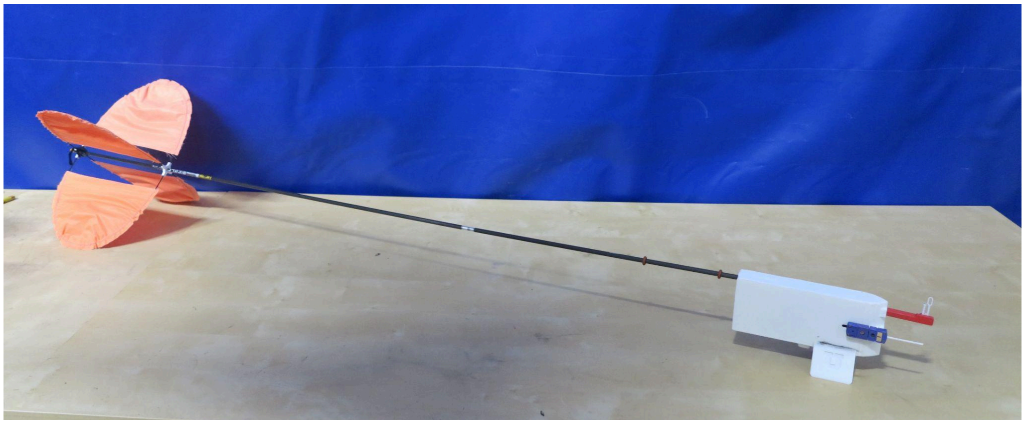

University of Utah Tethersonde User Manual

May 2024

Developed by Undergraduate Mechanical Engineers Carson Beagles, Quinn Brush, JonJon Drain, Tristan Archuleta, Matthew Garceau

Table of Contents

Introduction

1.1. Purpose

This document is a user manual for the tethersonde, developed by undergraduate University of Utah mechanical engineering students for the Environmental Fluid Dynamics Department (EFD).

1.2. System Overview

The tethersonde system is developed for boundary layer and meteorological measurements in conjunction with a moored weather balloon. The supported measurements include wind speed, direction, pressure, humidity, and temperature up to a 10 Hz sample rate. Data is written to an onboard microSD card and transmitted to a ground station via LoRa communication. An external LED reports calibration and battery life statuses to the ground during deployment. The device program code is written in Arduino (C++) and is controlled by an Arduino MKR WAN 1310.

Cost	$230
Weight	272 g
Sampling Rate	10 Hz
Run time	+6hr
Range	+1km

Sensor	Avg Err	Description
BN0085 IMU	0.433°	Inertial Measurement Unit
RevP.	0.2 m/s	Thermal Anemometer
SHT45	0.1°C	Ambient Temperature & Hydrometer
BMP390	3 Pa	Barometric Pressure
Thermocouple Type E	1.0°C	Fast Response Ambient Temperature

1.3. Part List

The parts list includes the cost for the creation of one sonde.

Part/ Part #	Price	Brand/Supplier
Arduino MKR WAN 1310	$45.60	Arduino
BMP390	$10.95	Adafruit
MAX31856	$17.50	Adafruit
SHT45	$12.50	Adafruit
MicroSD	$7.50	Adafruit
BNO085	$24.96	Adafruit
Wind Sensor Rev.P	$39.95	Modern Device
Li-Po-3.7v 2500mAh	$12.50	Adafruit
UCF Anetenna	$5.60	Arduino
TPS61023 (5v Booster)	$3.95	Adafruit
TPS61040 (12v Booster)	$2.95	Adafruit
JST-PH 2-Pin Connector	$0.75	Adafruit
RTC	$13.50	Adafruit
RGB LED	$2.00	Adafruit
PCB	$17.00	OSH Park
16GB MicroSD Card	$6.20	Amazon
JST 1.25mm Pitch 5 Pin Connector	$0.72	Digikey
JST JUMPER 05SR-3S	$1.73	Digikey
Male Thermocouple Type E Flat Pin Connector	$3.93	McMaster Carr
Female Thermocouple Type E Flat Pin Connector	$4.95	McMaster Carr
Thermocouple Type E Wire (approx 150mm)	$2.22 Per Ft	McMaster Carr
Thermocouple Type E Bare Wire (24 Ga)	$83.31 (5 Pack)	Omega

