Oceanography for Everyone

@DrAndrewThaler

Construction and Operation of the

OpenCTD

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.

Fourth Edition: December 2023.

Cover Art by Andrew Thaler.

Photographs by Andrew Thaler and Allie Wilkinson unless otherwise noted.

Contents

1 Overview

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

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

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

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

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

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

The ocean is not uniform, it its filled with swirling eddies, temperature boundaries, layers of high and low salinity, changing densities, and other physical characteristics invisible to surface observers (Sverdrup et al., 1942). By measuring salinity, temperature, and depth, scientists can begin to unlock ocean patterns hidden beneath the sea's surface. To reveal these patterns, oceanographers use a CTD—an instrument that measures conductivity (used to calculate salinity), temperature, and absolute pressure (used to determine water depth; Thomson and Emery, 2014).
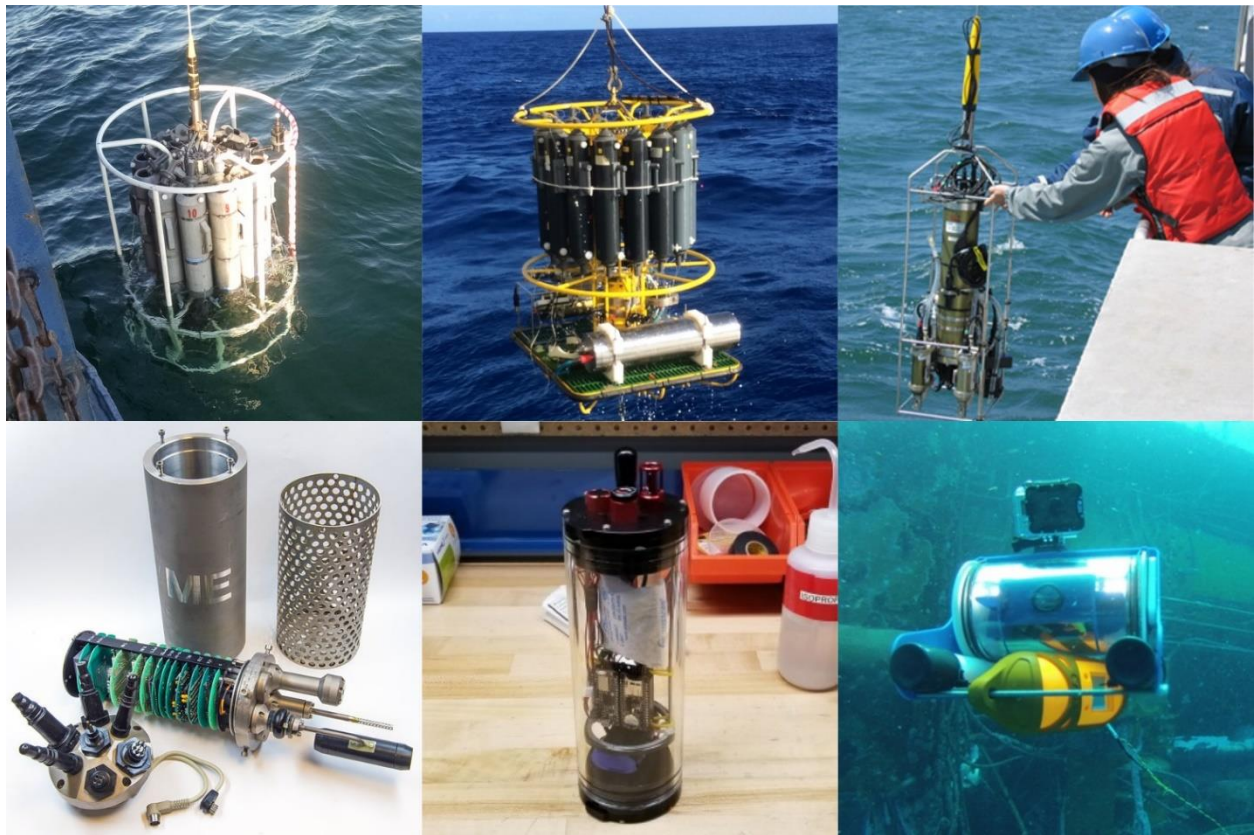
Figure 1. A selection of various CTDs used for field work. Top left: A CTD and rosette deployed from a NOAA boat (WHOI); top center: A CTD and rosette deployed by IFREMER; top right: a student deploys a CTD (MIT); bottom left: an 80s era ME GRISARD CTD; bottom center: the CTDizzle open-source CTD by 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. For near-shore oceanographic research on the continental shelf, where the water depth rarely exceeds 200 meters, this cost can be prohibitive. The expense of the CTD is a barrier-to-entry for formal and informal researchers working with limited budgets, including scientists from low- and middle-income countries and small island states, as well as citizen oceanographers, environmental educators, conservation and management practitioners, and students of all levels interested in understanding their local coasts and waterways.

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

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

The OpenCTD is a low-cost, open-source CTD designed intentionally for budget-restricted scientists, educators, and researchers working in nearshore coastal ecosystems, where entire research projects can be conducted for less than the cost of a commercial CTD. It was developed by a core team of marine ecologists in collaboration with a distributed community of scientists, engineers, makers, and conservation practitioners. It is assembled from components commonly available at large hardware stores or through major online retailers.

The OpenCTD is designed to be built by the end-user, providing both access to the tools of oceanography as well as the skills to maintain, repair, and replace OpenCTDs. For scientists working in remote settings, where commercial 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.

1.1 Building Skills

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

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

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

If you would prefer to practice on something simpler to get started, the TV-B-Gone and Space Shuttle Discovery Solder Kit by Adafruit are great introductory kits to teach soldering and basic electronic principles to students of all ages. Preparing the components needed for an OpenCTD will provide most students with sufficient grounding to tackle the more complex elements of constructing an OpenCTD.

- **Space Shuttle Discovery Solder Kit:** https://www.adafruit.com/product/5414
- **TV-B-Gone Kit - Universal v1.2:** https://www.adafruit.com/product/73

Arduino is a coding environment built on C/C++ and designed for flexibility and ease of use, but it can still seem daunting to first-time users. Practically, the OpenCTD source code is complete, and most users will only need to make small alterations to 3 lines of code during assembly and testing, but it is helpful to understand what the software is doing. Calibration and troubleshooting will require users to load different programs onto the completed OpenCTD. 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**

1.2 Project Status and Compatibility

OpenCTD is stable (as of 6/21/2023) on the ATSAMD21G18 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. There are currently two parallel OpenCTDs currently in development based on different sensor platforms.

An older version of OpenCTD is stable on the ATMEGA32U4 processor common to many Arduino systems. Construction workflow, as well as pinouts, bill of materials, and operation 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.

1.3. 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.

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

2 Materials, Tools, and Consumables

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

We created a custom printed circuit board that dramatically improves the construction experience. We also provide an alternative process using standard prototyping boards (Appendix A) for users who can't source custom PCBs.

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

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

Raw, calibrated absolute pressure sensors are available in both 14-Bar and 30-Bar modules from several vendors. While we recommend that you use pressure sensor breakout boards available from a variety of manufacturers, we also provide instructions for building your own pressure sensor from principal components (Appendix B).

While we provide a list of recommended parts and their suppliers, these are not necessarily the least expensive option and prices can vary significantly depending on local availability and 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 GitHub.

When available, we provide Amazon Affiliate links 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. We are, however, not endorsing Amazon and encourage users to seek out local suppliers and small businesses whenever possible.

2.1 Tools and Consumables

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

Tools for general electronics work: flush cutters, wire strippers, soldering iron, solder tip cleaner, helping hands, needle-nosed pliers, utility knife, heat gun.
Consumables for general electronics work: lead-free solder, electrical tape, 22-gauge silicone wire or 22-gauge solid core wire (if building the protoboard version), heat shrink tubing for small gauge wire.

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

Tools for housing and adhesion: Hand saw or PVC pipe cutter, drill with 1" bit or router and jig, 3M epoxy gun for 50ml cartridges.
Consumables for housing and adhesion: LOCTITE 2-part Adhesive Epoxy (Hysol E-120HP), venturi mixing nozzles for 50 mL epoxy cartridges, LOCTITE Epoxy Five Minute Instant Mix, sandpaper.

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

A note on Epoxies and Glues:

We 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 other marine hardware, however it is not always easy to source. Other two-part epoxies can also be used for building the sensor pass-though, 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.

2.2 Andrew's Toolkit

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

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

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

- **For benchtop work with reliable power:** Hakko FX888D-23BY Digital Soldering Station (https://amzn.to/3vCWNuj)
- **For field work with limited power access:** Hakko FX-901/P Cordless Soldering Iron (https://amzn.to/30Uz62r)
- **For workshops, small spaces, or where USB-C power supplies are preferable**: 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: zPower Turbo 65W USB-C Power Adapter (https://amzn.to/3CTNLOd).
- **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)
- **Chop saw:** The sliding compound miter saw Andrew uses is no longer in production, but pretty much any standard chop/miter saw will do and, in a pinch, a handsaw works just fine.

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

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

Programming and Calibration:

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

Consumables:

- **PLA Filament:** PolyTerra™ PLA (https://us.polymaker.com/products/polyterra-pla)
- **Lead-free solder:** Mudder Lead Free Solder Wire (https://amzn.to/38WOmAl)
- **Electrical tape:** 3M Scotch #35 Electrical Tape (https://amzn.to/2QeA9rS)
- **22-gauge silicone wire:** 22 Gauge Silicone Wire Kit (https://amzn.to/3CN0F0z)
- **Heat shrink tubing:** NTE Heat Shrink 2:1 (https://amzn.to/3lEIUY4)
- **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)
- **Venturi mixing nozzles for 50 mL epoxy cartridges:** 3M Static Mixing Nozzle, 38193, 50 mL (https://amzn.to/30YJB4N)
- **Five-minute epoxy:** Loctite Epoxy Five Minute Instant Mix (https://amzn.to/3tuKM8q)

2.2 Illustrated Parts List

The below section contains photographs of each component used in the OpenCTD build process. Please reference this section as you construct your CTD.

3D-printed parts

Control Unit Chassis

Sensor Baseplate for Sparkfun/DIY Pressure Sensor

Protective Cap

Sensor protector

Housing

Schedule 40 PVC Pipe

Oatey Plumber's Test Cap

46 to 70mm Hose Clamp

Short Length of Polypro Rope

OpenCTD Rev6 Custom Carrier Board

Custom Carrier Board

Adafruit Adalogger M0

SD Card

Real-time Clock Module

CR1220 Coin Cell Battery

Atlas EZO Conductivity Circuit

Female Headers
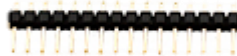
Male Headers

7-pin JST Header

MicroUSB Cable

10 kOhm resistor

4.7 kOhm Resistor

Sensors

Temperature Sensor

Conductivity Probe

Sparkfun 14-Bar Pressure Sensor Breakout Board

JST Female Connector

Power and Ballast

Battery

Board-mounted Switch

Assorted weights for ballasting CTD

Optional Accessories

Solderless Breadboard

Screw Terminals

Alligator Clip

Jumper wire

SD Card Reader

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

3 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.

3.1 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 2nd 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 the 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.

3.2 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.

3.3 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 approved life jacket.

3.4 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.

4 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 supports and protects the control unit as well as the sensor baseplate, 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 two parts which fit snugly together and are held in place with a zip tie.

All parts are designed to be printed in PLA at 200-micron resolution with 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 harm 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.

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

- o PolyTerra™ PLA (https://us.polymaker.com/products/polyterra-pla)

All 3D printer files are available in the OpenCTD GitHub repository:
https://github.com/OceanographyforEveryone/OpenCTD/tree/main/Hardware/3DPrints

OpenCTD 3D Printed Structural Components:

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.

There are two auxiliary caps which can be used to cover the access port and sensor passthrough. These can be used to protect the device during travel and shield the sensor when painting. These components are not necessary to complete the build.

The electronics chassis will need to be printed with supports. Once the supports are removed, test fit the two halves to ensure that they slot together firmly. If they are too tight, use the deburring tool or some sandpaper to shave down the inner walls of the top chassis piece.
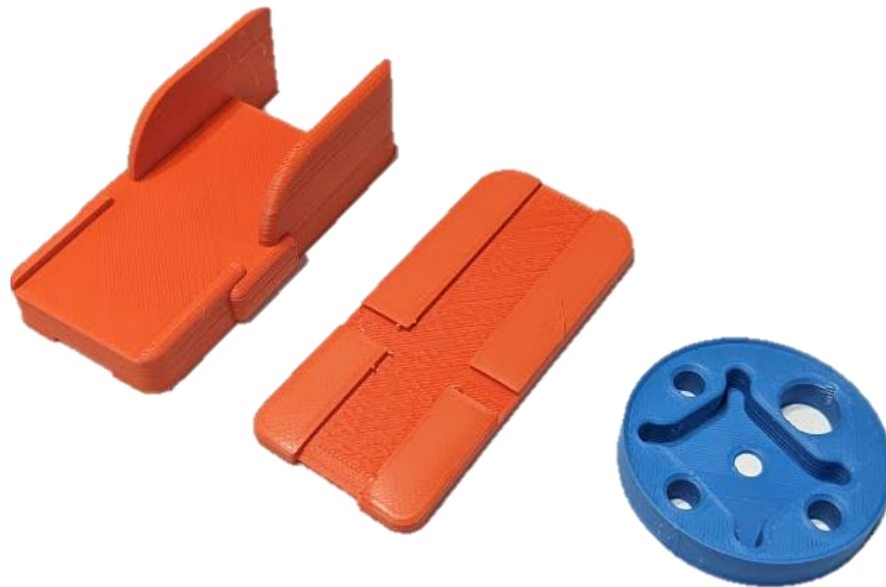
Figure 2: 3D-Printed parts for OpenCTD.

5 Prepare the Housing

The OpenCTD housing is comprised of a single piece of schedule-40 2" PVC pipe. Notches or holes cut in the bottom allow water to flow over the sensors while protecting them from impacts and entanglement. The sensor end will be sealed with high sheer-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.)**

1. Cut a 12" section of PVC pipe. Deburr the edges around each cut with a knife, sandpaper, or deburring tool.

2. If you have access to a router table, cut 4 1-inch-deep notches into one end of the PVC using a ½-inch router bit. If you have access to a drill and bit, cut 3 1-inch holes centered ¾-inch from the bottom.

3. Lightly sand the inside bottom of the tube, just above where you cut the slots or holes. This will provide a good surface for the epoxy to adhere to.
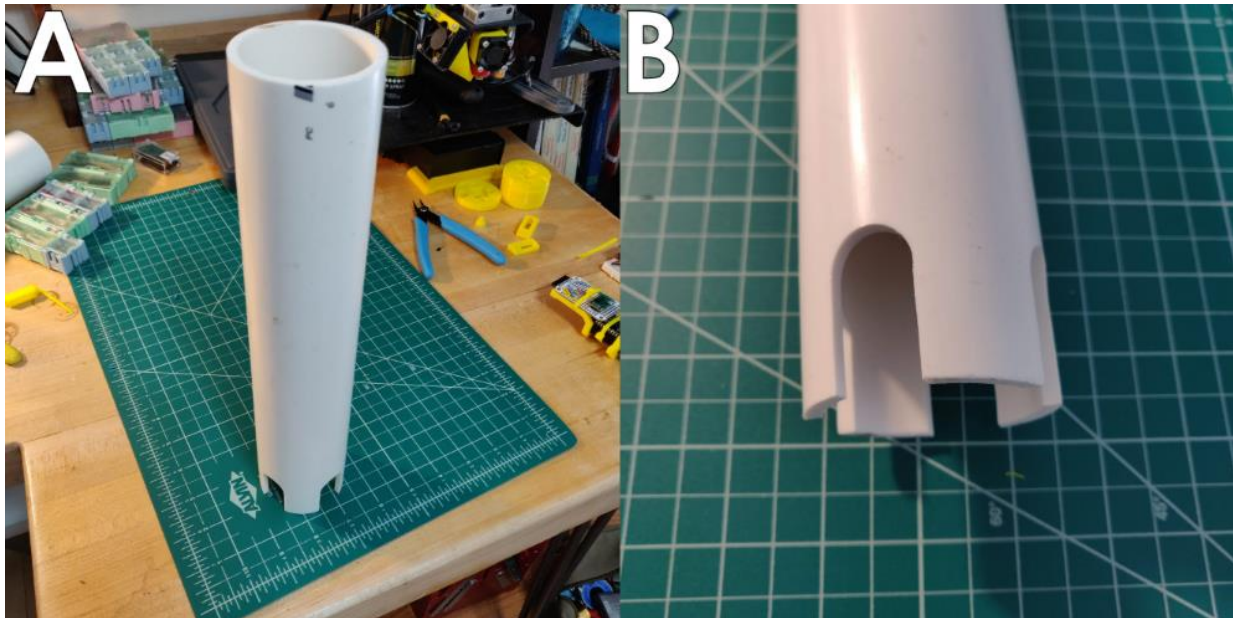
Figure 3. 12" PVC pipe (A) with notches cut at the base (B).

