Pressure Logger Instructions

# Overview

This pressure logger is intended for high rate, short duration pressure logging in water (ocean and fresh). Examples of typical use include wave height monitoring, coordinated tide and temperature observations, and storm surge observation.

# Specifications

|  |  |  |
| --- | --- | --- |
| Value | Range | Resolution |
| Depth Rating | 300m (limited by sensor) |  |
| Data Parameters |  |  |
| Temperature | 2-40 degrees C | 0.01 Deg C (accuracy +/- 1.5C) |
| Pressure | 0-30 Bar absolute | 0.2 mBar (accuracy +/- 200 mBar) |
| Battery Voltage | 0-5 Volts | 0.01V |
| Sampling Frequency | 0.5 to 20 Hz | 1 ms timing interval |
| Battery Capacity | 4500 mAh @ 3.7V (~14 days run time) |  |

# 

*Pressure Logger fully assembled*

# Electronics

The logger is made using an adafruit 32u4 data logger board (<https://learn.adafruit.com/adafruit-feather-32u4-adalogger>) and an adafruit RTC clock unit (<https://learn.adafruit.com/ds3231-precision-rtc-featherwing>)

The pressure sensor is from Blue Robotics (<https://bluerobotics.com/store/sensors-sonars-cameras/sensors/bar30-sensor-r1> ). It includes an MS5837 absolute pressure sensor housed in a pass-through that fits into the pressure housing endcap. This sensor is rated to 300m and has a pressure resolution of 0.2 millibar. (equivalent to 2mm water depth).

Tests have shown the system maintains a noise level below 0.5 mBar in a static environment. Drift over long periods (hours) has not been well characterized yet as it needs to be compared against well-calibrated barometric pressure sensors.

The pressure sensor is not rated for continuous contact with water (readings drift), so it is protected from water by operating in a mineral-oil filled chamber. The oil is kept in the chamber by a membrane (made of a nitrile glove finger) which transmits pressure from the external water to the interior oil without influencing the pressure. This nitrile membrane is protected from damage by a rubber cap with a small hole to allow water flow.

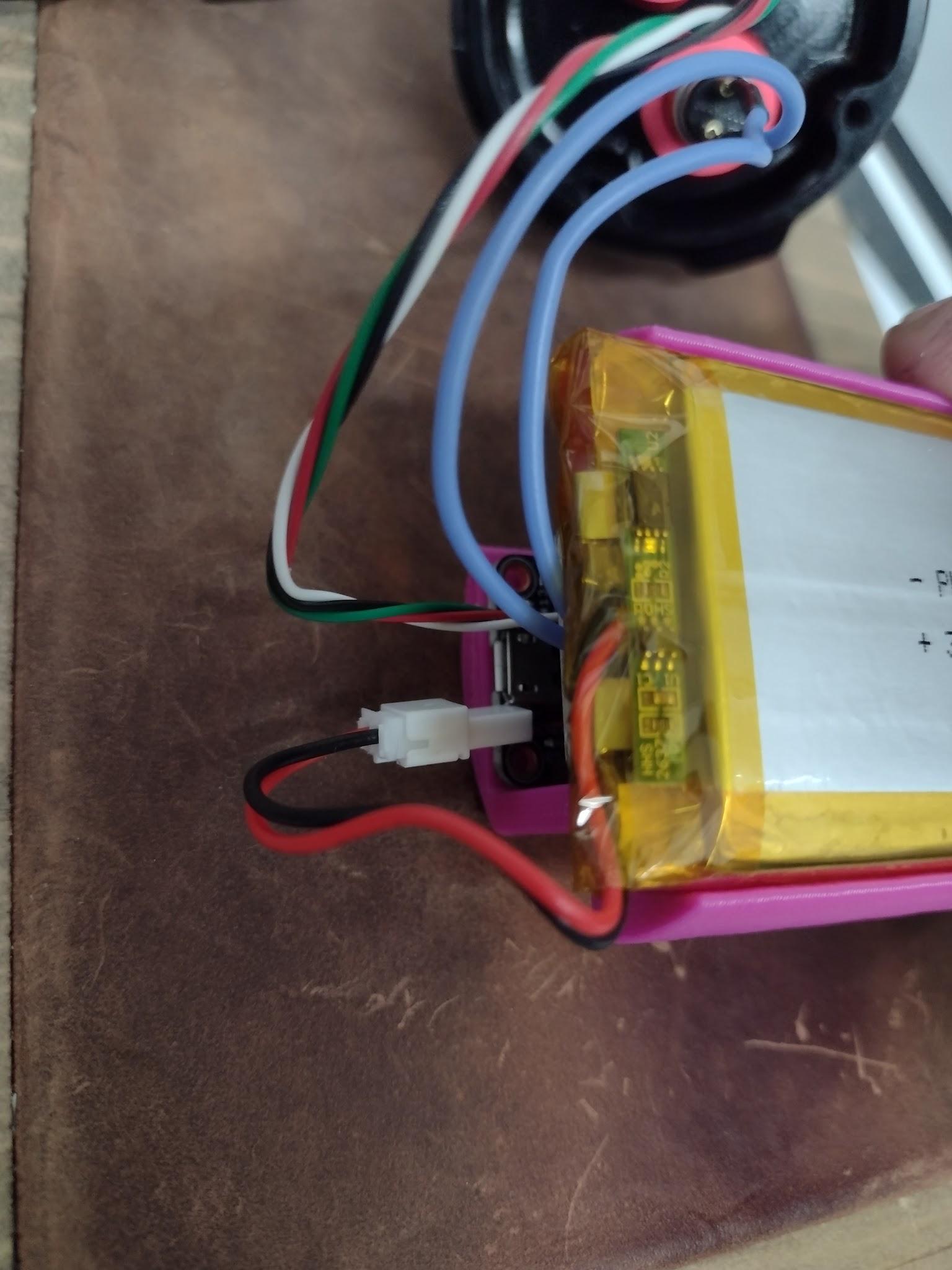
*The pressure sensor is covered in a protective tube filled with mineral oil and covered with a nitrile membrane.*