Tail Vane Shaft- Kinetic Pierce 300 Platinum (6mm)	$140 (12 pack)	Kinetic Pierce 300 Platinum (6mm)
(4x) 220 Ohm Resistors	$0.1(single resistor)	Digikey
(2x) 3mm Screws	$ 0.16 (Single Screw)	Digikey
(2x) 2mm Screws	$0.12 (Single screw)	Digikey

Implementation and Maintenance

2.1 System Requirements

2.1.1 Software Requirements

Arduino IDE: Ensure version 2.3.2 or later is installed

MATLAB: Ensure that version R2022b or later is installed and the Matlab Arduino hardware package is installed.

2.1.2 Hardware Requirements

The PCB manufactured for each tethersonde requires each sensor to be soldered onto the board except for the microcontroller. The microcontroller is connected through header pins, which may be removed to update its firmware.

Before deploying a sonde, ensure the battery on board is fully charged.

2.2 System Setup and Configuration

Install the required Arduino and Matlab libraries. Upon downloading and running the scripts in section 3, you will be prompted to install the missing libraries if they are not already present in the integrated development environment.

Pin List

Part	Pins (Arduino MKR WAN 1310)
MKR WAN 1310	Vin(5V Rail), GND(GND Rail)

RevP.	OUT(A0), 12V(12V Rail), GND(GND), SHDN(GND)
BMP390	Vin(5V Rail), GND(GND),SCL(SCL), SDL(SDL)
BN0085	Vin(5V Rail), GND(GND),SCL(SCL), SDL(SDL)
MicroSD	5V(5V Rail), GND(GND), CLK(D13), DO(D12), DI(D11), CS(D7)
SHT45	Vin(5V Rail), GND(GND),SCL(SCL), SDL(SDL)
RGB LED	GND(GND), Red(D3), Green (D4), Blue (D5)
MAX31856	5V(5V Rail), GND(GND), CLK(D13), DO(D12), DI(D11), CS(D6)
5V Booster	Vin(3.7V LiPo), GND(GND), 5V(5V Rail)
12V Booster	Vin(3.7V LiPo), GND(GND), 12V(12V Rail)

2.3 Files

Refer to the Github for all necessary code files.

https://github.com/UofU-EFD/TetherSonde2024

Master Arduino code

Matlab export code

3. Deployment Routines

Step 1: After PCB is complete, upload the tethersondeScriptLoraControl.ino to each MKRWAN1310 microcontroller. Make sure to remove the microcontroller from the PCB while uploading the code. Uncomment the desired identity value and frequency one wishes for a given tethersonde.

Step 2: Before powering the sonde on, place the orange protective sheath over the RevP node in front of it for self-calibration. Power the sonde up by plugging it into the battery. Upon startup, the sonde will flash green multiple times. If the setup is successful, the sonde will blink green and move into self-calibration. The sonde will then blink blue once it calibrates, and then it's ready to receive user input from the base station. Troubleshooting tips below.

- If the sonde blinks red only after a green flash, check the SD card first and ensure there is one present, then press the reset button.

- If the sonde blinks red, followed by cyan, the IMU failed to be read. This error is solved by pressing the reset button until the microcontroller can find the IMU.
- If the sonde flashes blink twice, then the LoRa module onboard the MKRWAN1310 has failed to be read, which signifies a deeper problem with the microcontroller. We recommend replacing the microcontroller if this occurs.

Step 3: Open…

- Arduino Script ‘timeStampTransmitter.ino’
- Matlab Script ‘baseStationController.m’
- Matlab Script ‘MessageTransmitter.m’

Step 4: Upload ‘timeStampTransmitter.ino’ to the base station Arduino. Ensure the correct baud rate (115200) in the Serial Monitor. **Serial monitor must be closed in order for Matlab to communicate with Arduino.**