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

mechanical pass-throughs is the single largest cost-saving step in the entire CTD construction process. 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.

6 Build the Control Unit

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

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

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

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

A "pin" refers to any electrical contact point and will generally be used in this guide to refer to the holes in circuit boards through which wires, headers, and other components will be soldered.

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

6.1 Prepare the M0 Adalogger

The Adafruit M0 Adalogger acts as both the main computer for the OpenCTD and writes data to the onboard SD Card drive.

Figure 4. Adalogger M0 without headers. Left: topside of the board. Right: bottom of the board.

1. Slot the male headers into the through-holes on the underside of the Adalogger M0, ensuring that the number of headers matches the number of available pins (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, snip down any protruding pins so that they sit flush with the solder joint.

5. Insert a microSD card into the SD card reader.

Figure 5: Adafruit M0 Adalogger with male headers (SD card not shown).

6.2 Prepare the Carrier Board

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

Figure 6: OpenCTD Carrier Board. Left: topside of the board. Right: bottom of the board.

1. Slot the female headers on the top 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.

- One way to align the pins is to is to plug the male headers from the Adalogger M0 into the female headers and to assemble the carrier board with the headers in position with both boards to ensure alignment.

2. Solder the female headers for the Adalogger M0 to the carrier board, starting with the four corners to hold them in place.

3. Slot the female headers on the underside of the carrier board in the pins in the box labeled DS3231.

4. Solder the female headers for the DS3231 Real-time Clock (RTC) to the underside of the carrier board.

5. Low-profile headers make it easier to remove the SD card from the Adalogger M0. Slot the low-profile female headers on the top of the carrier board in the pins in the box labeled Atlas EC EZO.

6. Solder the low-profile female headers for the Atlas EC EZO to the top of the carrier board.

7. Insert the JST 7-pin connector to the underside of the custom carrier board, ensure that the slot in the plastic connector housing is facing away from the board.

8. Solder the JST connector to the underside of the custom carrier board in the pin row labeled J4.

- **ALTERNATIVE:** if you were unable to source a JST connector, you can solder straight- or right-angle female headers to the 8-pin strip directly adjacent to the JST connector pins in the pin row labeled J3. This will allow you to build a simpler, but less reliable connection between the sensors and the control board.

9. Using the flush cutters, snip down any protruding pins so that they sit flush with the solder joint.

10. Solder the 10 kOhm and 4.7 kOhm resistors to their corresponding slots on the topside of the custom carrier board. You may need to use pliers to pull the resistors through such that they lie flat on the carrier board. Resistors can be inserted in either direction.

- The color code for a 10 kOhm resistor is Brown, Black, Orange, Gold.
- The color code for a 4.7 kOhm is Yellow, Purple, Red, Gold.

- To make it easier to solder the resistors, fold the legs down so that they hold the resistors in position on the board.

11. Using the flush cutters, trim down the protruding legs of the resistors so that they sit flush with the solder joint.
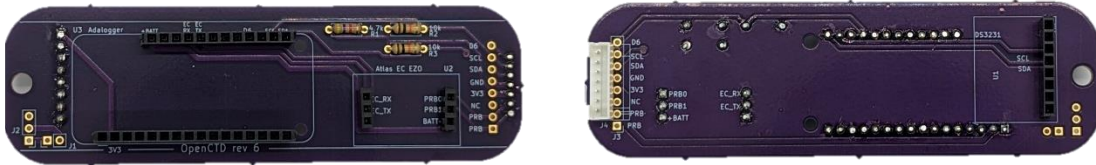

Figure 7. Carrier board populated with headers, resistors, and JST connector. Left: topside of the board. Right: bottom of the board.

6.3 Prepare the RTC

Figure 8: 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 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.

3. Using the flush cutters, snip down any protruding pins so that they sit flush with the solder joint.
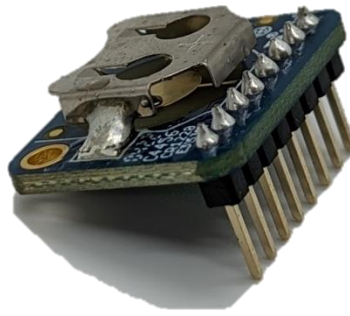
Figure 9.Real-Time Clock module with male headers soldered to the chip side of the board.

6.4 Assemble the Control Unit

1. Stack the Adalogger M0 on top of the custom carrier board by lining up the male, downward-facing header pins on the Adalogger with the female, upward facing header pins on the carrier board.

2. 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.

3. Insert the CR1220 battery into the RTC.

4. Insert the Atlas EZO conductivity circuit into the EZO slots on the topside of the carrier board. Ensure that the pin labels on the board correspond to the pin labels on the module.
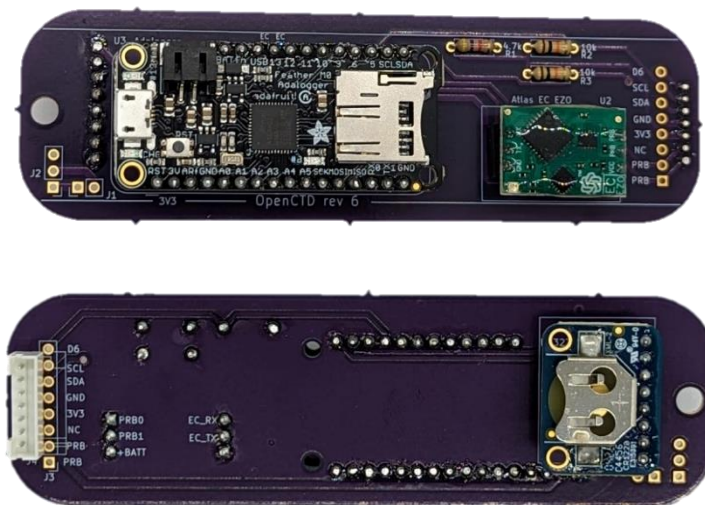

Figure 10. Carrier board populated with Adalogger M0, Atlas EZO EC chip, and real-time clock module. Above: topside of the board. Below: bottom of the board.

6.5 Software and Libraries

To prepare your computer to talk to the M0 Adalogger, download and install Arduino IDE (https://www.arduino.cc/en/Main/Software). You will need to install additional boards so that Arduino IDE will be able to recognize the Adalogger. For a detailed walkthrough of this process, with screenshots included, please visit: https://learn.adafruit.com/adafruit-feather-m0-adalogger/setup. Please be aware that the Arduino, Adafruit, 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.

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

1. Launch Arduino IDE.

2. Under *preferences,* in the *File* menu (on Apples, this in under the *Arduino* menu), add the following URL to the "Additional Boards Manager URLs" field: https://adafruit.github.io/arduino-board-index/package_adafruit_index.json

3. Click "Ok" and then navigate to *Tools > Board > Boards Manager*.

4. In *Boards Manager*, search for "Adafruit SAMD Boards" and install the library. Now search for "Arduino SAMD Boards" and install this library as well.

5. Restart Arduino IDE.

6. Install the required libraries listed below in section 6.6

7. Restart Arduino IDE.

6.6 Required Arduino Libraries

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

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

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

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

MS5803_14. For the MS5803 14-Bar pressure sensor.
Dallas Temperature Control. Allows you to communicate with the DS18B20 thermometer.
OneWire. For controlling the Dallas 1-wire family of devices, which includes the temperature sensors.
RTClib. For controlling and reading the Real-time Clock
SoftwareSerial. A modified version of the standard serial monitoring program that works with SAMD microcontrollers like the Adafruit Feather M0.

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

https://github.com/OceanographyforEveryone/OpenCTD/tree/main/Software/Support

6.7 Flash the OpenCTD software to the Adalogger

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

5. Under *Tools > Port*, select the COM port that reads "Adafruit Feather M0" (if should autodetect, but doesn't always detect as Arduino. M0 modules may come preloaded with Circuit Python but will update the first time you flash Arduino-based firmware to the board).

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. Note: the RTC sets its time based on the time displayed on the computer when the code is compiled. Should the RTC ever lose time, you can use the RTC_Reset program in the OpenCTD GitHub repository under Support to reset the clock to the system time on a connected PC.
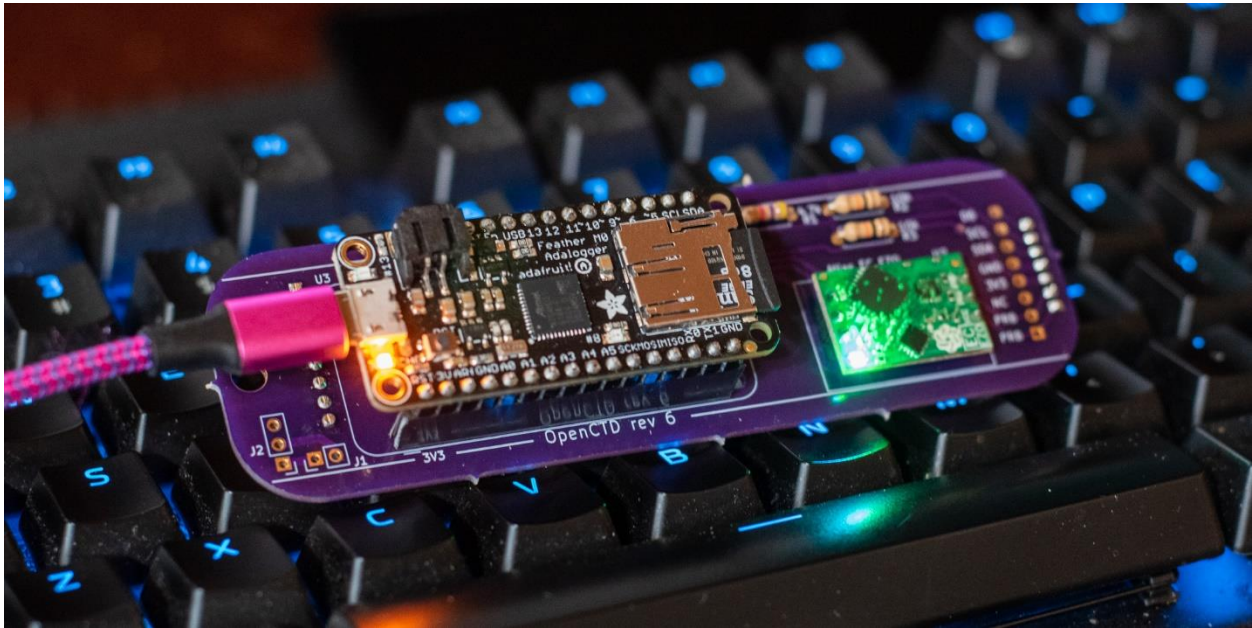

Figure 11. Control unit connected to PC for software validation.

6.8 Confirm Software Installation and Adalogger/RTC assembly.

Once the control unit is assembled and the OpenCTD software is flashed to the Adalogger M0, use the serial monitor in Arduino IDE to verify that everything is working correctly. Open the Serial Monitor by clicking the magnifying glass in the upper righthand corner of Arduino IDE or through *Tools > Serial Monitor*. The software will initialize the SD card reader and RTC and then go through a sensor check, which will fail. You should then see the date and time, followed by a series of zeros or "nan" (not a number) values. The EZO conductivity circuit should blink blue and green.

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

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

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.

Control Unit Checklist:

- o   Adalogger connects to PC.

- o   OpenCTD software compiles and uploads to Adalogger.

- o   Serial monitor connects and can read data.

- o   Real-time clock shows the correct time.

- o   Data is recorded on the SD Card.

7 Build the Sensor Package

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

7.1 Prepare the Sparkfun Pressure Sensor

For the Sparkfun Pressure Sensor Breakout Board:

1. Strip ½ cm off the end of 4 different colored 8-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 by color-coding each pin. Note that the wires must extend from the side opposite the pressure sensor.
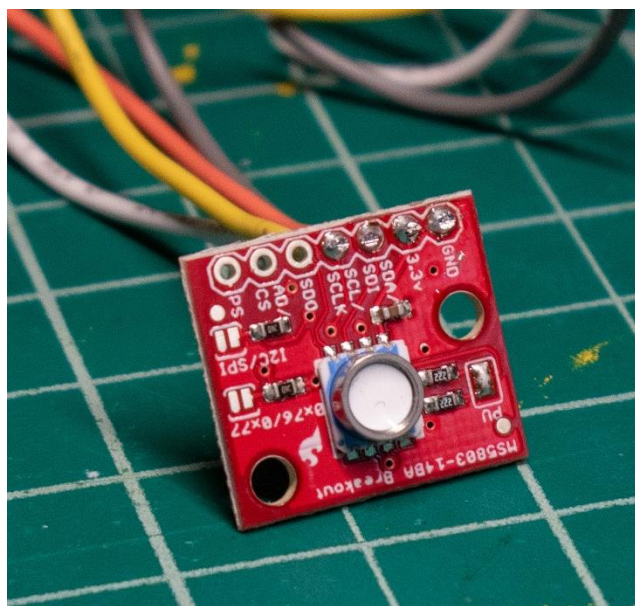

Figure 12. Sparkfun Pressure Sensor Breakout with soldered pins.

Test the Sparkfun Pressure Sensor breakout by connecting the four wires to their corresponding pins on the Adalogger M0, either by using a breadboard, alligator clips, or slotting the stripped terminal ends of the wire through their respective pin holes on the 8-pin strip on the carrier board labeled J3 (be careful not to create a short). Depending on the power from your USB connection, you may also need to temporarily plug the battery into the 2-pin battery JST connector.

7.2 Prepare the DS18B20 Temperature Sensors

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

1. Cut the wire on each temperature probe approximately 5 cm from the probe-end terminus.

2. Strip 2 cm of outer insulation, taking care not to break the inner wires.

3. Strip a centimeter of insulation from each inner wire, exposing the metal strands.

4. Twist all three black wire strands together and "tin" the wires by adding solder to the entire length. This will provide a base for more wire to be soldered to the temperature sensor packet. Do the same for the red and yellow wires, respectively (the colors may vary based on manufacturer, but the important thing is to ensure that matching colors are bundled in sets of three).
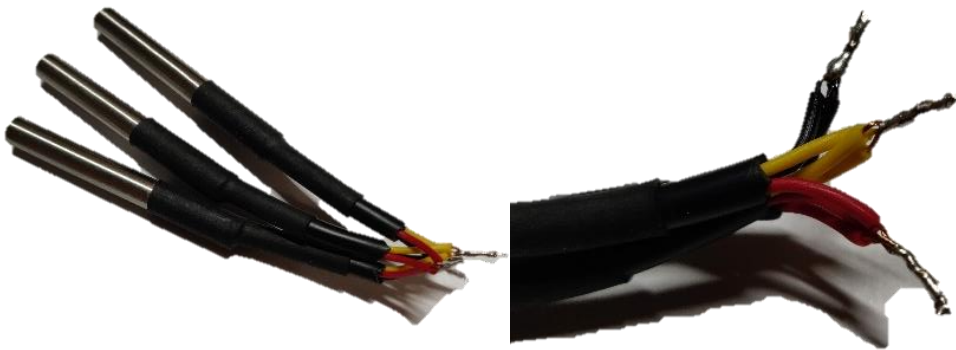
Figure 13. Three potted temperature sensors joined together with a close-up on the tinned tips.