# Battery

The logger is provided with a 4500mAh battery. It is a single cell battery providing an average 3.7V with a working range from 3.2V to 4.2V. The adafruit logger board automatically turns off when the voltage reaches a minimum to avoid damaging the battery.

\*NOTE ON POLARITY: The batteries provided with the loggers are wired in a “normal” direction with pin 1 of the connector negative and pin 2 positive. The Adafruit logger has a connector polarity reversed from almost all other devices that use the same connectors and battery types so it is provided with a short extension cable to reverse the polarity. When plugging in a battery, always make sure this cable is in place in the logger. When charging, always plug the battery directly into a charger without the extension cable.

There is a lithium button cell battery in the Clock board which provides constant backup power to retain the clock’s date and time reference when the main battery is removed. This is a lithium CR2012 size button cell.



*Battery connector: Always disconnect the battery here, not at the side of the board. The battery connector polarity is correct for the charger directly from the battery. The extension cable hidden between the circuit boards reverses the polarity for the adafruit board and should never be plugged into a charger.*

# Charging

When the logger is plugged into a USB port, an internal battery charging circuit will charge the LiPo battery at a rate of 0.1A. A yellow light next to the USB port on the logger board will indicate charging is active. The light will turn off when the battery is fully charged at 4.2V. This charge rate is very low for the capacity of battery provided (4500mAh) and charging this way will take approximately 45 hours.

Using an external charger is recommended. A charger should be provided with the loggers which is capable of charging 6 batteries simultaneously at 0.6A rate.

# 

*6 batteries attached to the external charger*

# Software

Assembled loggers are pre-programmed when shipped and do not need to be programmed to record data. As of this writing, they are shipped with 4Hz sampling rate. If you prefer to change the logging parameters (such as sampling interval) you should follow the instructions below to reprogram them.

It may be helpful to reprogram the loggers occasionally to update the clock time if it has drifted or if the clock battery needs to be replaced.