Step 5: In the Matlab ‘MessageTransmitter.m’, update the code to the correct port of the base station Arduino, as seen in Figure 1. Mac OS format will look analogous to ‘/dev/cu.usbmodem1301’, and Windows OS format will look analogous to ‘Com1’.

```
%% Message trasnmittier script
% The purpose of this script is to transmit the user inputs to each
% tethersonde
clear
clc

% Connect to the arduino 'Base Station'
reciever = serialport('/dev/cu.usbmodem1301',115200);
% Each tethersonde after starting is expecting an array of inputs
% These inputs are declination and the Unix timestamp
% [Float, Long]
% These are parsed by the arduino but sent as 1 message
% A successful message is achieed when the tethersonde blinks purple
% After blining purple the tethersonde is recording data locally at 10 Hz
% and trasnmitting on its frequency the same data at 5 Hz, the limit of the
% arduino

% Send the declination then the time stamp last
declination = -10.85; % This is 10.85E

% a signal to start transmitting data
initialTimeStamp = datetime('now', 'TimeZone','UTC','Format', 'yyyy-MM-dd HH:mm:ss.SSSSSSS');

% Convert to a unix time stamp
timeAbsolute = posixtime(initialTimeStamp);

% The base station expects to read a string so we must formulate a string
% here
messageValues = [declination,timeAbsolute];
messageString = string(messageValues);
fullMessage = join(messageString, ','); % Comma delimiter

% send datums to the arduino to sendo over the LoRa
write(reciever,fullMessage,'uint8');

```

Figure 1

Step 6: The ‘MessageTransmitter.m’, declination, and time stamps will be sent to all tethersondes. Refer to Figure 1. Run the script and ensure that all to-be-deployed sondes flash purple. If the sonde does not flash purple, run the script until the particular sonde flashes purple. Each s9onde is built to only listen for one message, so after a message has been received at its frequency, the sonde will begin to record and transmit data until its battery either dies or is removed.

Step 7: In the Matlab 'baseStationController.m' update the code to the correct port of the base station Arduino, as seen in Figure 2. Mac OS format will look analogous to '/dev/cu.usbmodem1101', and Windows OS format will look analogous to 'Com1'.

```
%% Tethersonde Reciever controller
% This script access the arduino base station through the serial port
% here we can collect and plot any data the is recieved by the arduino

clear
clc
%% connect to the arduino serial port
% Verify the port connection is correct by checking yourv arduino IDE first
% also check to see if any other serial monitors are open, they need to be
% closed.

reciever = serialport('/dev/cu.usbmodem1301',115200);
timestamp = 1; % time in milliseconds

% Logging commands
duration = 0.01 * 3600; % 3 hour duration in seconds (1 h = 3600 s)
elapsedTime = 0; % initialize 1 seond, this is for matlab's timer
stream = []; % stream data array initilization
index = 1; % this is for indexing the proper rows of the array
% these index values are forvthe 6 potential sondes that could be recorded
% in this script
index1 = 1;
index2 = 1;
index3 = 1;
index4 = 1;
index5 = 1;
index6 = 1;

% Declare timer
startTime = tic;

% Declare the initial arrays and time stamps
% This is pre allocating for size and type

```

Figure 2

```
    % Plotting code for real time profiles
    % There are only 3 plots currently but more can be included
    % Azimuth, temperature, and windspeed
    subplot(2, 2, 1);
    plot(timeStampData1(:,1), dataSet1(:,3), 'b');
    hold on
    plot(timeStampData2(:,1), dataSet2(:,3), 'r');
    hold on
    plot(timeStampData3(:,1), dataSet3(:,3), 'g');
    % hold on
    % plot(dataSet4(:,1), dataSet4(:,3), 'k');
    % hold on
    % plot(dataSet5(:,1), dataSet5(:,3), 'c');
    % hold on
    % plot(dataSet6(:,1), dataSet6(:,3), 'm');
    title('Azimuth');

    subplot(2, 2, 2);
    plot(timeStampData1(:,1), dataSet1(:,5), 'b');
    hold on
    plot(timeStampData2(:,1), dataSet2(:,5), 'r');
    hold on
    plot(timeStampData3(:,1), dataSet3(:,5), 'g');
    % hold on
    % plot(dataSet4(:,1), dataSet4(:,5), 'k');
    % hold on
    % plot(dataSet5(:,1), dataSet5(:,5), 'c');
    % hold on
    % plot(dataSet6(:,1), dataSet6(:,5), 'm');
    title('Temperature');
```

Figure 3

Step 8: The script contains the code to record between 1 and 6 tethersondes simultaneously. Ensure that each sonde has uncommented corresponding sections, as seen in Figure 3.