7.3 Prepare the Atlas Conductivity Sensor

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

1. Cut the probe wire such that the full length of the probe and wire extends a little more than 2/3rds of the way up the PVC pipe.

2. Use a utility knife to strip and expose 2 cm of wire.

    a. The wires may be co-axial (one wire is wrapped around the insulated core of the other) in which case you will have to separate the shroud from the central core and expose both (see Figure 14 below). The shrouded wire will need to be twisted together once extracted from the insulation. Ensure that the insulation in the internal wire extends far enough that that wire will not contact the exposed external wire.

    b. The wires may be embedded in extruded insulation rather than simply wrapped and will require a bit of care to extract without damaging the internal insulation.

Figure 14. Conductivity probe leads with coaxial wires stripped and soldered.

3. Strip half a centimeter of insulation from each inner wire. Ensure that the insulation on the innermost wire extends at least a centimeter beyond the outer wire to avoid accidental contact.

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

7.4 Pot the Pressure Sensor and Build the Sensor Wiring Harness

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 snuggly into the central hole. This sensor will be potted first with 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.

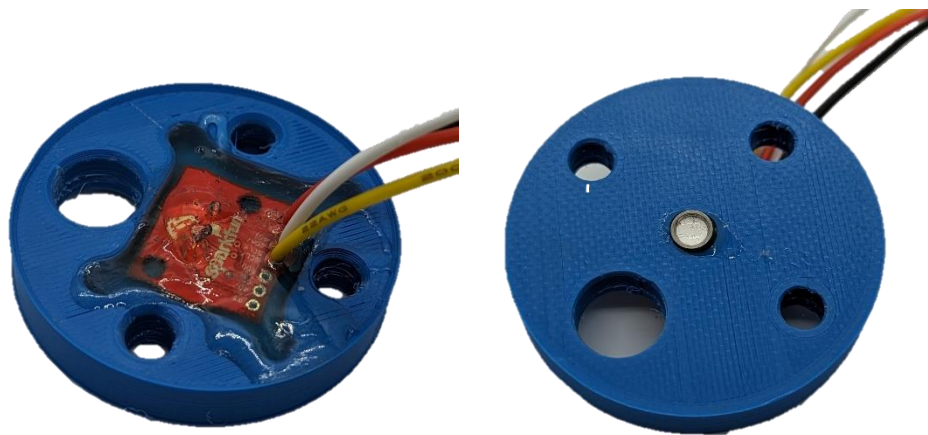4. Let the assembly sit for a 10 to 20 minutes to allow the epoxy to set.

Figure 15: SparkFun MS5803-14BA Pressure Sensor Breakout potted in Epoxy. Topside view on right, underside on left.

5. Once the epoxy has set, insert each temperature probe into the three holes such that 1.5 to 2 cm extends out of the bottom of the baseplate.

6. The pressure and temperature sensors share power. Solder the 3V3 (or VCC) wire from the pressure sensor to the positive (red) wire cluster of the temperature sensor assembly.

7. Solder the GND wire from the pressure sensor to the negative (black) wire cluster of the temperature sensor assembly.

8. Solder additional wire on to extend the GND, 3V3, DATA (the third wire on the temperature sensor cluster), SCL, and SCK wires to align with the wires on the conductivity probe. Seal each joint with heat shrink tubing.

9. Insert the conductivity probe into the large hole such that 1.5 to 2 cm extends out of the bottom of the baseplate. If the probe will not seat into the baseplate, use a knife, deburring tool, or other cutting instrument to slightly widen the opening until the sensor seats. It will be a very snug fit.
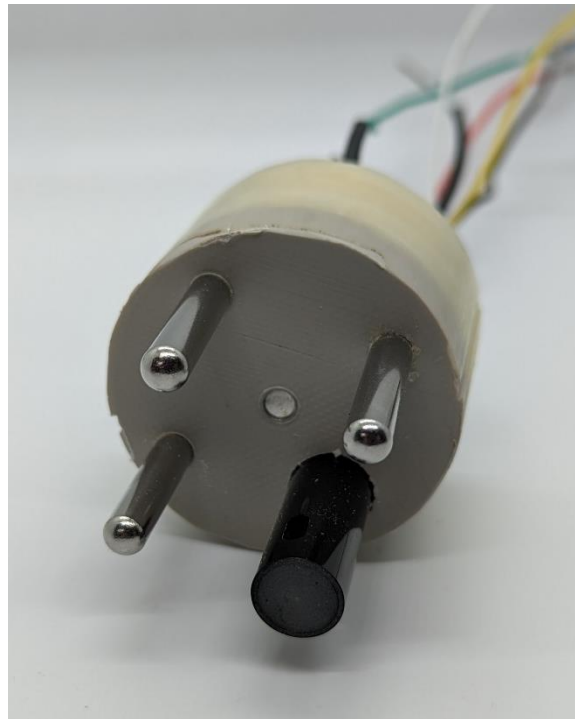
Figure 16: OpenCTD Baseplate with all sensors installed. Conductivity is the large black probe. Pressure is the small white circle in the center. The three temperature probes are the silver probes surrounding the pressure sensor.

7.5 Attach the JST Connector

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

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

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

   a. **ALTERNATIVE:** If you are crimping your own JST female connector with wires, note which wire colors you use for each connection.

2. You will want the JST connector to stick out about 8 centimeters from the top of the PVC pipe.

3. Trim 8 pieces of heat shrink tubing and slide them down to the base of the JST connector wiring harness. These will be use to insulate soldering joints after the wires are connected.

4. Solder the two wires from the conductivity probe to the wires on the JST connector that correspond to the pins marked PRB. As the conductivity probe is bidirectional, it does not matter which of the two PRB leads connects to which pin. Seal the soldered joint with heat shrink. You may need to use several layers of heat shrink to insulate the coaxial portion of the wire.

5. Solder the GND and 3V3 wires to the leads for the positive and negative terminals. Seal the solder joint with heat shrink.

6. Solder the SDA, SCL, and D6 wires to the leads for SDA and SCL on the pressure sensor and DATA on the temperature sensors. Seal the solder joint with heat shrink.

7. Using a large piece of heat shrink, seal the entire area around the solder joints for added protection.
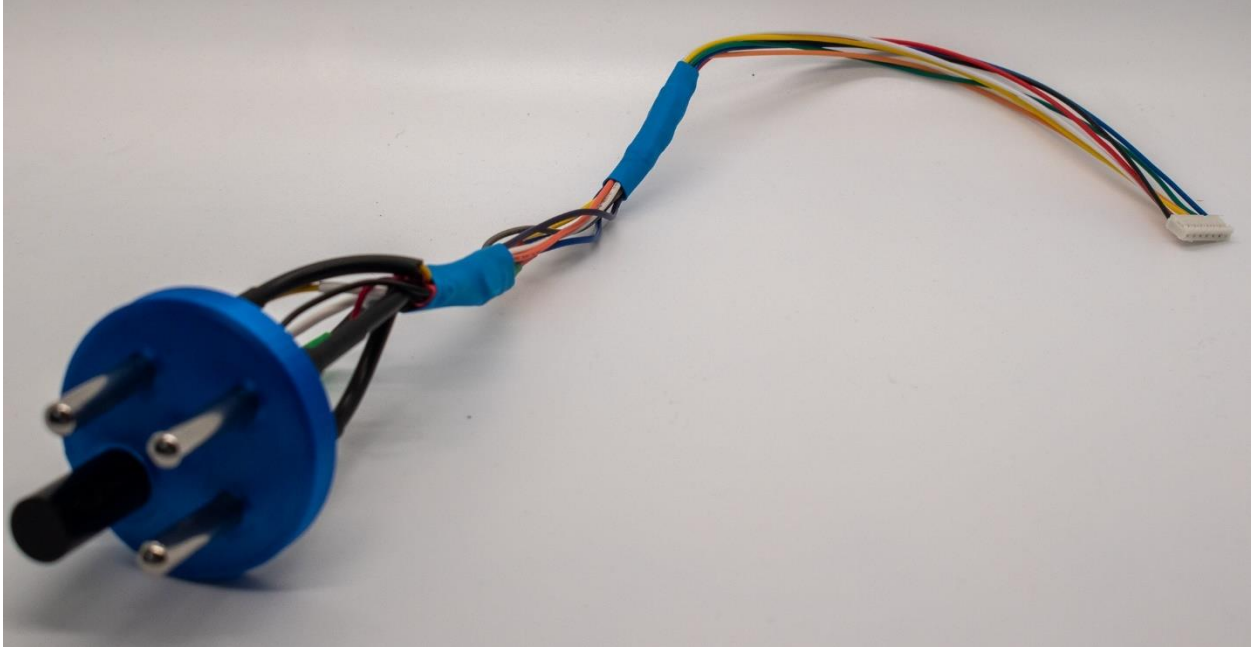
Figure 17. Completed sensor package.

8. **ALTERNATIVE**: You can use pre-made solder-seal wire connectors which contain a glob of solder inside an adhesive-lined heat shrink tube to solder and seal the wires in one step. This is a much easier process and results in a waterproof seal.

Checklist:

o 8-pin JST connector connects securely to JST socket on control board.

o OpenCTD powers on and connects to sensor packet.

o Pressure sensor engages and reports a value around 1010 mbar (if at sea level).

o Temperature sensors engage and report a temperature that matches ambient temperature.

o Conductivity sensor engages and reports a value when suspended in conductive fluid (this will be wildly inaccurate until calibration).

7.6 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 epoxy cures. Within the first 20 minutes, you may gently lift the housing vertically to check that no epoxy is leaking out the bottom. After about 2 hours of cure time, the epoxy should have set enough to allow you to continue the build process without disturbing it. Do not submerge the OpenCTD 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 site slightly back from the rim of the PVC pipe and baseplate clears any notches cut in the sides of the pipe.

3. Ensure that the inside of the PVC pipe is clean and free of obstruction.

<p style="text-align:center;color:red;font-weight:bold;">ENSURE EVERYTHING IS SET CORRECTLY
BEFORE PROCEEDING TO THE NEXT STEP.</p>

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

5. Check to ensure that nothing is leaking and let the epoxy cure undisturbed for up to 24 hours. We have generally found that an overnight cure is sufficient for testing at the surface.

Figure 18. A clear OpenCTD showing the thick epoxy potting layer.

8 Power the OpenCTD

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

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

1. Solder the 3-pin toggle switch to the pins labeled J2 on the topside of the carrier board. Trim the excess pins from the underside.
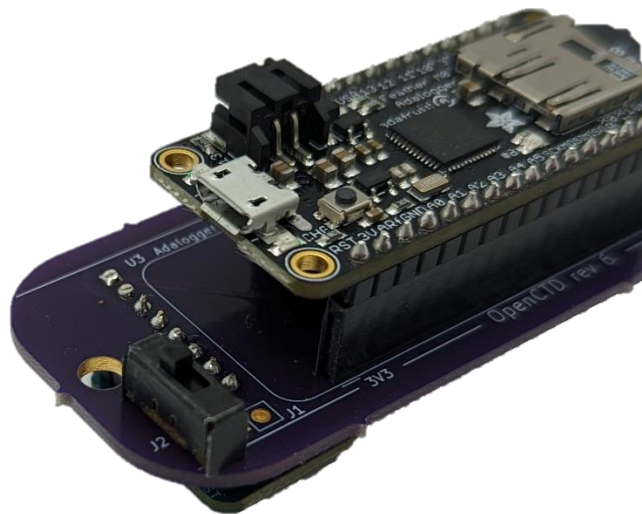
Figure 19. Closeup of carrier board highlighting switch soldered to J2 pins.

2. Snip the positive (red) lead from the battery about halfway up the connection, ensuring that there is enough space for the 2-pin JST connector to reach the battery connector on the Adalogger M0. Strip away a few millimeters of insulation on both sides of the wire.

3. Solder each end of the snipped positive lead to one of the two pins labeled J1 on the carrier board. The lead coming from the battery will be soldered to the underside of the

carrier board. The lead coming from the 2-pin JST connector will be soldered to the topside of the carrier board.
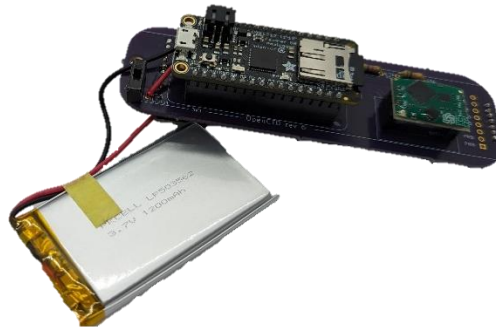

Figure 20. Carrier board with battery leads soldered to pins labeled J1.

4. Plug in the battery and ensure that the switch functions correctly.

5. Slide the battery and control board into the 3D printed chassis assembly such that the battery sits in the small compartment and is isolated from the control board. This will protect both the battery and the electronics from damage caused by short circuits or light flooding.

6. Use a large zip tie to hold the 3D printed chassis assembly closed.
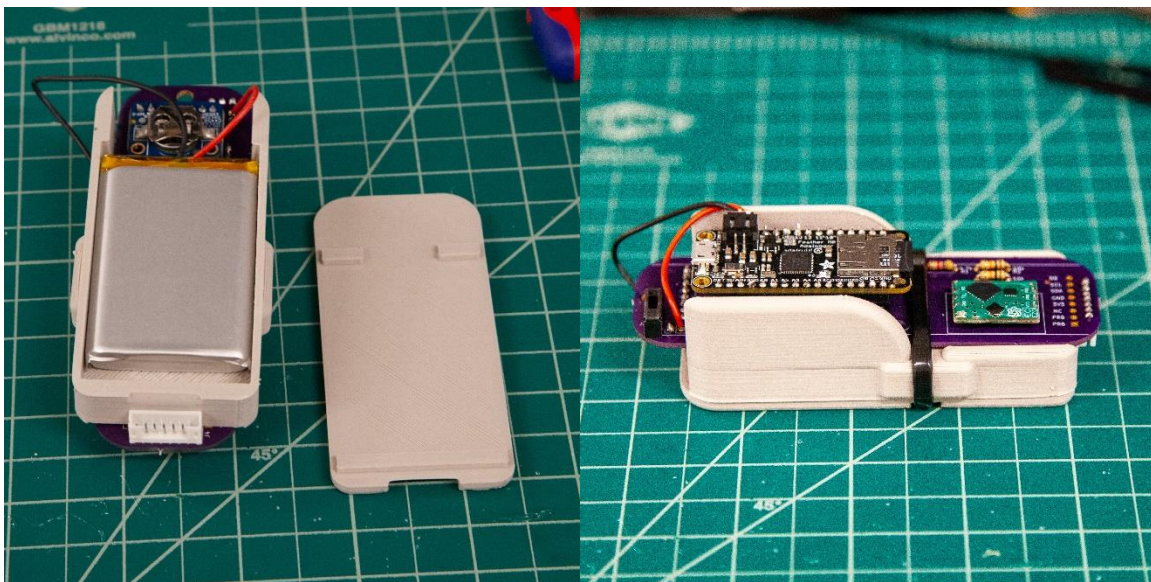
Figure 21. OpenCTD chassis showing battery resting in protective sleeve (left) and control unit secured with zip tie (right).

9 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.

9.1 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.