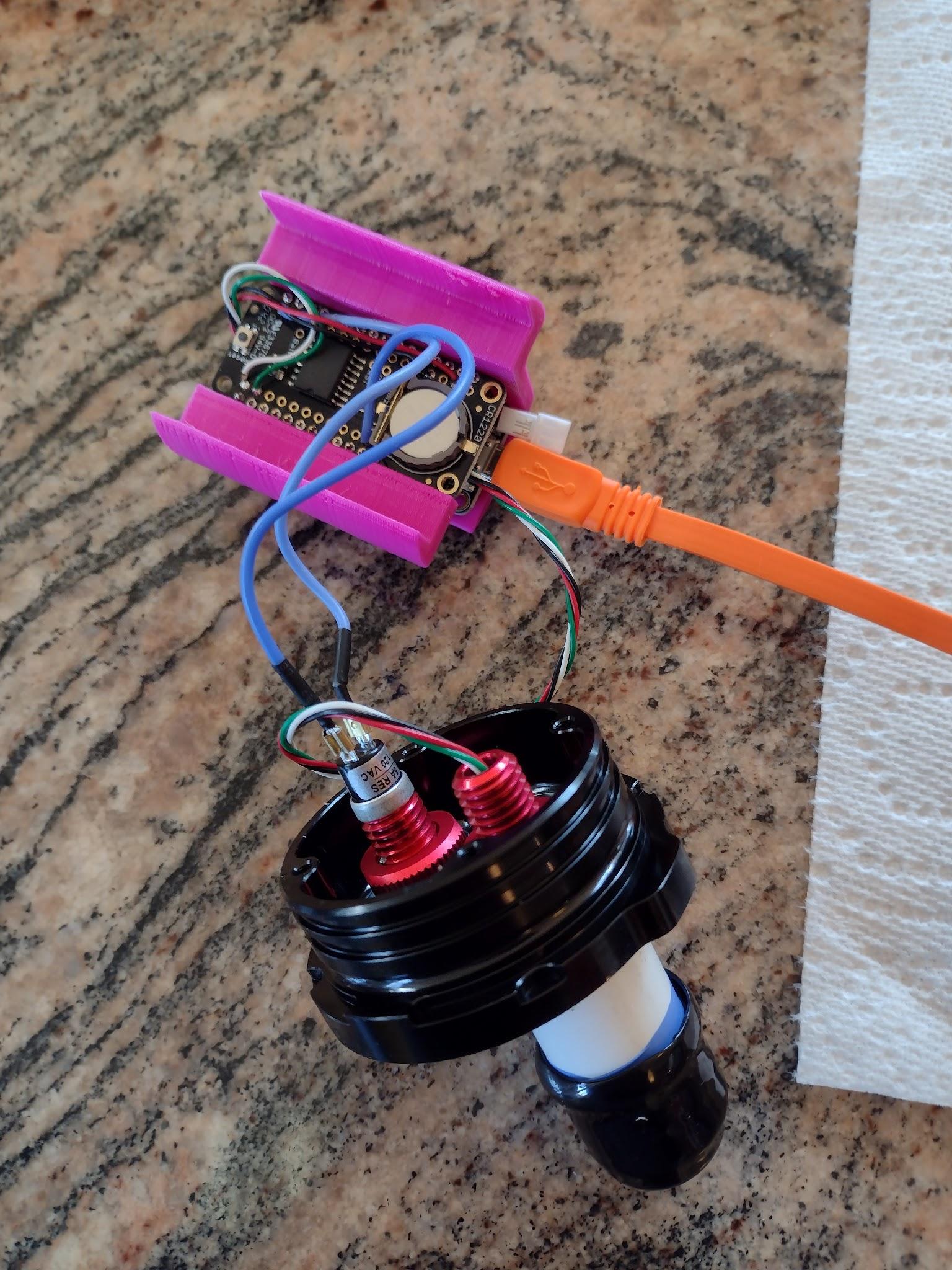
The logger is programmed in the Arduino environment: Arduino IDE (download at <https://www.arduino.cc/en/software>)

Instructions for connecting a logger and installing are here (<https://learn.adafruit.com/adafruit-feather-32u4-adalogger/setup>). This gives instructions for adding the required libraries to the Arduino environment to compile this code.

The loggers should be distributed with the Arduino code sketches needed to run them,

Wavelogger2.ino and ds3231.ino, both available at <https://github.com/JonPompa/Wavelogger2>

The logger board has a USB port on it and can be connected with a USB cable to a computer running the Arduino IDE for programming.