Step 9: After the previous steps have been completed, 'baseStationController.m' can be run to begin a live feed of azimuth, wind velocity, and temperature. More plots can be added to the script following the subplot format present.

4. Exporting Data

All data is stored on the onboard microSD card in .txt format. After each launch, power off the device and transfer the data from the card to a computer. Each sonde will have its own sets of data. If the device is powered on again without removing the data, all data on the card will be erased. Transmitted data can also be stored as a timetable using the baseStationController MATLAB script. This will be saved in the same directory where the baseStationController MATLAB script is stored. To rename the filename of the timetables produced, edit the following string for each deployed sonde in the MATLAB script.

```
% Save as a time table.
filename = 'sonde1.csv';
writetimetable(T1, filename);
```

5. Troubleshooting

5.1 General Troubleshooting Tips

First, ensure the setup and deployment routines outlined in sections 2 and 3 were correctly performed in the instance of device failure. Next, inspect the electrical hardware, looking for solder bridging across pins, corrosion on devices, and the absence of LED lights on the devices that have them (BNO085, BMP390, SHT45, MicroSD breakout, Arduino MKR WAN 1310). Ensure the LiPo battery is fully charged before installment.

If Rev.P does not produce reasonable values for the expected wind speed of the day, the device may be damaged or uncalibrated. Ensure the calibration sequence is performed correctly before deployment with the node cover installed. A new calibration curve may need to be produced if the device still malfunctions after the calibration sequence. This can be achieved by testing the sonde package in a wind tunnel against known speed outputs and altering the log scale regression in the Arduino script. The regression values are in the “tethersondeScriptLoRaControl” Arduino script as follows.

```
// Calibration values
float a = -0.4693;
float b = 0.0147;
float c = 0.5874;
float d = 0.2344;
```

If it is determined that a sub-device is inoperable (either through code failures, boosters unable to power the PCB, visual damage, etc.), remove the soldered component and replace it.

During the startup process, there are signal lights that will flash specific colors depending on the type of error that occurs. If the LED blinks red, either the SD card reader has failed or the LoRa module was not read. To solve this, re-insert the SD card and reset it. If the IMU fails to be read, the LED will blink red for a count of 20 and then blink cyan. This is solved by resetting the Arduino until the IMU can be read; usually, three resets will work.

Appendix

Building Tethersondes

Refer to the parts list for a total bill of materials for sonde production.

Assembling Circuitry

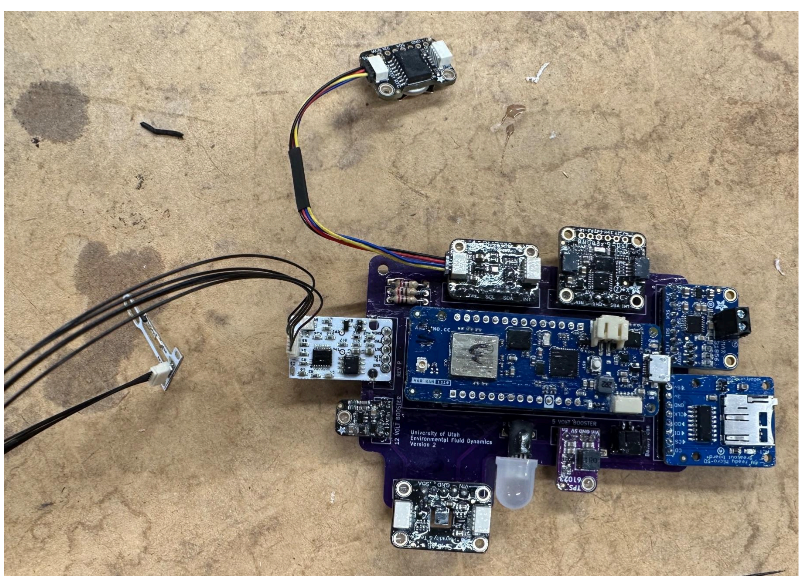

Figure A1: Final printed circuit board assembly

Gerber files for printing the printed circuit board (PCB) are available at the GitHub link below.

