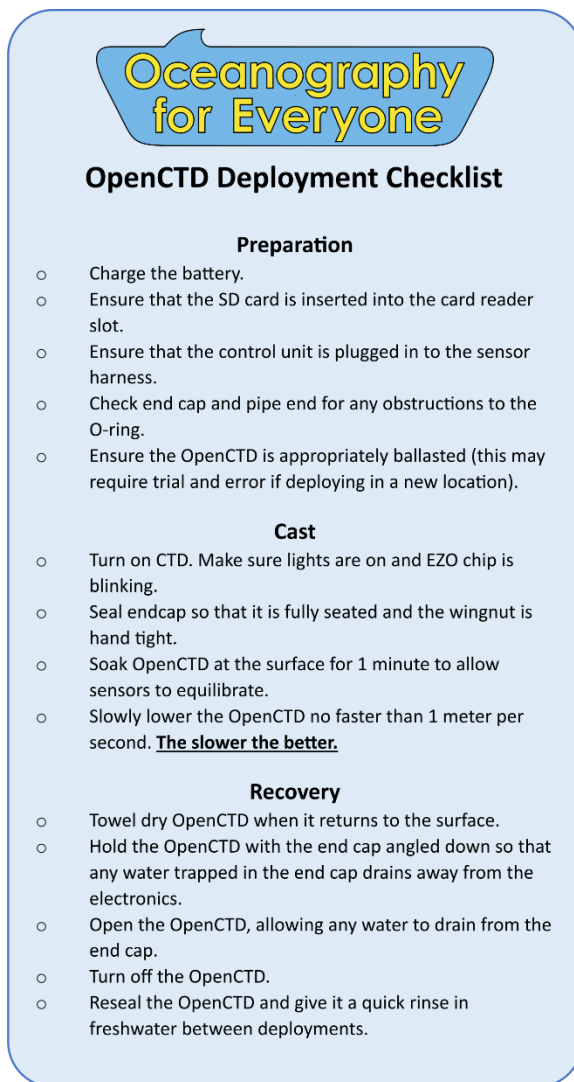
*Figure 30. Clear OpenCTD with fishing weights inside.*

## 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. This is a great opportunity to encourage your students to get creative.

## The First Deployment

It is finally time to make your first cast! If you have an OpenCTD Rev8 Kit or bought stickers from the Oceanography for Everyone store, affix your stickers now.



The image shows a blue-bordered sticker with a light blue background. At the top, there is a logo for "Oceanography for Everyone" in yellow text on a blue background. Below the logo is the title "OpenCTD Deployment Checklist" in bold black text. The checklist is organized into three sections: "Preparation", "Cast", and "Recovery", each with a list of tasks marked by a small circle. The "Preparation" section includes five tasks: charging the battery, inserting the SD card, plugging in the control unit, checking for obstructions, and ballasting. The "Cast" section includes four tasks: turning on the CTD, sealing the endcap, soaking at the surface, and lowering slowly. The "Recovery" section includes five tasks: towel drying, holding at an angle, opening to drain, turning off, and resealing/rinsing.

**Oceanography for Everyone**

**OpenCTD Deployment Checklist**

**Preparation**

- Charge the battery.
- Ensure that the SD card is inserted into the card reader slot.
- Ensure that the control unit is plugged in to the sensor harness.
- Check end cap and pipe end for any obstructions to the O-ring.
- Ensure the OpenCTD is appropriately ballasted (this may require trial and error if deploying in a new location).

**Cast**

- Turn on CTD. Make sure lights are on and EZO chip is blinking.
- Seal endcap so that it is fully seated and the wingnut is hand tight.
- Soak OpenCTD at the surface for 1 minute to allow sensors to equilibrate.
- Slowly lower the OpenCTD no faster than 1 meter per second. **The slower the better.**

**Recovery**

- Towel dry OpenCTD when it returns to the surface.
- Hold the OpenCTD with the end cap angled down so that any water trapped in the end cap drains away from the electronics.
- Open the OpenCTD, allowing any water to drain from the end cap.
- Turn off the OpenCTD.
- Reseal the OpenCTD and give it a quick rinse in freshwater between deployments.

Figure 31. OpenCTD deployment sticker with a checklist of task to complete before, during, and after each cast.

The OpenCTD is sealed with a 2” plumbers test cap, tightened with a large wingnut. The wingnut should be hand tight. We have provide a 3D-printed wrench to aid in tightening the test cap to the appropriate level.

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 benchmark provides a good margin of error for operation. **Before the first data cast, 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, and especially if it has sat unused for a while, 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 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.