*The logger can be connected to a computer to program it using a USB cable attached to the USB port on the logger board. This port also charges the battery (slowly) from the USB port.*

Arduino sketch Ds3231.ino is used to set the RTC clock on the logger and should only need to be used once when the logger is first powered or when the button cell backup battery is replaced. DS3231.ino should be uploaded to the logger through the Arduino IDE. You can verify success by opening a serial terminal and seeing confirmation of the current time and date. (ctrl+shift+m opens a terminal window in a windows environment) The baud rate of the terminal window should be set to 56700 or whatever rate is specified in the version of ds3231.ino that you are using)

Wavelogger2 is the main program running the data collection. It should be compiled and uploaded to the logger after the ds3231.ino clock setting sketch is run. In this program, there is a set of user parameters to set for the interval between data pushes to the log file and also for frequency of sampling. The sampling interval is given in milliseconds. Know that there is also a 50ms sampling duration added to the period you specify, for example, an interval of 200ms will mean a total time between samples of 250ms, leading to a 4Hz actual sample rate.

Code example:

//User settings//

int logInterval = 4000; //milliseconds between data saves to the SD card \*note max of 32000 allowed with an int data type

int senseInterval = 200; //milliseconds between data read attempts for pressure and temp sensor.

//Data reading +overhead takes about 50ms so a value of 200 here gives ~4Hz,

//Fastest achievable is about 18Hz by setting senseInterval low (5ms)

bool recordBatVolts = true; //record or don't record battery voltage

Log files are started when power is turned on. The log files are named for the date on which they start, using year, month, day format. i.e. 20210131.txt

# Assembly

The logger housing must be opened to access the data port and card reader. Once the logger has been appropriately programmed, it is not necessary to plug it into a computer to restart logging. Inserting a data card and turning on power will always initiate logging.

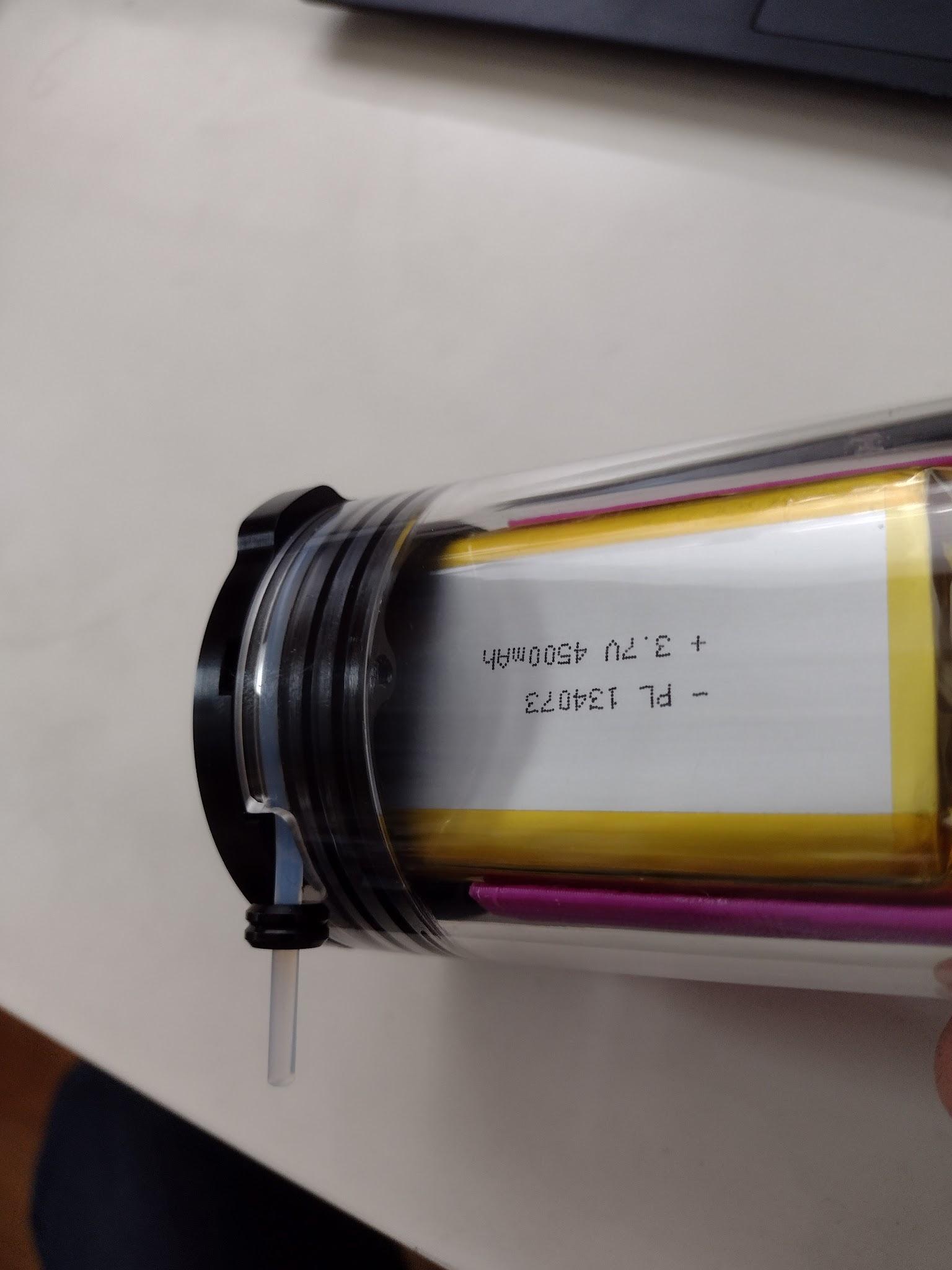
It is recommended to open the logger from the sensor endcap end.