https://github.com/UofU-EFD/TetherSonde2024

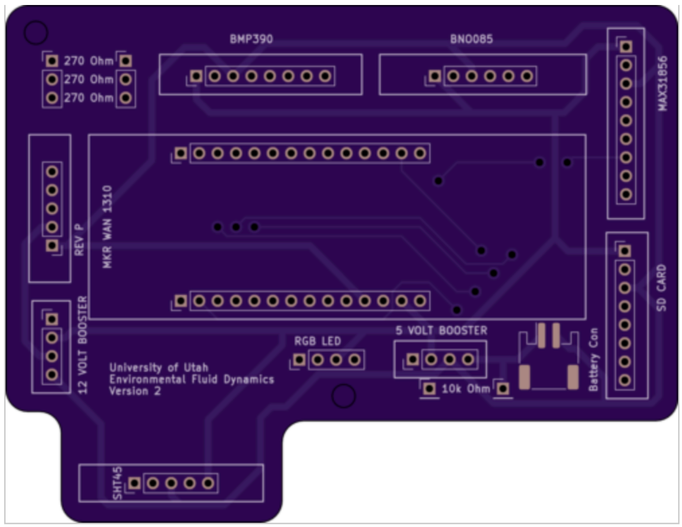

Figure A2: PCB developed in KiCad, available at the GitHub link

1. **Extend nodes from Rev.P Wind Sensor**
    a. The Rev.P from Modern Device features a detachable thermistor and thermometer nodes. A solid black line marks the detachment point. Carefully detach the nodes from the sensor's body using a pair of wire snips.

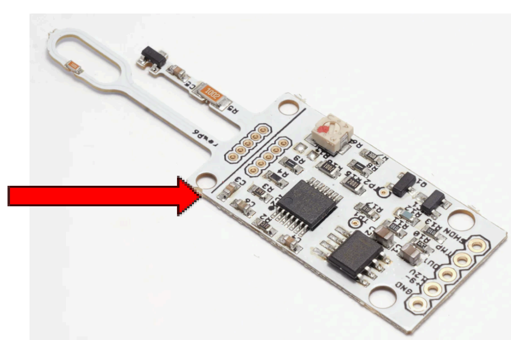

Figure A3: Rev.P Wind Sensor with the arrow showing detachment point

    b. Solder 2 JST 5 Pin female connectors onto the copper holes on both the node and sensor bodies. Thin-diameter rosin core solder wire (0.5-0.8mm) is recommended. Install the JST connectors facing toward each other if the nodes have not been detached yet (front of the connector to the front of the connector or back to back), as shown in the figure below (Figure A4). This ensures the JST connectors are in the correct orientation such that a JST 5-pin cable can connect the sensor body to the nodes in line without crossing the wires.

Figure A4: JST 5 Pin Female Connector (Left). Rev.P Wind Sensor with JST 5 Pin connectors installed (Right)

2. **Solder male 0.1” pin headers onto components**
    a. Using thin-diameter rosin core solder wire (0.5-0.8mm recommended), solder male pins onto corresponding components with the long end pointing down through the back. The following components require installing male headers: Rev.P, BNO085, BMP390, MAX31856, MicroSD Breakout, TPS61040, TPS61023, SHT45.
3. **Solder 2 1x14 0.1” Female headers onto PCB**
    a. Inside the solder mask box labeled “MKR WAN 1310,” solder 2 rows of 1x14 0.1” pitch female headers onto the PCB. This allows the microcontroller to be removed.
4. **Solder 1x5 0.1” Female header onto PCB**
    a. Inside the solder box labeled “SHT45”, solder 1 row of 1x5 0.1” pitch female headers onto the PCB. This allows for the removability of the SHT45 device, which can become waterlogged.
5. **Solder Common Anode LED on PCB**
    a. The chosen LED for the PCB design is a common anode LED. To ensure proper installation, the anode lead of the LED (commonly the longest lead), should be lined up with the 3rd hole from the left in the row of copper holes that correspond to the LED.
    b. Cut a piece of heat shrink to the length of the LED leads before soldering.
    c. Solder the LED leads into the copper holes labeled as “RGB LED”.
    d. Using a heat gun, shrink the wrap around the LED