While handling the OpenCTD, take care not to damage the sensors, especially the gel pack around the pressure sensor. Sand can be particularly destructive to the gel pack and can become lodged inside the through hole on the conductivity probe. When working from a beach, take extra care to avoid allowing the sensors to come in contact with the sand.

## Stewardship

Whenever you introduce a new tool into the marine environment, you run the risk of introducing additional harm 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 harm 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>)

## Works Cited

- Fofonoff, N.P., Millard Jr, R.C., 1983. Algorithms for the computation of fundamental properties of seawater. UNESCO Technical Papers in Marine Sciences 44.  
<https://doi.org/10.25607/OBP-1450>
- 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., Sturdivant, S.K., Neches, R.Y., Levenson, J.J., 2024. The OpenCTD: A Low-Cost, Open-Source CTD for Collecting Baseline Oceanographic Data in Coastal Waters. *Oceanography* 37, 167–173. <https://doi.org/10.5670/oceanog.2024.602>
- 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.

## Resources and Datasheets

### General Guides

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

[mightyohm.com/files/soldercomic/FullSolderComic\\_EN.pdf](https://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>

### Sensors

MS5803-14BA Miniature 14 bar Module:

[https://cdn.sparkfun.com/datasheets/Sensors/Weather/ms5803\\_14ba.pdf](https://cdn.sparkfun.com/datasheets/Sensors/Weather/ms5803_14ba.pdf)

EZO-EC Conductivity Circuit Datasheet – [https://www.atlas-](https://www.atlas-scientific.com/_files/_datasheets/_circuit/EC_EZO_Datasheet.pdf)

[scientific.com/\\_files/\\_datasheets/\\_circuit/EC\\_EZO\\_Datasheet.pdf](https://www.atlas-scientific.com/_files/_datasheets/_circuit/EC_EZO_Datasheet.pdf)

EC K 1.0 Conductivity Probe Datasheet – [https://www.atlas-](https://www.atlas-scientific.com/_files/_datasheets/_probe/EC_K_1.0_probe.pdf)

[scientific.com/\\_files/\\_datasheets/\\_probe/EC\\_K\\_1.0\\_probe.pdf](https://www.atlas-scientific.com/_files/_datasheets/_probe/EC_K_1.0_probe.pdf)

How to properly cut your probe cable – [https://files.atlas-scientific.com/how-to-properly-](https://files.atlas-scientific.com/how-to-properly-cut-probe-cables.pdf)  
[cut-probe-cables.pdf](https://files.atlas-scientific.com/how-to-properly-cut-probe-cables.pdf)

Conductivity Accuracy Graph – [https://www.atlas-](https://www.atlas-scientific.com/_files/instructions/conductivity_accuracy_graph.pdf)

[scientific.com/\\_files/instructions/conductivity\\_accuracy\\_graph.pdf](https://www.atlas-scientific.com/_files/instructions/conductivity_accuracy_graph.pdf)

## Bill of Materials

While we strive to maintain an up-to-date Bill of Materials with the best available prices, local availability and price fluctuations mean that this will never reflect the best prices for every component. Parts that are only available in bulk may be difficult to source individually. Always check alternative suppliers for better prices. The leftmost column corresponds to where each part appears in the checklist and, subsequently, the build order.

	Item	Links	Price	Price per CTD
<b>Prepare the Adalogger M0</b>	Adafruit Feather M0 Adalogger	<a href="https://www.adafruit.com/product/2796">https://www.adafruit.com/product/2796</a> <a href="https://amzn.to/36DrCm4">https://amzn.to/36DrCm4</a> <a href="https://www.digikey.com/en/products/detail/adafruit-industries-llc/2796/5804105">https://www.digikey.com/en/products/detail/adafruit-industries-llc/2796/5804105</a>	\$19.95	\$19.95
	Pack of 2 Micro SD Card 4GB Flash Memory Card Micro SDHC with SD Adapter (4 GB)	<a href="https://amzn.to/4gfQVAU">https://amzn.to/4gfQVAU</a> Check local electronics stores and online vendors	\$15.49	\$7.75
	MicroUSB Cable	<a href="https://amzn.to/3CEEIp0">https://amzn.to/3CEEIp0</a> Check local electronics stores and online vendors	\$5.24	\$2.62