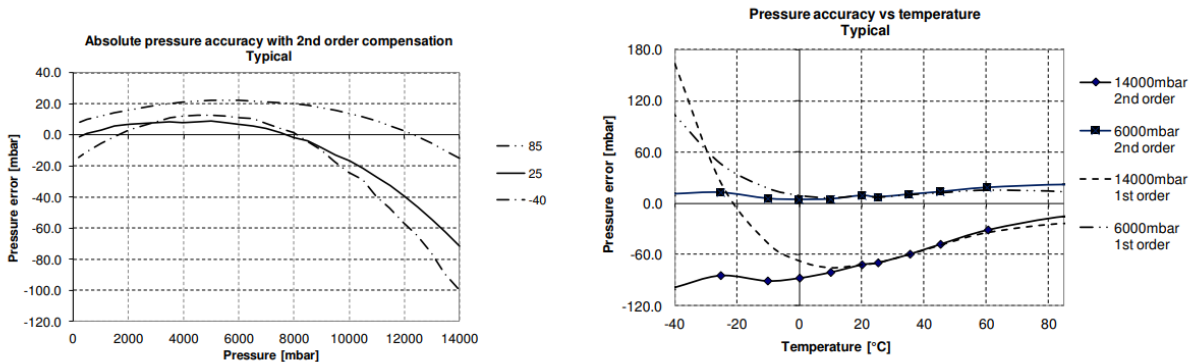### PRESSURE ERROR VS PRESSURE AND TEMPERATURE

Figure 22. 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; $P_{(DEPTH)}$ is the pressure (in millibars) at depth); $P_{(SURFACE)}$ is the pressure at the surface in millibars; and **G** is acceleration due to gravity, which for most field purposes can be assumed as 9.81 m/s^2 (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 $P_{(SURFACE)}$. For deployments in alpine lakes and other high-altitude environments, be aware that surface pressure may be significantly less than 1013.25 mbar.

9.2 Temperature

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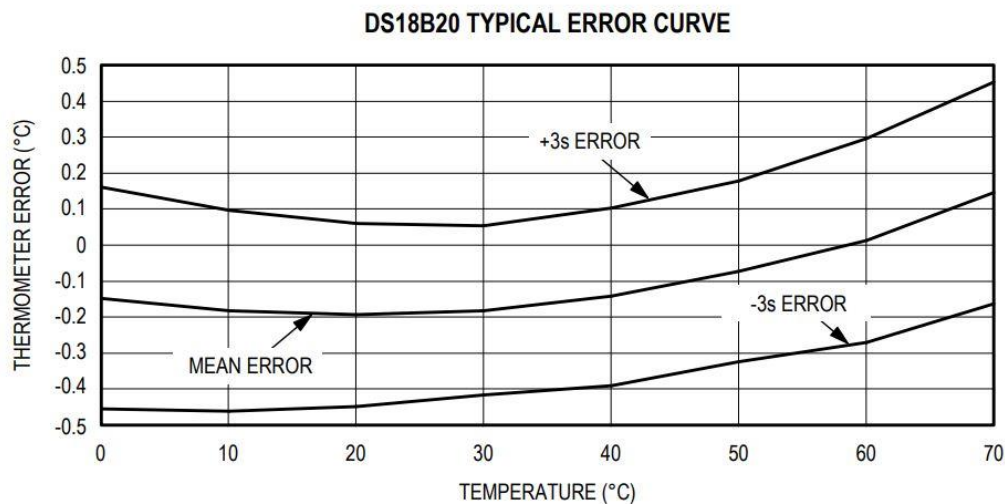

Figure 23. Typical error curve for DS18B20 digital thermometer. Provided by supplier.

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 two different designs of DS18B20 sensors in stainless steel cladding: flat-topped and hemispherical. For sensors clad in a flat top housing, the time constant was 5.7 seconds. For those clad in a hemispherical housing, it was 8.5 seconds.

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

The DS18B20 temperature sensor is factory calibrated but the user should perform a single-point or two-point calibration in cases where desired precision exceeds the error rate of ±0.1ºC. The sensor offset is, however, linear, which means that a single calibration constant can be determined to correct the accuracy of the temperature measurements averaged across three sensors. A calibration constant can be established by plotting the average deviation between the OpenCTD temperature sensors and water of known temperature.

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

9.3 Conductivity (Salinity)

The OpenCTD uses a graphite conductivity probe manufactured by Atlas Scientific. Electrical conductivity probes are relatively simple devices consisting of two electrodes of known surface area and known distance from each other. As conductive fluid passes between the two electrodes, they measure the resistance of the liquid. The Atlas probe has a response time of 90% within 1 second, a 343-meter maximum operating depth, and a 1 to 110ºC temperature range. The K 1.0 probes can measure conductivity from 5 to 200,000 µS/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 24. 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 25ºC but the Atlas EZO-EC conductivity circuit also allows for temperature compensation if maintaining a stable 25ºC 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 bottled water is produced using reverse osmosis and has no added minerals, making this brand a useful 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 25ºC.

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. Cut a hole in the bottom of a Styrofoam cooler large enough for the OpenCTD housing to fit snuggly 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.
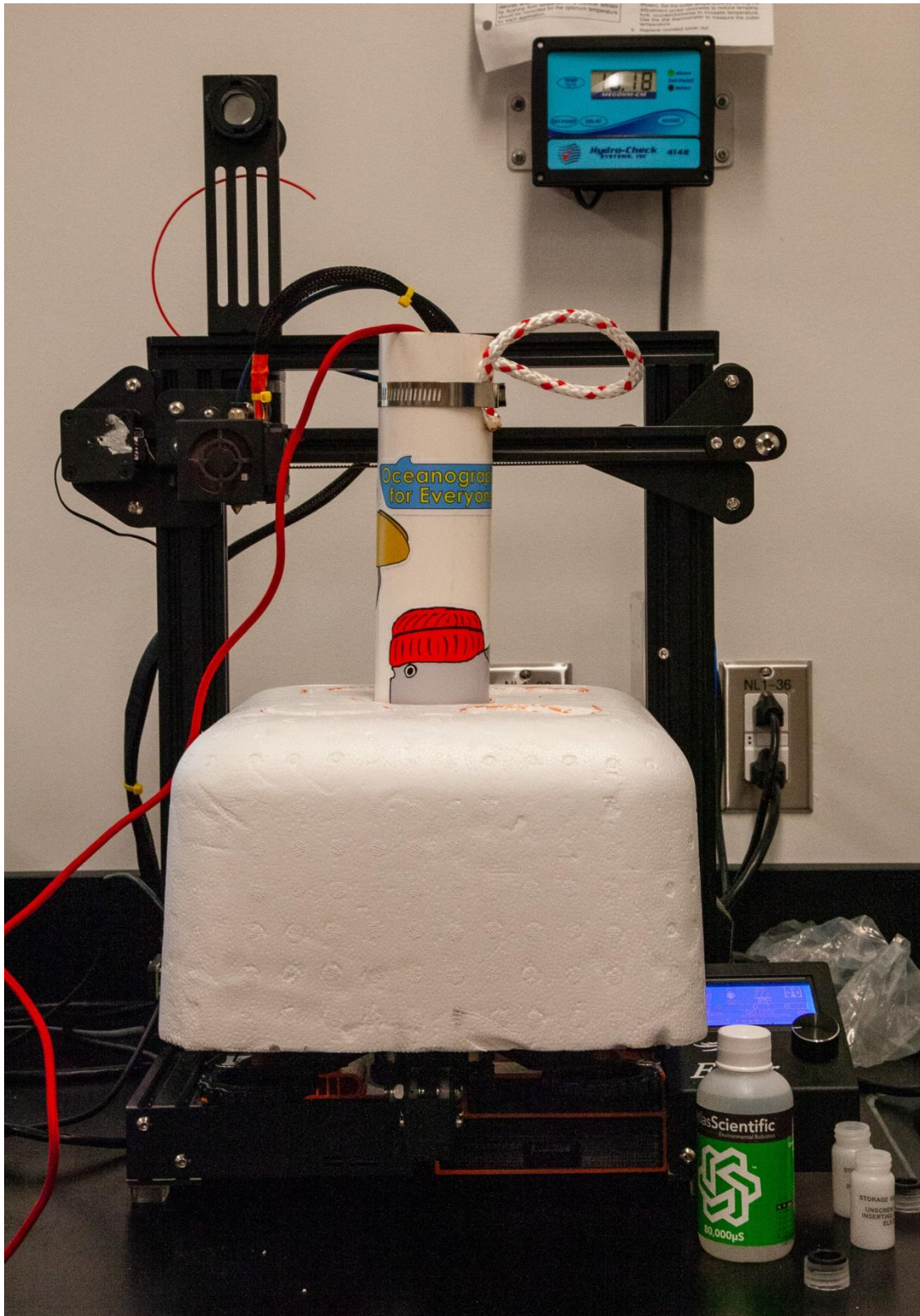
Figure 25. OpenCTD mounted inside a Styrofoam cooler on top of a 3D printer heat bed in order to maintain an ambient 25ºC temperature. Photo by Allie Wilkinson.

When you are ready to calibrate your OpenCTD, insert the device into the hole in the Styrofoam cooler such that it sits inside the coffee mug or other thermal mass, and allow it to reach equilibrium (usually 15 minutes or less).

We have determined through experimentation that even the least expensive 3D printers can maintain a very stable temperature throughout the calibration process. In cases where the ambient temperature is more than 5°C cooler than 25°C, you may want to store the conductivity solution on the heated bed as well. This process will not work for rooms where the ambient temperature is greater than 25°C. Please reference the end of this section for alternatives to temperature compensate the OpenCTD during calibration.

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) found in the OpenCTD GitHub repository. If only a single calibration standard is available, one-point calibration is possible, but not recommended. This software will also output the average temperature reading from the three temperature sensors.

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 also 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 you 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 probe storage bottle with distilled or RO water and dry with a clean paper towel.

9. Fill the probe storage bottle with the less conductive of the two conductivity solutions (12,880 µS if you're using the Atlas standard solutions). Slide the cap ring of the probe storage bottle onto the conductivity probe so that it sits well above the small hole between electrodes. Slide the probe storage bottle onto the probe and screw it into the cap so that it seals tightly, and the calibration standard completely covers the small hole between electrodes. Tap the probe gently to shake out any bubbles.
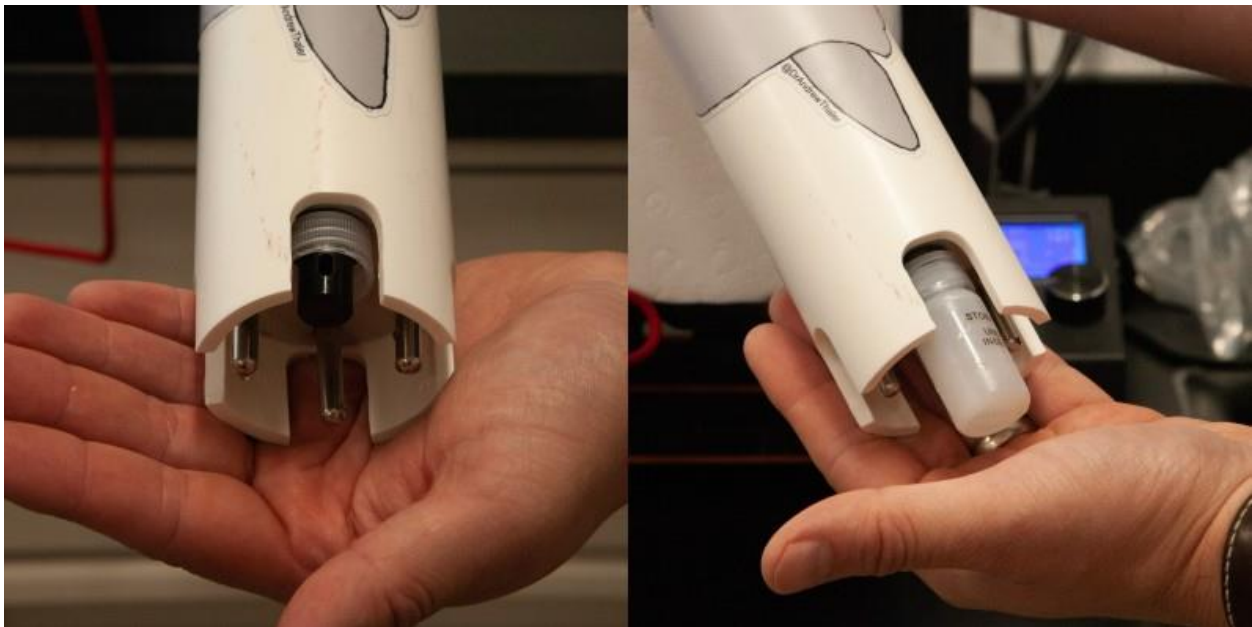

Figure 26. OpenCTD conductivity sensor with the cap ring of the probe storage bottle placed above the electrode opening (left) and with the conductivity probe completely submerged in the probe storage bottle. Photos by Allie Wilkinson.

10. Place the probe with the bottle attached 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 storage bottle, 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 probe storage bottle with distilled or RO water and dry with a clean paper towel.

13. Fill the probe storage bottle with the more conductive of the two conductivity solutions (80,000 µS if you're using the Atlas standard solutions). Slide the cap ring of the probe storage bottle onto the conductivity probe so that it sits well above the small hole between electrodes. Slide the probe storage bottle onto the probe and screw it into the cap so that it seals tightly, and the calibration standard completely covers the small hole between electrodes. Tap the probe gently to shake out any bubbles.

14. Place the probe with the bottle attached 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.

15. 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.

16. 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) for more thorough instructions.
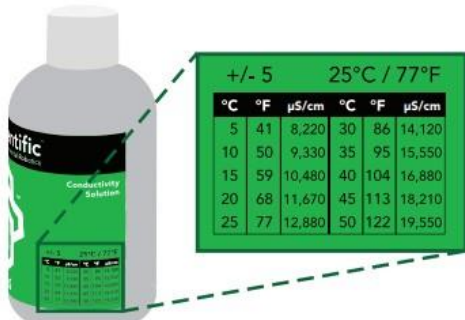
Figure 27. Location of conductivity temperature offsets. Image from Atlas Scientific.

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).

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

9.4 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 $^{\circ}$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 date 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.

10 Final Housing 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.

10.1 Casting Loop

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

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

10.2 Weighted Casting Line

For deeper deployments, or operations in high current, a weighted anchor line, with the anchor or weight held several feet below the CTD, with 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.

10.3 Internal Ballast

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

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

Figure 30. Clear OpenCTD with fishing weights inside.

10.4 External Ballast

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

preferable to mount the entire OpenCTD to a heavier object for deployment. This is a great opportunity to encourage your students to get creative.

10.5 The First Deployment

It is finally time to make your first cast!

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

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

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

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

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

While calibration plays a significant role in affecting data quality, response times and the method of deployment also have an influence. The DS18B20 clad in stainless steel has a time constant of 5.7s. The conductivity sensor maintains a response time of approximately one second for a 90% value (Atlas-Scientific, 2017). The pressure sensor response time is variable between 0.5 to 8.22

m/s depending on the resolution selected by the user (Measurement Specialties, 2012). In order to ensure that the device has reached relative equilibrium with the surrounding water, it is necessary to leave it at station for several minutes before beginning an in-situ deployment. When faster response time is required, the sensors used in the most recent version of the OpenCTD can easily be replaced by more accurate, but also more expensive alternatives.

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 back 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.

10.6 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)

11 Works Cited

Fofonoff, N.P., Millard Jr, R.C., 1983. Algorithms for the computation of fundamental properties of seawater.