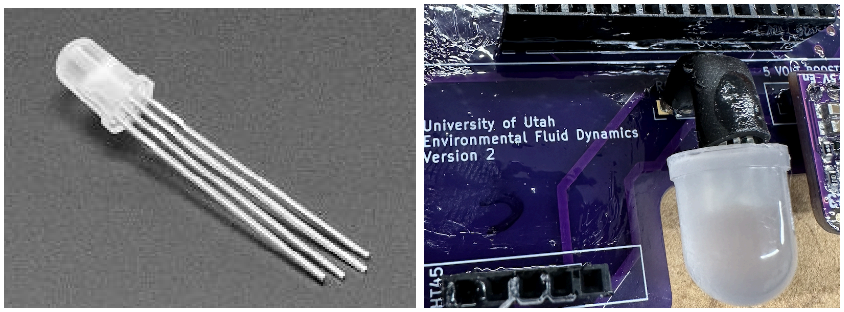

Figure A5: Common Anode RGB LED with the most extended lead being the anode lead (Left). LED installed with shrink wrap on PCB (right)

6. **Solder Devices on PCB**
    a. Solder the devices onto the PCB. This includes the Rev.P sensor body, BMP390, BNO085, MAX31856, MicroSD Breakout, TPS61023, and TPS61040.
    b. All devices have corresponding solder mask labels. The TPS61023 is the 5V booster, and the TPS61040 is the 12V booster.
    c. The PCB is designed to have the devices soldered hanging off of it. To ensure the proper orientation of all devices, refer to Figure A1 for the complete PCB design.The BNO085 must be oriented with the I2C pins soldered to the PCB (Vin,3Vo,GND,SDA,SCL)
7. **Solder Resistors on PCB**
    a. Solder 3x 270*Ω* resistors in the top left corner of the PCB as labeled by the solder mask row of “270 Ohm” text. These correspond to the LED light that is to be installed.
    b. Solder 1x10k*Ω* resistor in the bottom of the LED as labeled by the solder mask text “10k Ohm”.
    c. For battery indication capabilities of the tethersonde, an additional 10k*Ω* resistor is added on the back of the PCB to create a voltage divider. Connect one end of the resistor to the ground pin of the 5 V booster (TPS61023) and the other end to the lead downstream lead of the 10k*Ω* resistor connected on the front of the PCB. Refer to Figure A6 for the correct placement of the additional 10k*Ω* resistor on the back of the PCB.

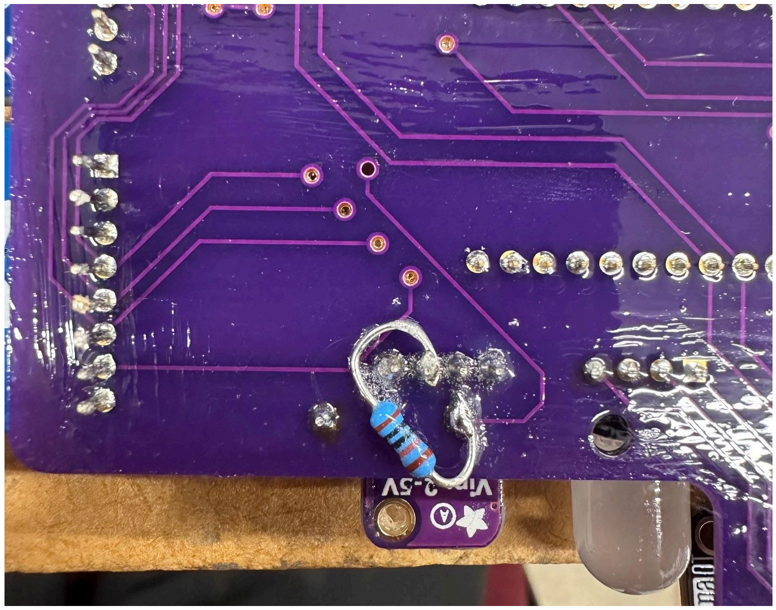

Figure A6: 10kΩ resistor used for battery indication on the back of PCB

8. **Solder JST 2-pin battery connector on PCB**
    a. Solder JST 2-pin battery connector on PCB. Solder mask text “Battery Con” marks the location of the copper pads for soldering.