*Opening an endcap is achieved by pulling the white plastic cord then gently sliding the endcap out by hand. Before closing an endcap, always clean and inspect orings. The orings should be lubricated with silicone grease.*

To assemble the logger, slide the electronics tray (with battery) into the housing, being careful to seat the battery into the blank endcap and to align the switch and battery so that they don’t interfere. The end cap with the sensors must be rotated so that its tab aligns with the notch in the housing body for it to seat fully. See images below for important alignment notes.

The following images show some important details of how the parts fit together.



*The battery fits down into the blank endcap when the battery and electronics tray are inserted fully into the pressure housing as shown here. This is only possible at some angles because of the shape of the endcap interior.*



*This image shows mirrored versions of the logger endcap. Unfortunately, some loggers were built in each configuration and the configurations require attention to assemble correctly. The notch in the housing that allows insertion of the locking cord must align with a small tab on the endcap. That alignment sets the position of the switch in the endcap. Because the switch terminals extend downward far enough to interfere with the battery, the electronics tray must be rotated to place the battery under the sensor side of the endcap, not the switch. Please pay attention to this when assembling the loggers!*

The housing is sealed with o-rings on the circumference of the end caps. When the housing is assembled, the rubber o-rings must be inspected, cleaned (very carefully look for threads, fibers, and hair on the rings), and lightly greased with silicone grease.

When the end caps are pushed in, it may be helpful to remove the top of the power switch completely. This allows air to vent from the housing through a small hole (this also turns the logger on). Once the end caps are seated, they should be fastened by sliding the plastic lock cords around the circumference of the housing in the locking groove. The power switch top may then be re-inserted, being careful to note whether it is set in the on or off state. It is also helpful to inspect and grease the o-rings of the switch at this time.

# Power Switch

The logger housing can be assembled with the switch in the off position. This is when the switch top is fully screwed into the base.

**At deployment time, the logger can be turned on by twisting the switch approximately one half turn in the unscrewing direction (counterclockwise).**

**Verify that the logger is recording by watching for an intermittent flashing of a red LED on the processor board next to the USB port.** The red light flashes when data is written to the SD card, so if you have a short data writing interval, it flashes often and if you have a long data writing interval, it flashes seldom.



*Here the power switch is shown on the end cap next to the bare pressure sensor. On finished loggers, the pressure sensor is protected by a cover.*

# Data

Whenever the logger is turned on, it immediately starts recording data. Data is appended to an existing file if the logger has been started more than once in a day. It will start a new file if it is turned on during a new day.

Data is stored on a micro SD card which is inserted into the logger board (at the end opposite from the sensors and USB port). This card should be formatted in FAT32 format to work with the software.



*The SD card can be seen in its slot under the battery in this image of the disassembled logger.*

Data is recorded in ASCII text format. The data fields include a header line explaining the columns. The data includes time in year, month, day, hour, min, sec, and milliseconds. The millisecond value is not referenced to the seconds value, and so may roll over independently from the count of seconds.