Hooker, S.K., Boyd, I.L., 2003. Salinity sensors on seals: use of marine predators to carry CTD data loggers. Deep Sea Research Part I: Oceanographic Research Papers 50, 927–939. https://doi.org/10.1016/S0967-0637(03)00055-4

Lewis, E., 1980. The practical salinity scale 1978 and its antecedents. IEEE Journal of Oceanic Engineering 5, 3–8. https://doi.org/10.1109/JOE.1980.1145448

Oskui, S.M., Diamante, G., Liao, C., Shi, W., Gan, J., Schlenk, D., Grover, W.H., 2016. Assessing and Reducing the Toxicity of 3D-Printed Parts. Environ. Sci. Technol. Lett. 3, 1–6. https://doi.org/10.1021/acs.estlett.5b00249

Stammer, D., Balmaseda, M., Heimbach, P., Köhl, A., Weaver, A., 2016. Ocean Data Assimilation in Support of Climate Applications: Status and Perspectives. Annual Review of Marine Science 8, 491–518. https://doi.org/10.1146/annurev-marine-122414-034113

Stephen, R.C., Howard, F.J., Dean, R., Susan, W., Ariel, T., Mathieu, B., Denis, G., Jianping, X., Sylvie, P., Thresher, A., Traon Pierre-Yves, L., Guillaume, M., Birgit, K., Kjell-Arne, M., Velez-Belch Pedro, J., Isabelle, A., Brian, K., Jon, T., Molly, B., Steven, J.R., 2016. Fifteen years of ocean observations with the global Argo array. Nature Climate Change Marine Res Inst 6. https://doi.org/10.1038/NCLIMATE2872

Sverdrup, H.U., Johnson, M.W., Fleming, R.H., 1942. The Oceans: Their Physics, Chemistry, and General Biology. Asia Publishing House.

Thaler, A., Parsons, E.C.M., de Vos, A., Rose, N.A., Smith, C., Fretz, D., 2019. Bot Meets Whale: Best Practices for Mitigating Negative Interactions Between Marine Mammals and MicroROVs. Front. Mar. Sci. 6. https://doi.org/10.3389/fmars.2019.00506

Thaler, A.D., Freitag, A., Bergman, E., Fretz, D., Saleu, W., 2015. Robots as vectors for marine invasions: Best practices for minimizing transmission of invasive species via observation-class ROVs. Tropical Conservation Science 8.

Thomson, R.E., Emery, W.J., 2014. Data Analysis Methods in Physical Oceanography. Newnes.

12 Resources and Datasheets

General Guides

Soldering is Easy: Here's How To Do It:
mightyohm.com/files/soldercomic/FullSolderComic_EN.pdf
Environmental Monitoring with Arduino: Building Simple Devices to Collect Data About
the World Around Us: https://amzn.to/2M17yUi

Pressure Sensor

MS5803-14BA Miniature 14 bar Module:
https://cdn.sparkfun.com/datasheets/Sensors/Weather/ms5803_14ba.pdf

Temperature Sensor

DS18B20 1-Wire Digital Thermometer:
https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf

Conductivity Sensor

EZO-EC Conductivity Circuit Datasheet – https://www.atlas-
scientific.com/_files/_datasheets/_circuit/EC_EZO_Datasheet.pdf
EC K 1.0 Conductivity Probe Datasheet – https://www.atlas-
scientific.com/_files/_datasheets/_probe/EC_K_1.0_probe.pdf
Conductivity Accuracy Graph – https://www.atlas-
scientific.com/_files/instructions/conductivity_accuracy_graph.pdf

13 Bill of Materials

	Item	Links	Price	Price per unit
Housing	2" Schedule 40 PVC	https://amzn.to/2JZnNAm Check local hardware stores	$11.60	$5.80
	Oatey/Cherne 270229 Mechanical Test Plug	https://amzn.to/2WLCcS7 Check local hardware stores	$7.88	$7.88
	Hose clamp (AB-36/46mm to 70mm)	https://amzn.to/3t7G1UX Check local hardware stores	$12.49	$2.50
	1/4" PolyPro Rope	https://amzn.to/3w7su1G Check local hardware stores	$6.11	$0.12
OpenCTD Custom Carrier Board	Custom Carrier Board	Available from Oceanography for Everyone	$9.37	$9.37
	Adafruit Feather M0 Adalogger	https://www.adafruit.com/product/2796 https://amzn.to/36DrCm4 https://www.digikey.com/en/products/detail/adafruit-industries-llc/2796/5804105	$19.95	$19.95
	SanDisk Industrial 8GB Micro SD Memory Card	https://amzn.to/46XyjhP Check local electronics stores	$37.72	$7.54
	Adafruit DS3231 Precision RTC Breakout	https://www.adafruit.com/product/3013	$17.50	$17.50
	CR1220 3V Coin Cell Battery	https://amzn.to/2NYFQpf https://www.adafruit.com/product/380 Check local electronics stores	$5.99	$0.60
	Atlas Scientific EZO Conductivity Circuit	https://www.atlas-scientific.com/product_pages/circuits/ezo_ec.html https://amzn.to/40aT5aN	$67.99	$67.99
	Short Feather Female Headers	https://www.adafruit.com/product/2940 Check local electronics stores	$1.50	$1.50

Short Feather Male Headers	https://www.adafruit.com/product/3002 Check local electronics stores	$0.50	$0.50
7-pin JST Header/Cable Bundle	https://amzn.to/35Uvh3i Check local electronics stores	$13.99	$1.75
MicroUSB Cable	https://amzn.to/3CEElp0 Check local electronics stores	$6.76	$6.76
10 kΩ Resistor (2)	https://amzn.to/3JcUFjk Check local electronics stores	$6.75	$0.14
4.7 kΩ Resistor (1)	https://amzn.to/3JdZhpt Check local electronics stores	$6.16	$0.06

	DS18B20 Temperature Sensor in Stainless Steel Housing	https://amzn.to/2YOevug	$11.58	$6.95
	(Preferred) Atlas Scientific K 1.0 Mini Conductivity Probe	https://www.atlas-scientific.com/product_pages/probes/ec_k1-0-mini.html https://amzn.to/493Pbon	$109.99	$109.99
Sensors	(Alternative) Atlas Scientific K 1.0 Conductivity Probe	https://www.atlas-scientific.com/product_pages/probes/ec_k1-0.html https://amzn.to/40d3Cma	$139.99	$139.99
	MS5803-14BA Pressure Sensor Breakout	https://www.sparkfun.com/products/12909 https://amzn.to/2rfyEgA	$64.50	$64.50

	Lithium Ion Battery Pack 3.7V 2000 mAh	https://www.adafruit.com/product/2011 https://amzn.to/3QtHWP8 Check local electronics stores	$12.50	$12.50
Power and Ballast	(Internal Switch) Board-mounted switch	https://www.adafruit.com/product/805 Check local electronics stores	$0.95	$0.95
	(External Switch) 10mm X 3mm Round Rare Earth Magnet (2)	https://amzn.to/3Ifpbri	$5.99	$0.40

	(External Switch) 10mm round double-sided adhesive mounting dots (2)	https://amzn.to/2OdU85B	$8.98	$0.02
	3-ounce fishing weights	https://amzn.to/3CI46Vy Check any fishing store	$12.99	$2.60
Consumables	Polymaker Polyterra Matte PLA Filament 1.75mm	https://amzn.to/3Js8XMU https://us.polymaker.com/products/polyterra-pla	$19.99	$19.99
	Lead Free Solder Wire	https://amzn.to/3Qa7O1e Check local electronics stores	$6.99	$6.99
	3M Scotch #35 Electrical Tape	https://amzn.to/2QeA9rS Check local electronics stores	$11.99	$2.40
	22-Gauge Silicone Wire	https://amzn.to/3Fy8Uz0 Check local electronics stores	$25.98	$18.19
	Solder Seal Wire Connectors AWG 26-24	https://amzn.to/498dKAw Check local electronics stores	$9.99	$0.80
	Multi-Colored Heat Shrink Pack	https://www.adafruit.com/product/1649 Check local electronics stores	$4.95	$4.95
	3M Garnet Sandpaper, 9-Inch by 11-Inch	https://amzn.to/3vCfaQ9 Check local electronics stores	$5.19	$5.19
	LOCTITE Epoxy Adhesive: E-120HP	https://www.grainger.com/product/4UK12	$29.14	$29.14
	3M Static Mixing Nozzle, 38193, 50 mL	https://amzn.to/30YJB4N	$21.99	$1.83
	Loctite Epoxy Five Minute Instant Mix	https://amzn.to/3tuKM8q Check local hardware stores	$11.95	$5.96
	Conductivity Calibration K 1.0 Set	https://atlas-scientific.com/calibration-solutions/conductivity-calibration-k-1-0-set/	$13.49	$13.49

Alternative Build Pathways

Adalogger + RTC Featherwing	Adafruit Feather M0 Adalogger	https://www.adafruit.com/product/2796 https://amzn.to/36DrCm4	$19.95	$19.95
	Adafruit DS3231 Precision RTC FeatherWing	https://www.adafruit.com/product/3028 https://amzn.to/2PRHmvS	$13.95	$13.95
	Short Feather Female Headers	https://www.adafruit.com/product/2940 Can be sourced from any electronics supplier	$1.50	$1.50
	Short Feather Male Headers	https://www.adafruit.com/product/3002 Can be sourced from any electronics supplier	$0.50	$0.50
	Adafruit Stacking Headers	https://www.adafruit.com/product/2830	$1.25	$1.25
	MicroUSB Cable	https://amzn.to/3CEElp0 Can be sourced from any electronics supplier	$6.81	$6.81
	CR1220 3V Coin Cell Battery	https://amzn.to/2NYFQpf https://www.adafruit.com/product/380	$4.75	$0.48
	SanDisk MicroSD Memory Card	https://amzn.to/2FTkfKW	$6.95	$6.95
Protoboard Control Unit	Adafruit Perma-Proto Small Breadboard PCB	https://www.adafruit.com/product/1214	$7.95	$2.65
	Atlas Scientific EZO Conductivity Circuit	https://www.atlas-scientific.com/product_pages/circuits/ezo_ec.html https://amzn.to/2K0Xk5z	$67.99	$67.99
	Right Angle Female PCB Header	https://amzn.to/2ICUOki Can be sourced from any electronics supplier	$9.99	$1.00
	Solid Core Wire	https://www.adafruit.com/product/290 Can be sourced from any electronics supplier	$2.95	$0.12

	10 kΩ Resistor (2)	https://amzn.to/3JcUFjk Can be sourced from any electronics supplier	$6.18	$0.12
	4.7 kΩ Resistor (1)	https://amzn.to/3JdZhpt Can be sourced from any electronics supplier	$6.17	$0.06
DIY Pressure Sensor	MS5803-14BA Pressure Sensor SMC	https://www.digikey.com/en/products/detail/te-connectivity-measurement-specialties/MS580314BA01-00/5277631 https://www.mouser.com/ProductDetail/Measurement-Specialties/MS580314BA01-00 https://www.tti.com/content/ttiinc/en/apps/part-detail.html?partsNumber=MS580314BA01-00	$34.29	$34.29
	SOIC to DIP Adapter	https://www.adafruit.com/product/1212	$2.95	$0.41
	10 kΩ Resistor	https://amzn.to/3JcUFjk Can be sourced from any electronics supplier	$6.18	$0.06
	0.1 uF Capacitor	https://amzn.to/32tMBol Can be sourced from any electronics supplier	$5.92	$0.24

Appendix A. Alternate Control Unit

Components for Adafruit Adalogger M0 and Real-time Clock Featherwing

Adafruit Adalogger M0

Adafruit Real-time Clock Featherwing

Male Headers

Female Headers
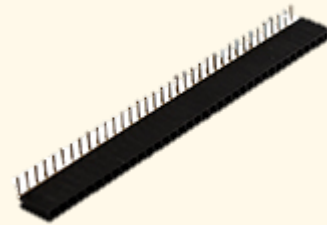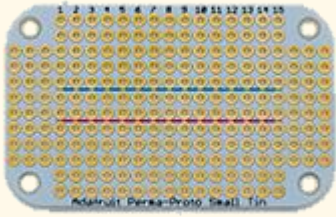
Extra Long Headers

MicroUSB Cable

3V CR 1220 Coin Cell Battery

SD Card

Components for Protoboard Control Unit

Atlas EZO Conductivity Circuit   Right angle headers

Solid core wire   *10 kOhm resistor*   *4.7 kOhm Resistor*

Preparing the Adalogger and Featherwing RTC for use with a protoboard

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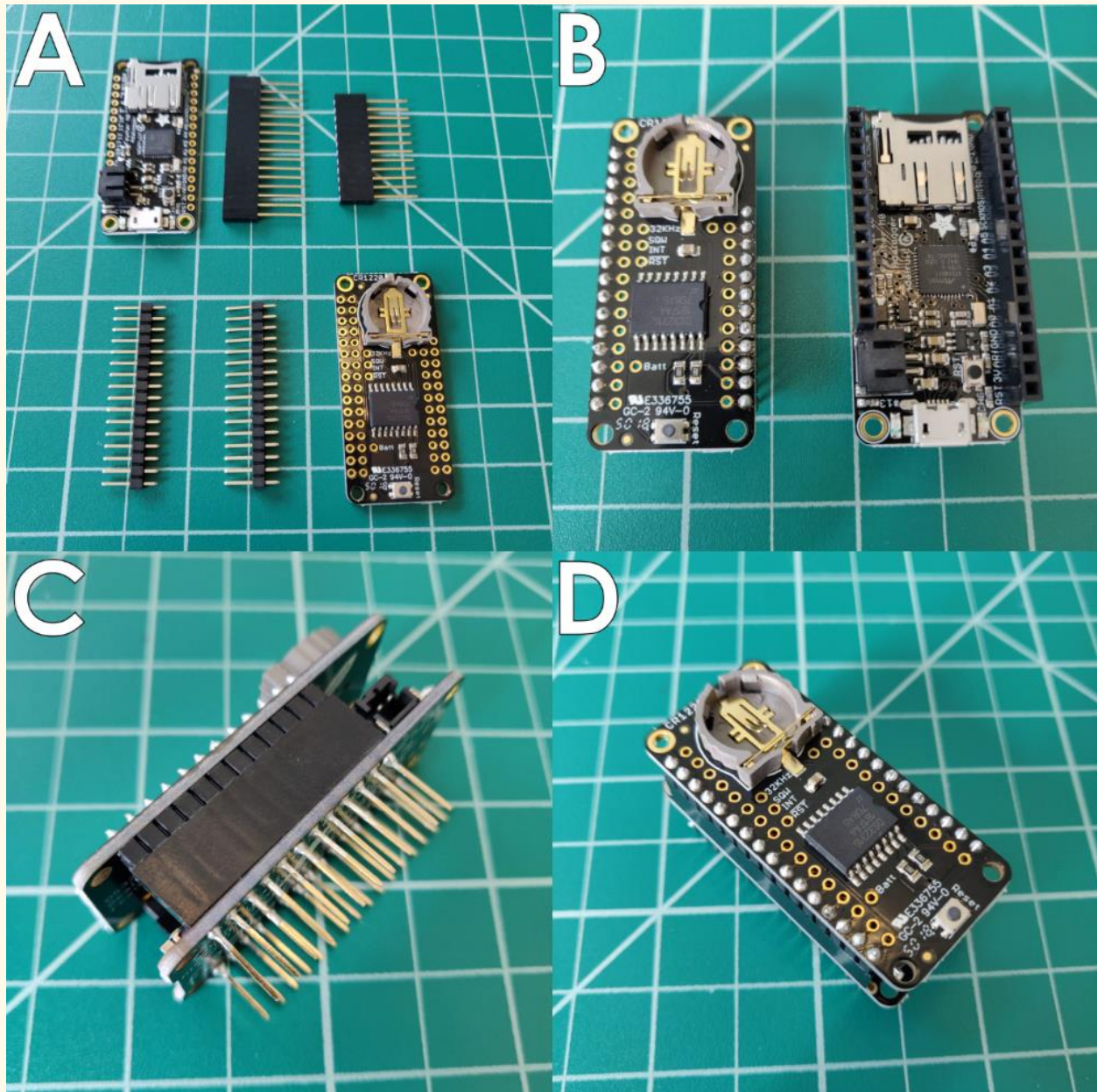

Figure 31. Adalogger and Real-time Clock with associated headers (A); after soldering (B); connected in profile (C); connected from above (D).

1. Slot the stacking headers into the through-holes on the Adalogger, ensuring that the number of headers matches the number of available pins. The female

end of the stacking headers should face up. If you are using generic stacking headers, cut them to length with flush cutters so that the number of posts matches the number of pins on each side.

2. Flip the Adalogger over so that male headers are facing up. Ensure that the headers are square and perpendicular to the face of the Adalogger.