	Item	Links	Price	Price per CTD
<b><u>Prepare the Carrier Board</u></b>	OpenCTD Rev8 Custom Carrier Board	Available from Oceanography for Everyone <a href="https://oceanographyforeveryone.bigcartel.com/">https://oceanographyforeveryone.bigcartel.com/</a>	\$15.00	\$15.00
	Pin Header, Male and Female Single Row Header Pins	<a href="https://amzn.to/3Sqe4Xo">https://amzn.to/3Sqe4Xo</a> Check local electronics stores and online vendors	\$7.49	\$1.12
	Breadboard-friendly SPDT Slide Switch	<a href="https://www.adafruit.com/product/805">https://www.adafruit.com/product/805</a> Check local electronics stores and online vendors	\$0.95	\$0.95
	10 kΩ Resistor (2)	<a href="https://amzn.to/3JcUFjk">https://amzn.to/3JcUFjk</a> Check local electronics stores and online vendors	\$7.85	\$0.16
	JST PH 2mm 4-pin Vertical Connector	<a href="https://www.adafruit.com/product/4390">https://www.adafruit.com/product/4390</a> Check local electronics stores and online suppliers	\$3.50	\$0.35

	Item	Links	Price	Price per CTD
<b><u>Prepare the Real-Time Clock</u></b>	Adafruit DS3231 Precision RTC	<a href="https://www.adafruit.com/product/5188">https://www.adafruit.com/product/5188</a>	\$13.95	\$13.95
	CR1220 3V Coin Cell Battery	<a href="https://www.adafruit.com/product/380">https://www.adafruit.com/product/380</a> <a href="https://amzn.to/2NYFQpf">https://amzn.to/2NYFQpf</a> Check local electronics stores and online suppliers	\$0.95	\$0.95

	Item	Links	Price	Price per CTD
<b><u>Prepare the MAX31865 PT1000 Temperature Amplifier</u></b>	MAX31865 PT1000 Temperature Amplifier	<a href="https://www.adafruit.com/product/3648">https://www.adafruit.com/product/3648</a>	\$14.95	\$14.95
	3-pin JST SM Plug + Receptacle	<a href="https://www.adafruit.com/product/1663">https://www.adafruit.com/product/1663</a>	\$1.50	\$1.50

	Item	Links	Price	Price per CTD
<b><u>Assemble the Control Unit</u></b>	Black Nylon Machine Screw and Stand-off Set – M3 Thread	<a href="https://www.adafruit.com/product/4685">https://www.adafruit.com/product/4685</a> Check local electronics stores and online suppliers	\$16.95	\$0.36
	Mini Conductivity K 1.0 Kit	<a href="https://atlas-scientific.com/kits/mini-conductivity-k-1-0-kit/">https://atlas-scientific.com/kits/mini-conductivity-k-1-0-kit/</a>	\$199.99	\$199.99

	Item	Links	Price	Price per CTD
<b>Powering the OpenCTD</b>	Lithium Ion Polymer Battery Ideal For Feathers - 3.7V 400mAh	<a href="https://www.adafruit.com/product/3898">https://www.adafruit.com/product/3898</a>	\$6.95	\$6.95

	Item	Links	Price	Price per CTD
<b>Prepare the PT1000 Temperature Probe</b>	Platinum RTD Sensor - PT1000	<a href="https://www.adafruit.com/product/3984">https://www.adafruit.com/product/3984</a>	\$14.95	\$14.95
	Solder Seal Wire Connectors, Waterproof Heat Shrink Butt Connectors Terminals	<a href="https://amzn.to/4vfpDPU">https://amzn.to/4vfpDPU</a> Check local electronics stores and online suppliers	\$26.99	\$0.47

	Item	Links	Price	Price per CTD
<b><u>Prepare the Sparkfun Pressure Sensor</u></b>	SparkFun Pressure Sensor Breakout - MS5803-14BA	<a href="https://www.sparkfun.com/sparkfun-pressure-sensor-breakout-ms5803-14ba.html">https://www.sparkfun.com/sparkfun-pressure-sensor-breakout-ms5803-14ba.html</a>	\$80.50	\$80.50
	22 Gauge Silicone Wire Kit 10 Color Each 5 ft Flexible 22 AWG Stranded Tinned Copper Wire	<a href="https://amzn.to/4aOldae">https://amzn.to/4aOldae</a> Check local electronics stores and online suppliers	\$11.48	\$0.92
	JST Plug Connectors	<a href="https://amzn.to/4vi6MUq">https://amzn.to/4vi6MUq</a> Check local electronics stores and online suppliers	\$7.39	\$0.18

	Item	Links	Price	Price per CTD
<b><u>Cutting the PVC Housing</u></b>	2" PVC Pipe 1FT Long Sch40	<a href="https://amzn.to/4orlhCy">https://amzn.to/4orlhCy</a> Check local hardware stores and online suppliers	\$15.99	\$8.00