The data columns are:

Year Month Day Hour Minute Second Milliseconds Pressure Temperature BatteryVolts

Data for temperature is recorded in degrees C.

Data for pressure is provided in absolute millibars.

Battery voltage is recorded in volts.

\*Note that as battery voltage drops for the last few hours of recording, before the low voltage cutoff shuts off the logger, the pressure reading will drift upward by several millibar.

At the time of this writing, there is no provision for retrieving data from the logger other than manually extracting the microSD card and reading it in another device. Future improvements include implementing a set of commands to retrieve the data via the USB connection and to add a USB connection through a connector on the pressure housing to prevent the need to open the loggers after each deployment.

# Data Processing

This logger records data files that become large quickly – i.e. hundreds of megabytes over a few days. Typically processing the data is best done in a scientific programming language set up for such data. Examples would be the R language, Matlab, or Python with NumPy.

A few notes on importing data follow

1. The files have a header row. This header row gives column names. If the logging was restarted within a 24 our period, there will be an extra copy of this header line in the data for each restart instance. This may challenge your data importing script!
2. The pressure data is in absolute pressure. The sensor is sensitive enough to track barometric pressure very well, so your pressure data processing may need to include an effort to obtain a local barometric pressure record and subtract that pressure from the recorded data. You may be able to use an on-shore logger to serve as a reference pressure recorder if you have a spare logger available.
3. Battery voltage is included as a debugging and convenience feature. You can reduce the data file size by turning off battery voltage logging (set the battery logging parameter value to “false” in the wavelogger2.ino file).
4. Battery voltage will influence the pressure reading when the battery is very low. Do not use the last few hours of the pressure record before the logger shuts down due to low battery state. The pressure reading will drift upward during that time.

# Mounting

The logger can be attached to a mounting location with cable ties, hose clamps, or other purpose-made clamps around the housing body. If using hose clamps, you may want to cover the clamp in a rubber tube such as a bicycle tire inner tube to prevent scratching the housing. If using cable-ties, it is suggested to avoid nylon cable ties which lose a significant amount of their strength when submerged in water over time.

# Credits

This manual is a living document and is subject to change.

This logger has been designed by Jon Pompa using readily available parts and code as referenced above.

Major inspiration was taken from Luke Miller’s Open Wave Height Logger project (which provides a much higher level of documentation, testing, and a longer-duration logger design)

Much of this project is also built on the Adalogger family of devices from Adafruit which provide nearly complete hardware and software out of the box.

The underwater housings and sensors are stock items from Blue Robotics.

# Contact

Please direct questions or ideas to Jon Pompa - [jon@reefgen.io](mailto:jon@reefgen.io)

Plans for incremental upgrades at the moment include data access and battery charging by connector, better wire management, and mounting brackets.

If anyone wants a construction manual for building these loggers, please make a request.

# Bill of Materials

|  |  |  |
| --- | --- | --- |
| Part |  | Qty |
|  |  |  |
| Pressure housing |  |  |
|  | Blue robotics 2" x 100mm housing | 1 |
|  | Blue robotics 2" blank endcap with flange | 1 |
|  | Blue robotics 2" endcap 2 hole with flange | 1 |
|  | BR switch / vent penetrator (modified to vent) | 1 |
|  | BR Bar30 pressure sensor | 1 |
|  | BR sealing cord (included with endcaps) | 2 |
|  | BR sealing orings (included with endcaps) | 4 |
|  |  |  |
| Electronics |  |  |
|  | Adafruit Adalogger 32u4 | 1 |
|  | Adafruit precision RTC featherwing | 1 |
|  | SD card | 1 |
|  | stacking header pins | 2 |
|  | header pins (included with adalogger) | 1 |
|  | Lipo single cell 4500mAh 13x40x72mm | 1 |
|  | CR1220 button cell (3V lithium) | 1 |
|  | Mounting bracket (parts holder\_R2.sldprt) | 1 |
|  |  |  |
| Sensor protection |  |  |
|  | PVC 1/2 pipe bored out for sensor plus oring groove | 1 |
|  | 5min epoxy (gorilla) to seal pvc to sensor | 1 |
|  | Nitrile glove finger | 1 |
|  | mineral oil | 1 |
|  | Rubber cap (with hole) to go over pvc pipe | 1 |
|  | oring m2x17.5 | 1 |