3. Solder the four corner pins, double check that the headers are straight, and then solder the remaining pins.

4. Slot the male headers into the outer pins of the RTC such that the short ends pass through the board and the long header pins face down. If you are using generic stacking headers, cut them to length with flush cutters so that the number of posts matches the number of pins on each side.

5. Solder the four corner pins, double check that the male headers are straight and aligned with female headers on the Adalogger, and then solder the remaining pins.

6. Stack the RTC on top of the Adalogger by lining up the male, downward-facing header pins on the RTC with the female, upward facing header pins on the Adalogger.

7. Insert the CR1220 battery into the RTC.

Benchmark: Breadboarding the Protoboard Circuit

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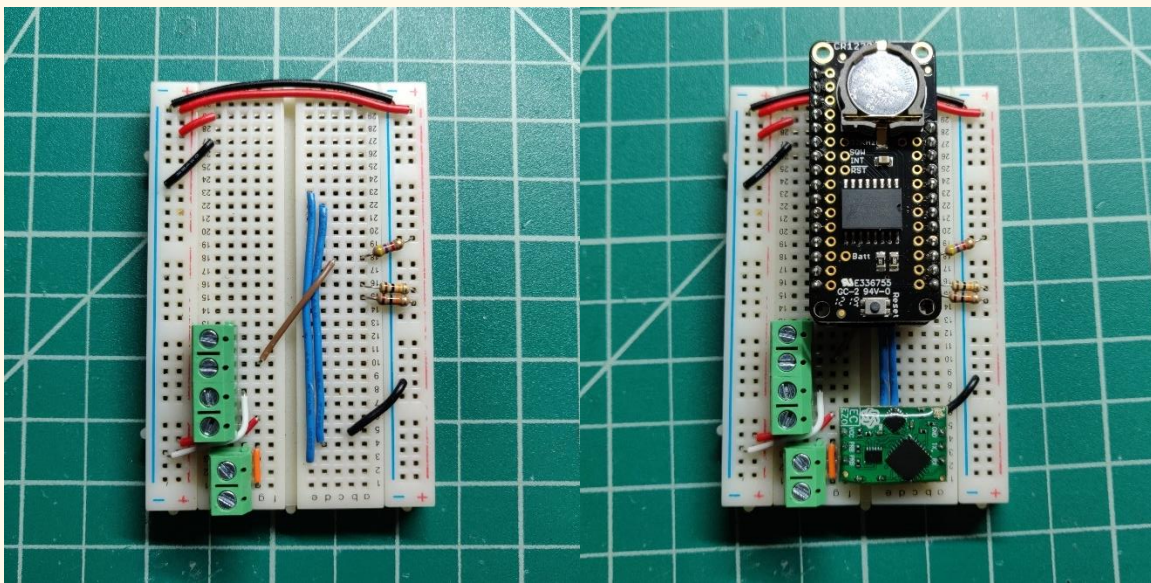

Figure 32. Breadboard with just jumper wires, screw terminals, and resistors attached and with Adalogger/RTC and EZO conductivity circuit.

1. Reference Figure 9 and the photo and pinout guide below. Place Atlas EZO circuit and connect jumper wires in accordance.

2. Connect a 4.7 kΩ resistor between pin D6 and the positive voltage rail.

3. Connect a 10 kΩ resistor between the SDA pin and the positive voltage rail and connect another 10 kΩ resistor between SCL pins and the positive voltage rail.

4. Connect the temperature sensors to digital pin 6, positive, and ground using the screw terminal blocks.

5. Connect the pressure sensor to SCL, SDA, 3.3V, and GND. The solid core wire on the pressure sensor can be loosely slotted directly into the pin holes on the RTC.

6. Connect the temperature probe to the two probe pins on the Atlas EZO using the screw terminal blocks or BNC connector. It does not matter which wire goes to which probe pin. Make sure the conductivity probe itself is submerged or it will read zero.

7. Connect the Adalogger to your PC using a micro-USB cable and open the serial monitor in Arduino IDE.

8. Power the Adalogger down, remove the SD card, and read the data file.

Breadboard Diagram

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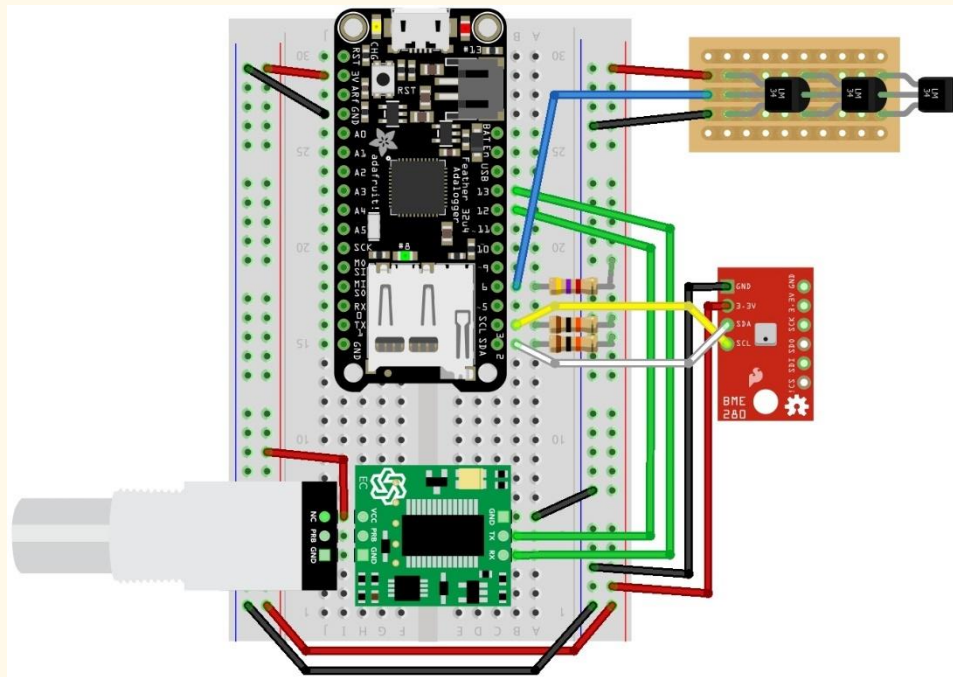

Figure 33. Diagram of breadboard assembly to test completed sensors. Note the presence of BNC connector for the conductivity probe and the use of a barometric pressure sensor to stand in for the OpenCTD pressure sensor.

Checklist

o Atlas EZO circuit is blinking with alternating between green and blue lights.

o Temperature probes are reading ambient temperature and are within 0.5°C of each other.

o Pressure sensor is reading atmospheric pressure (it should be around 1012 mbar if you're near sea level).

o Conductivity sensor is reading the conductivity of whatever fluid it is in.

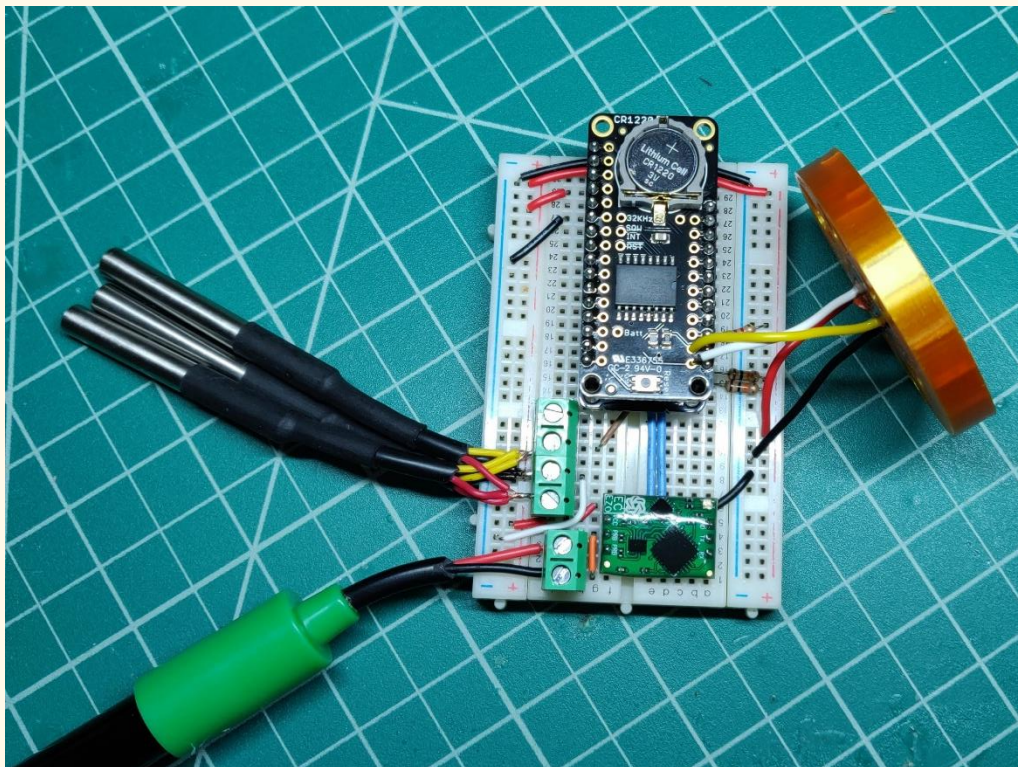o Data had been logged to SD Card.

Figure 34. Completed breadboard set-up with all sensors ready for testing.

Assembling the Protoboard Control Unit

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

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

1. Once the Adalogger has been tested, it no longer needs the long pins that connected it to the breadboard. Snip these pins off at the base of the Adalogger. The Adalogger should now sit flush inside the Electronics Chassis.

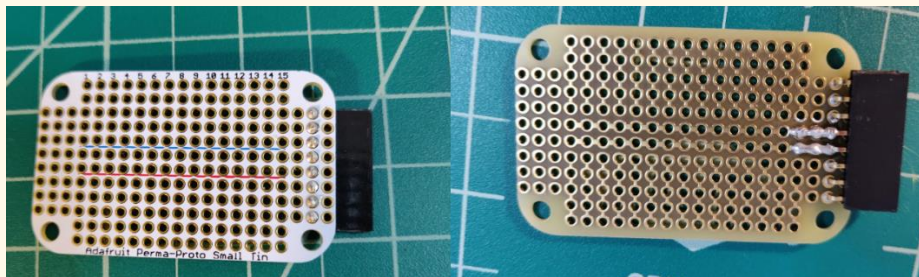2. Cut an 8-pin length from a right-angle female header strip.

Figure 35. Perma-Protoboard with 8-pin connector top (A) and bottom, with solder bridges (B).

3. Solder the 8-pin connector to the second to last set of through-holes on the right side of the Perma-Protoboard, on the underside such that the pins emerge through the middle column of unconnected through-holes.

4. On the underside of the board, build a solder bridge between the positive and negative strips and their respective, adjacent connector pins. This works best with a soldering iron heated to 750ºC. Hotter irons will cause the surrounding solder to melt.

Powering the Protoboard

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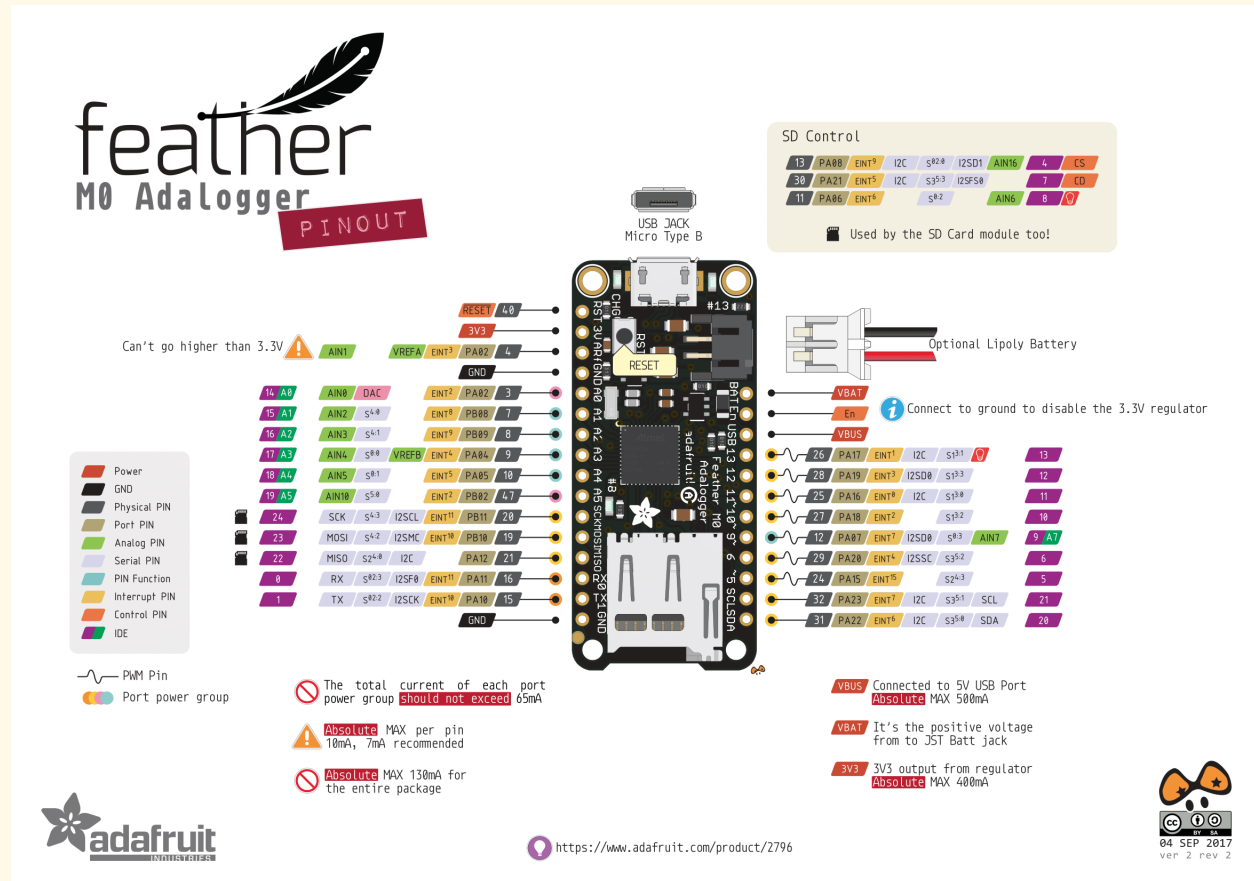

Figure 36. Pinouts for the Adalogger Feather M0. Note, these pinouts also correspond to those of the RTC.

1. Connect red and black solid core wire to the 3.3V and GND pins on the left side of the RTC.

2. Place the complete Adalogger/RTC assembly into the Electronics Chassis and stretch these two wires until the meet the end of the Chassis.

3. Cut the wires near the far end of the zip-tie hole in the e-chassis and strip off half a centimeter of insulation from the leads.