	Item	Links	Price	Price per CTD
<b><u>Pot the Pressure Sensor</u></b>	Loctite Epoxy Five Minute Instant Mix	<a href="https://amzn.to/3tuKM8q">https://amzn.to/3tuKM8q</a> Check local hardware stores and online suppliers	\$14.49	\$7.25

	Item	Links	Price	Price per CTD
<b><u>Potting the OpenCTD</u></b>	Epoxy, Loctite E-120hp, 1.69 FL. oz. Cartridge 6430A24	<a href="https://www.mcmaster.com/products/6430a24/">https://www.mcmaster.com/products/6430a24/</a>	\$24.13	\$24.13
	Long Taper Tip Nozzle with Bayonet Connection for Two-Part Cartridge	<a href="https://www.mcmaster.com/74695A12/">https://www.mcmaster.com/74695A12/</a>	\$2.57	\$0.13

	Item	Links	Price	Price per CTD
<b><u>Final Preparation and First Deployment</u></b>	Hose clamp (AB-36/46mm to 70mm)	<a href="https://amzn.to/3t7G1UX">https://amzn.to/3t7G1UX</a> Check local hardware stores and online suppliers	\$14.99	\$6.00
	1/4" PolyPro Rope	<a href="https://amzn.to/3w7su1G">https://amzn.to/3w7su1G</a> Check local hardware stores and online suppliers	\$7.80	\$0.32
	Oatey Gripper 2" Mechanical Test Plug	<a href="https://www.lowes.com/pd/Oatey-2-in-Gripper-PVC-DWV-Test-Plug/3880607">https://www.lowes.com/pd/Oatey-2-in-Gripper-PVC-DWV-Test-Plug/3880607</a> Check local hardware stores and online suppliers	\$8.18	\$8.18

	Item	Links	Price	Price per CTD
<b><u>Calibration</u></b>	Conductivity Calibration K 1.0 Set	<a href="https://atlas-scientific.com/calibration-solutions/conductivity-calibration-k-1-0-set/">https://atlas-scientific.com/calibration-solutions/conductivity-calibration-k-1-0-set/</a>	\$15.99	\$15.99

## Alternate Pressure Sensor

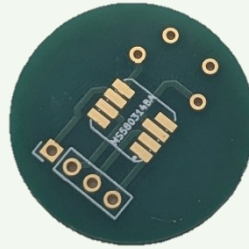
	Item	Links	Price	Price per CTD
<b>Alternate Pressure Sensor</b>	MS5803-14BA Pressure Sensor SMC	<a href="https://www.digikey.com/en/products/detail/te-connectivity-measurement-specialties/MS580314BA01-00/5277631">https://www.digikey.com/en/products/detail/te-connectivity-measurement-specialties/MS580314BA01-00/5277631</a>	\$23.10	\$23.10
	OpenCTD Custom Pressure Sensor Breakout Board	Available from Oceanography for Everyone	\$10.00	\$10.00
	10 kΩ Resistor	<a href="https://amzn.to/3JcUFjk">https://amzn.to/3JcUFjk</a> Can be sourced from any electronics supplier	\$6.18	\$0.06
	0.1 uF Capacitor	<a href="https://amzn.to/32tMBol">https://amzn.to/32tMBol</a> Can be sourced from any electronics supplier	\$5.92	\$0.24

## Appendix A. Alternate Pressure Sensor

### Components for Oceanography for Everyone Pressure Sensor Breakout Board



*MS5803-14BA Pressure Sensor SMC*



*OpenCTD Pressure Sensor Breakout Board*



*10 kOhm Resistor*



*0.1 uF Capacitor*

## DIY Pressure Sensor

To build the pressure sensor, you will need to surface mount the pressure sensor chip onto the OpenCTD Pressure Sensor Breakout Board. 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.

1. Add a small amount of solder to each pad. If you have solder flux, add a dab of flux 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 OpenCTD Pressure Sensor Breakout Board. 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. On the side opposite the pressure chip, solder a 10 kOhm pull-up resistor to the pins labeled 10 kOhm.
4. On the side opposite the pressure chip, solder a 0.1 $\mu$ F capacitor to the pins labeled 0.1 $\mu$ F.
5. On the side opposite the pressure chip solder a 20 cm length of wire to the pins labeled CLK, GND, SDA, and VCC. Keep track of which wire goes to which pin by color-coding each pin (we recommend red for VCC; black for ground; yellow for CLK; and blue for SDA).
6. Strip 1/4 cm of insulation from the exposed ends of each wire.