4. Solder these wires to the positive (3.3V) and negative (GND) rails of the Perma-Protoboard along columns 5 or 6.

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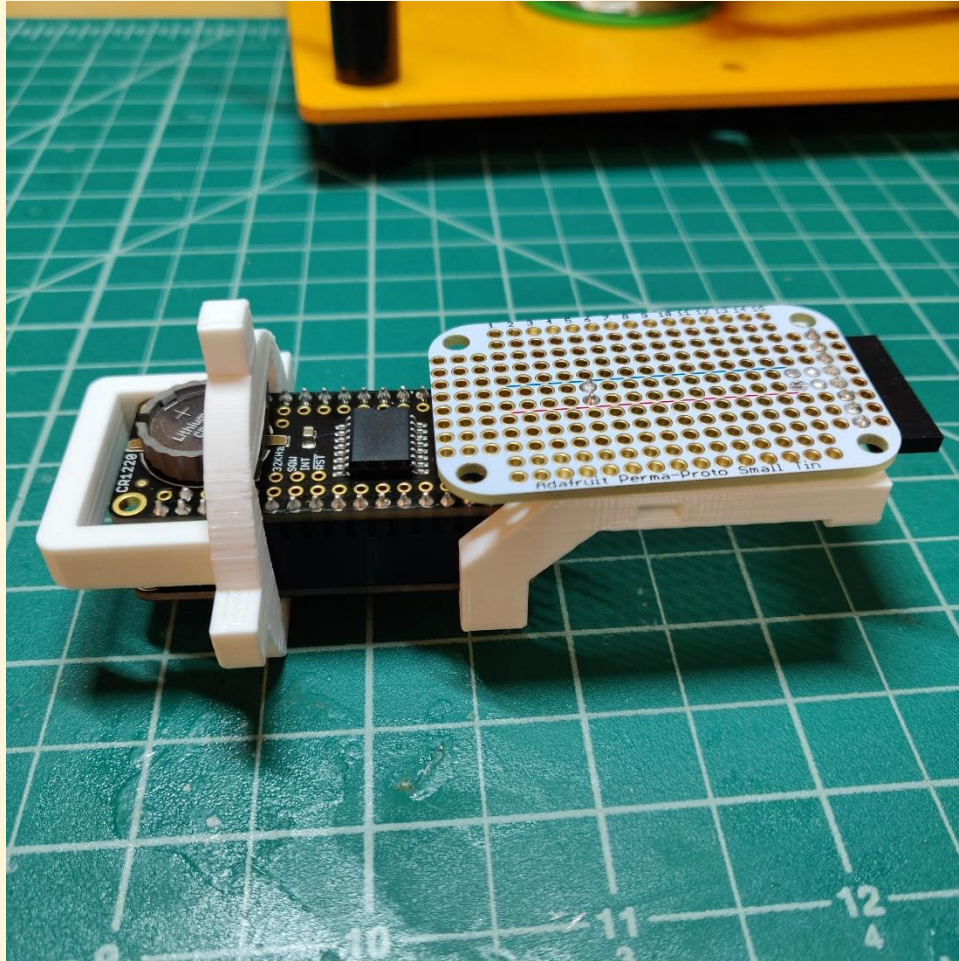

Figure 37. The Adalogger/RTC fit into the electronic chassis.

Populating the Perma-Protoboard

Skills: Soldering
Consumables: Solder, solid-core wire
Parts: Adalogger/RTC connected to Adafruit Perma-proto board, Atlas EZO circuit, 10 kOhm pull-up resistor (x2), 4.7 kOhm pull-up resistor

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

1. Solder the Atlas EZO circuit to columns 1 through 3 on the Perma-protoboard such that the GND and VCC pins correspond to the correct power rails. Holding the board so that "Adafruit Perma-Proto Small Tin" is correctly oriented, GND/TX/RX should be on top and VCC/PRB/PRB should be below.

2. On the underside of the Perma-proto board, bridge the GND pin of the Atlas EZO circuit to the negative rail with solder.

3. On the underside of the Perma-proto board, bridge the VCC pin of the Atlas EZO circuit to the positive rail with solder.

4. Using solid core wire, connect pin 12 on the RTC to TX on the Atlas EZO.

5. Using solid core wire, connect pin 13 on the RTC to RX on the Atlas EZO.

6. Connect each PRB pin to the through holes adjacent to pins 7 and 8 on the 8-pin right-angle header. It does not matter which PRB pin connects to which of the two header pins.

7. Connect the 4.7 kOhm pull-up resistor to column 8 and the positive (red) rail.

8. Connect a 10 kOhm pull-up resistor to column 10 and the positive (red) rail.

9. Connect a 10 kOhm pull-up resistor to column 11 and the positive (red) rail.

10. Using solid core wire, connect pin 6 on the RTC to upper column 8 on the Perma-proto board.

11. Using solid core wire, connect the SCL pin on the RTC to upper column 10 on the Perma-proto board.

12. Using solid core wire, connect the SDA pin on the RTC to upper column 11 on the Perma-proto board.

13. Using solid core wire, connect upper column 8 on the Perma-proto board to the through hole adjacent to pin 1 on the 8-pin right-angle header.

14. Using solid core wire, connect upper column 10 on the Perma-proto board to the through hole adjacent to pin 2 on the 8-pin right-angle header.

15. Using solid core wire, connect upper column 11 on the Perma-proto board to the through hole adjacent to pin 3 on the 8-pin right-angle header.

16. Starting from the inside pins closest to the positive and negative rails, bridge the 8-pin right-angle header pins to the respective adjacent solid core wires on the perma-proto board with solder.

17. Clip off the one unused pin from the male end of the sensor connector and slot it into the unused female header on the protoboard control unit. This will prevent the connector from being inserted into the protoboard control unit backwards, causing a short.

18. Using zip ties, attach the Adalogger/RTC and Perma-proto board to the 3D-printed electronics chassis.
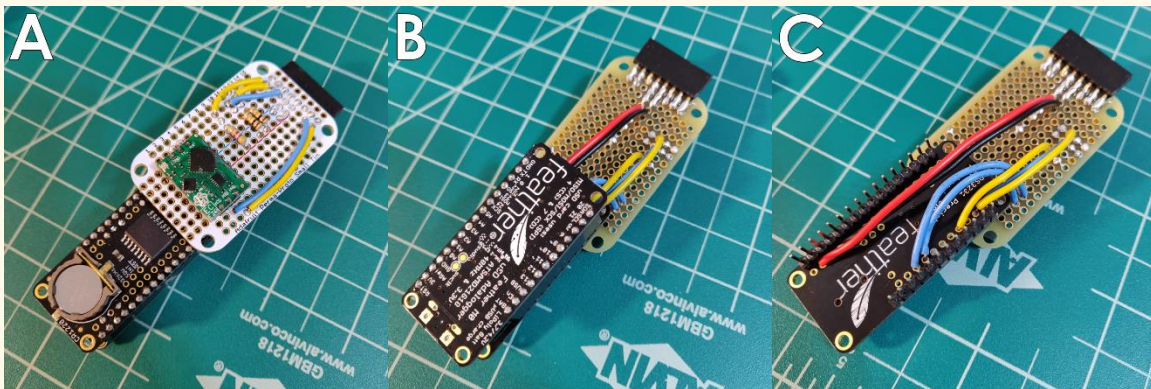

Figure 38. Fully connected CONTROL UNIT shown from the top (A), bottom (B), and bottom but without Adalogger attached (C).
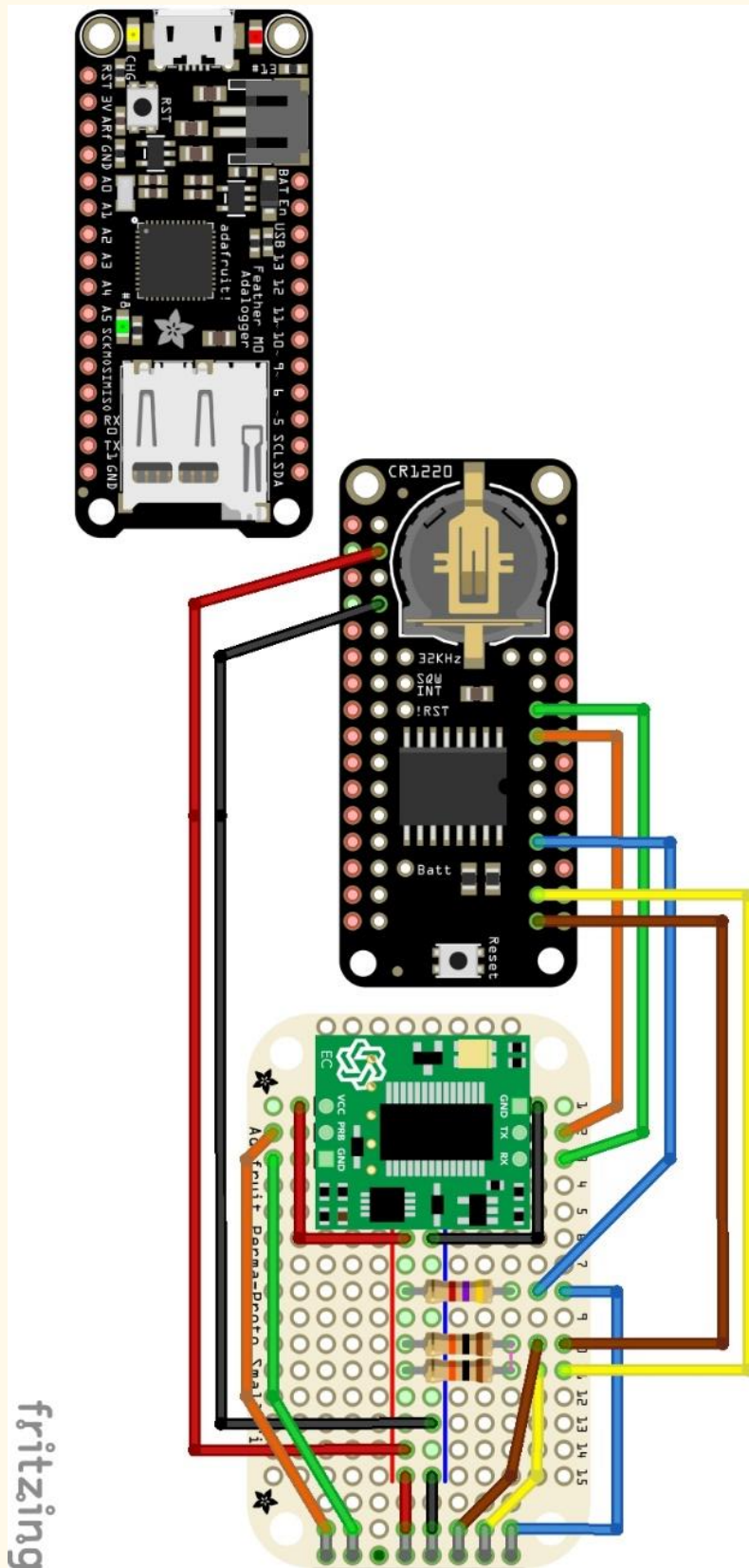
Figure 39. Layout of connections on the Perma-Protoboard.

Benchmark: Testing the Protoboard Control Unit

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

1. Holding the Perma-Protoboard face up with the connector on the right, the order of pinouts is, from top to bottom, D6 (Temperature), SCL (Pressure), SDA (Pressure), Ground, 3.3 Volt, UNUSED, Probe (-), Probe (+). Note that, as the conductivity probe functions like a resistor, it does not matter which line from the probe connects to which of the two pins.

2. Using alligator clips and jumper cables, of by soldering temporary headers onto each sensor lead, connect each sensor to its corresponding pin on the control unit. .

3. Connect the Adalogger to your computer and check the serial monitor. If all sensors are reading accurately, you're ready to prepare and pot the sensors.

Checklist

o   Protoboard control unit can read pressure sensor when connected.

o   Protoboard control unit can read temperature sensors when connected.

o   Protoboard control unit can read conductivity sensor when connected.

Creating ad-hoc connector

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

1. Solder a 5 cm length of 20-gauge wire to the positive (VCC) and ground (GND) wires of the pressure sensors and seal the connection with heat shrink tubing.

2. Solder the positive wires from the pressure and temperature sensors together, add a 15 cm length of wire to each joined end and seal with heat shrink.

3. Solder the ground wires from the pressure and temperature sensors together, add a 15 cm length of wire to each joined end and seal with heat shrink.

4. Attach a 25 cm length of 20-gauge wire to the SDA and SCL wires of the pressure sensor and seal with heat shrink.

5. Attach a 15 cm length of 20-gauge wire to each of the two wires of the conductivity probe and seal with heat shrink.

6. All wires should extend about 1 cm beyond the edge of PVC pipe when the baseplate is dry fit into position. Cut off any excess so that all wires are of equal length. Use electrical tape to tape the free wires to the side of the conductivity probe. This will relieve strain on the solder joins.

7. Strip and tin the ends of the sensor wires. Cut the braided sleeve to length and slide down over the sensor wires. This will provide additional protection to the wires and solder joins.

8. Prepare an 8-pin header by tinning the short ends of each pin.

9. Solder the sensor wires to an 8-pin header. They will align with their respective contacts on the Control Unit 8-pin female output. Take care not to apply too much heat to the header or the plastic strip will melt. Seal each pin with heat shrink tubing.
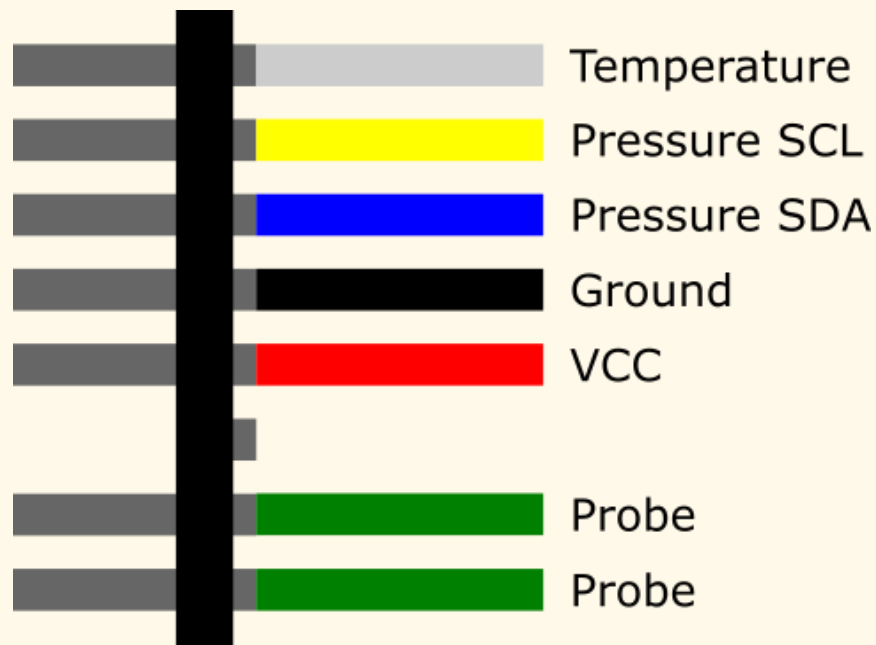
Figure 40. Illustration of connector with wires coming up from the housing. Wires leading to sensors on right, vertical black bar represents 8-pin headers. Pins leading to CONTROL UNIT are on the left.

10. Test the connector by plugging it into the breadboard and reading the sensors. You will have to rearrange the breadboard in order to connect to all seven leads. You can also test each sensor independently using jumper wires.

11. Slide the 3D-printed connector cover over the 8-pin connector.

12. Fill the entire void within the connector with 5-minute epoxy. This prevents strain on the header and solder joints as well as prevent moisture from entering the connector and prevents short circuits. Allow the glue to cure for 30 minutes.

13. Once cured, test the connection again by plugging it into the breadboard and reading the sensors. You will have to rearrange the breadboard in order to connect to all seven leads. You can also test each sensor independently using jumper wires.

Protoboard Alternative Power Switch

1. Adhere the 3-pin toggle switch to the associated slot in the back of the 3D-printed electronics chassis.

2. Snip the positive lead from the battery about halfway up the connection, ensuring that there is enough space for the 2-pin JST connector to reach the battery connector on the Adalogger M0.

3. Solder each end of the snipped positive lead to one of the two pins on the chassis, ensuring that one of the leads is soldered to the middle pin. And seal each connection with heat shrink.

4. Plug in the battery and ensure that the switch functions correctly.

5. Tape the battery to the back of the chassis in a convenient position for accessing the SD card.

Appendix B. Alternate Pressure Sensors

Components for Board Mounted Pressure Sensor

DIP to SOIC Adapter

Pressure Chip on DOIC to SOIC Adapter

10 kOhm Resistor

0.1 uF Capacitor

Components for OpenROV 30-bar IMU

OpenROV 30bar IMU

Baseplate for 30 bar IMU

DIY Pressure Sensor

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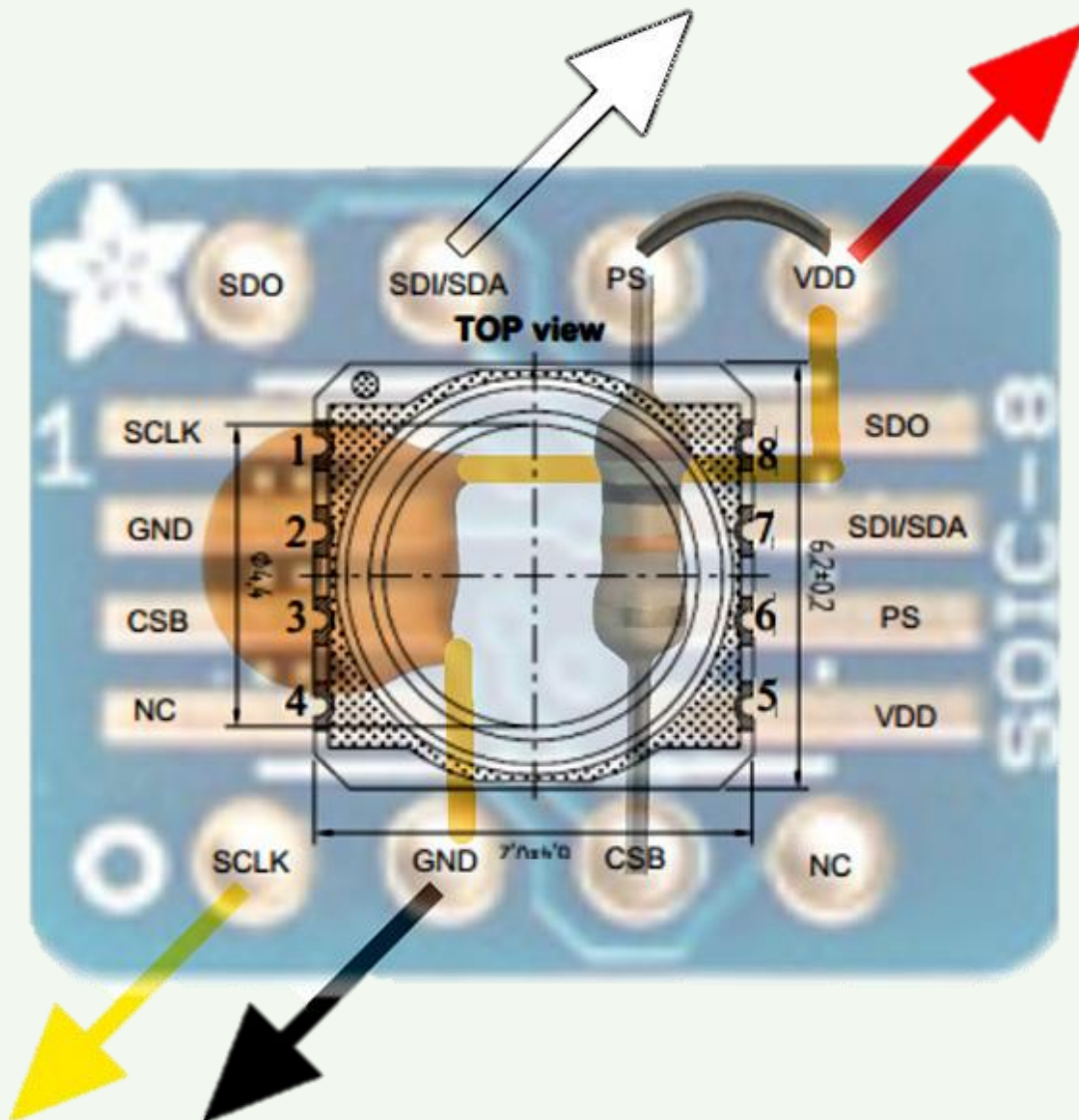

Figure 41. Diagram of pressure sensor chip and component alignment on SOIC-8 to DIP adapter. Note alignment of pin 1 on the chip to the alignment indicator (in this case, the Adafruit star, but on most boards simply a dot). Image courtesy of the Cave Pearl Project, who have an excellent guide to using and soldering these little chips here: https://thecavepearlproject.org/2014/03/27/adding-a-ms5803-02-high-resolution-pressure-sensor/).

1. Make sure that side of the SOIC8-to-DIP adapter with the larger pads is facing up. Add a small amount of solder to each pad. If you have solder rosin, add a dab of rosin to each pad to help the solder flow.

2. Ensure that the dot and notch on the pressure sensor line up with the dot on the SOIC-to-DIP adapter. Solder the sensor in place by applying a small amount of solder to the vertical grooves and drawing it down onto the solder pads. Be careful not to short any connections.

3. Bridge the pads for pin 5 (VDD/VCC) and pin 6 (PS) by drawing a small drop of solder between them.

4. Flip over the adapter and solder a 10kOhm pull-up resistor between pin 3 and 6.

5. Insert the 100nF (104) capacitor between pins 2 and 5 but don't solder yet. Bend the capacitor's legs to hold it in place.

6. Solder a 6 cm length of solid core wire to pin 1 (SCL) and pin 7 (SDA). Keep track of which wire goes to which pin by color-coding each pin (we recommend red for VCC; black for ground; yellow for SCL; and blue for SDA).

7. Solder a 6 cm length of solid core wire to pin 2 (GND) and pin 5 (VDD/VCC) such that the legs of the capacitor are captured in the solder join. You may have to twist the wire a bit to get both the wire and the capacitor leg to fit in a single pin hole. Keep track of which wire goes to which pin by color-coding each pin (we recommend red for VCC; black for ground; yellow for SCL; and blue for SDA).

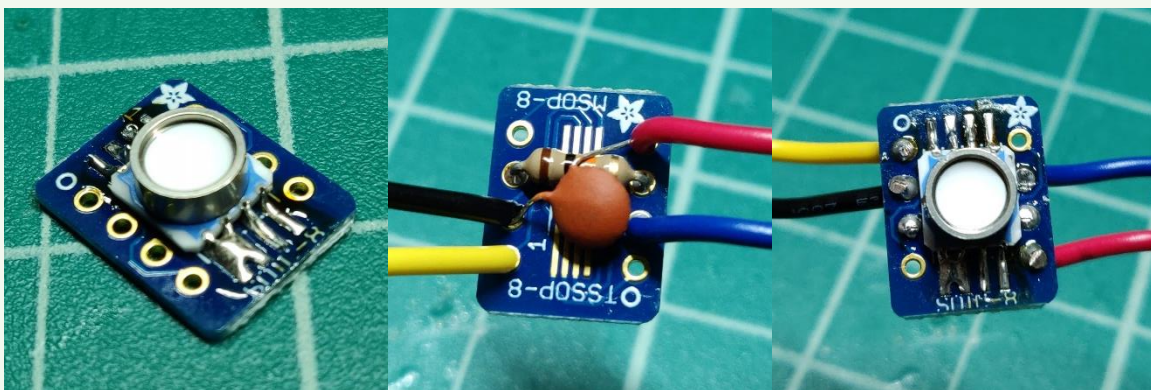8. Strip ½ cm of insulation from the exposed ends of each wire.

Figure 42. Building the Pressure Sensor unit. A: Components of the sensor. B: Surface mounting the sensor. C: Wiring up the components.

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

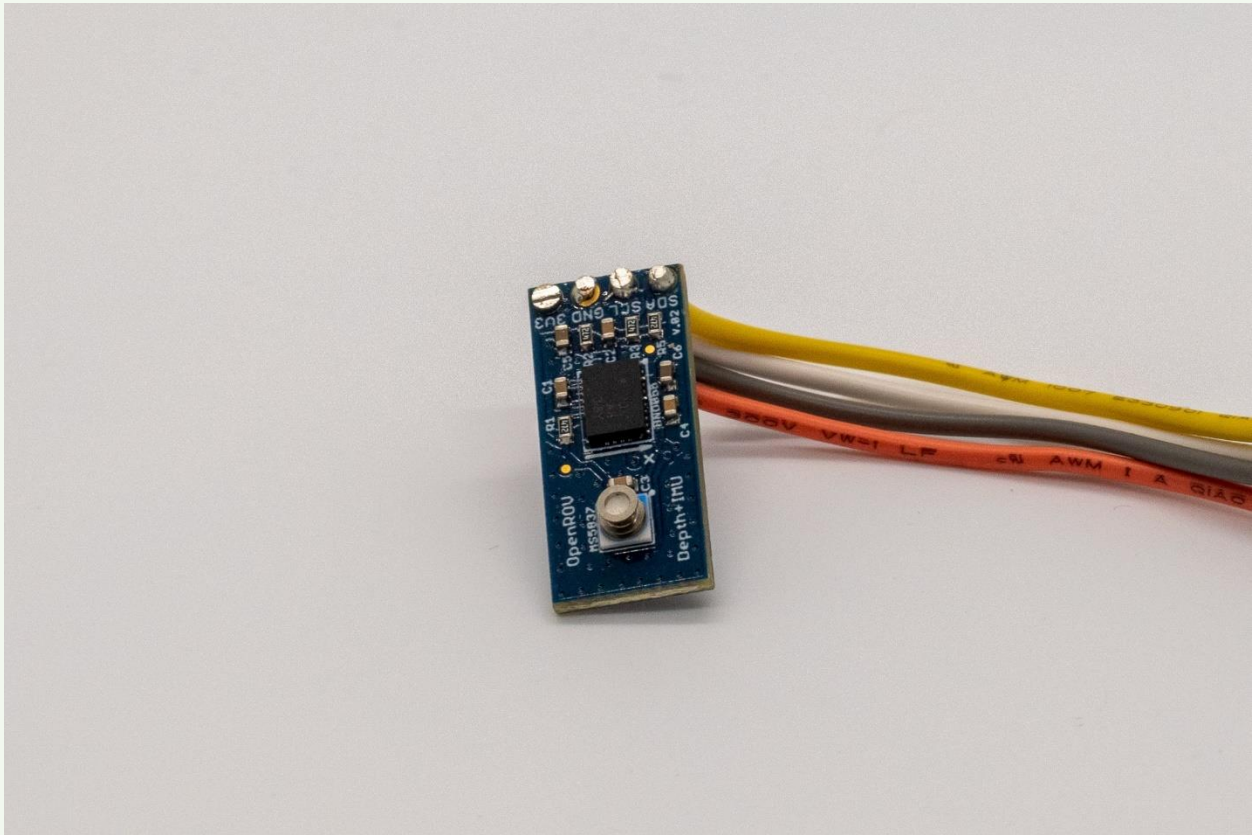You will also need to 3D print the corresponding baseplate in order to pot the sensor.

Figure 43. OpenROV IMU with 30-bar pressure sensor.

Appendix C. External Magnetic Switch

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

1. Assemble the inner switch assembly by running two 5 cm lengths of stripped solid core wire into the associated holes in the 3D-printed assembly.

2. Pinch the wires into place and superglue the outer, insulated wires to the edge of the assembly.

3. Clip off the battery leads about 2 cm from the base. Be careful not to let the leads contact each other.

4. Solder and seal enough wire to the battery leads to allow the battery to sit comfortably in the bottom of the CTD housing (about 10 cm). To reduce strain on the soldered battery leads, tape the wires to the body of the battery.

5. Now solder the wires with the JST connector to the switch assembly above the switch so that they emerge from the top of the CTD. Leave enough wire that you can easily attach and detach the JST connector from the Adalogger.
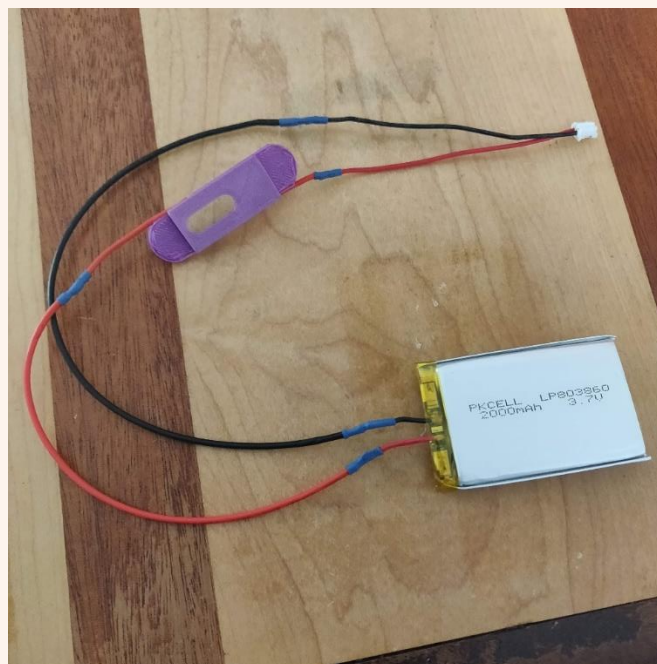

Figure 44. Inner battery switch assembly before battery is taped.

6. Place a 10 mm round magnet inside the housing (you can hold it in place by putting a magnet on the outside, too).

7. Using 10mm adhesive pads, adhere the internal switch component to the wall of the housing. Ensure that the internal magnet moves freely and contacts the two expose wires and that the switch assembly is seated low enough that it does not interfere with the test cap.

8. Stick the switch pin to another magnet using the 10mm double-sided pads.

9. Superglue the external switch assembly to the outside of the PVC pipe such that it traps the pin, is even and parallel with the internal assembly, and the pin can move freely. Line the switch up carefully and tape in place with electrical tape. Take extra caution not to allow the glue to leak into the inner walls of the external switch assembly or it could interfere with movement of the magnet.

10. Connect the control unit to the battery and ensure that the switch works consistently.
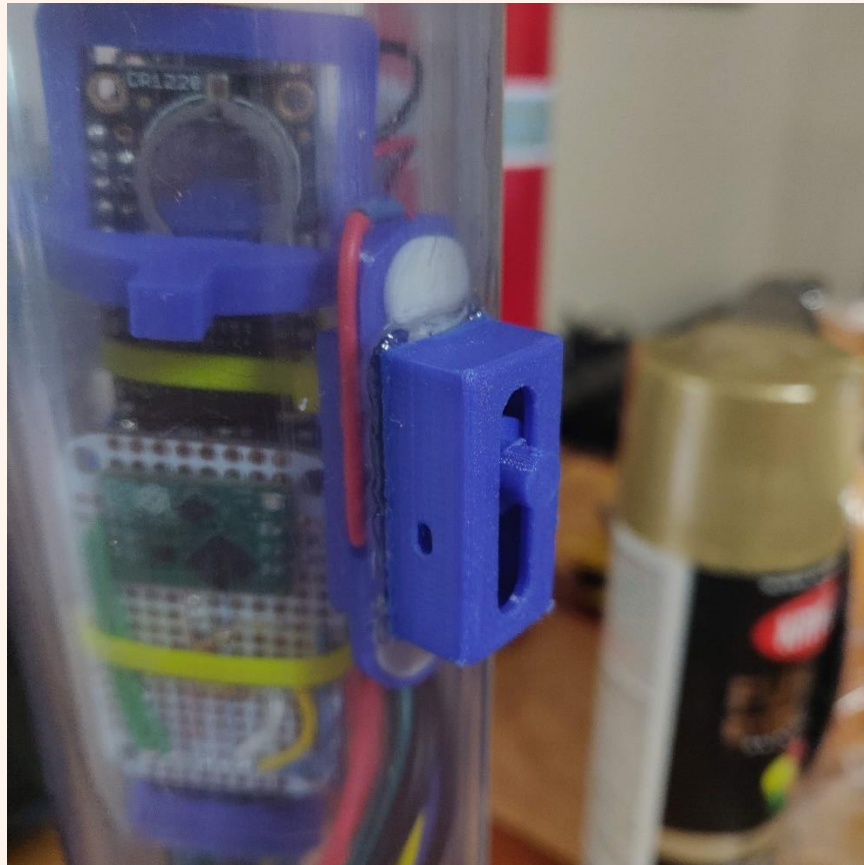
Figure 45. The complete switch assembly in a clear OpenCTD.