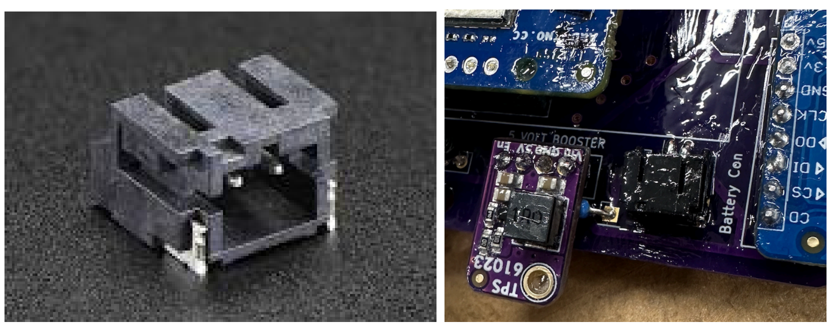

Figure A7: JST 2-pin battery connector (Left). Battery connector installed on PCB (Right)

9. **Connect RTC Module**
    a. Connect the RTC module with a 4-pin QT-QT Stemma extension cable into the I2C port on the module. Connect the other end of the cable to the I2C port on the BMP390.
    b. A CR1220 3V battery externally powers the RTC module. Install this battery before use.

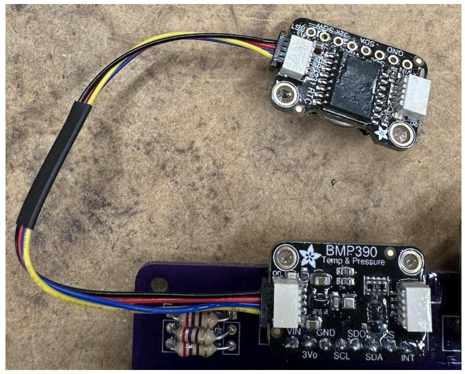

Figure A8: RTC Module connected to BMP390 with QT-QT stemma cable

10. Connect SHT45 to the corresponding female header row on PCB

   a. Refer to Figure A1 for proper placement of the SHT45 device.

11. Remove female headers from Arduino MKR Wan 1310 and Install on PCB

   a. To reduce the device's form factor, the top row of female headers on the Arduino MKR Wan 1310 is removed. To do so, use a pair of pliers to pull the female headers off the protruding pins on the top of the Arduino. Use wire snips to trim the pins on the top of the Arduino, as seen in Figure A1.

12. Connect 2500 mAh LiPo battery

   a. Connect the 2500 mAh LiPo battery to the onboard connector. Ensure that the positive and ground leads on the battery are oriented, as shown in Figure A9, with the red on the left. If battery leads are reversed, remove them from the white male connector and swap them to achieve the proper orientation.

   b. Upon connection, tiny LED lights on the Arduino and I2C devices will power on. The Rev.P Node extension will also begin to heat up. These are telling of an adequately powered device.

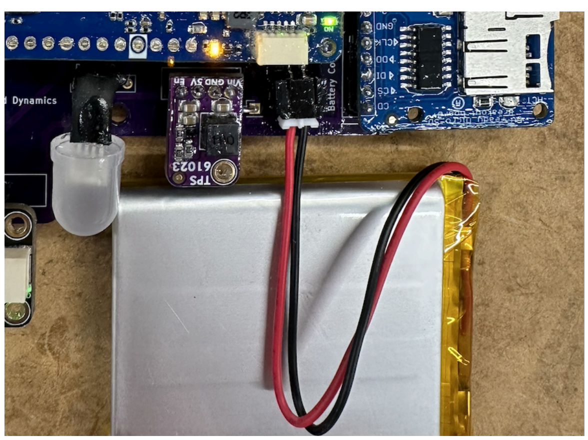

Figure A9: 2500 mAh LiPo battery installed

Assembling the Housing

Refer to the GitHub repository for necessary files needed for 3D printing

Locate:

- Main housing V2.3
- Main housing Lid V2.3
- Radiation Sheild V2.3
- Anemometer Holder
- Anemometer Cover

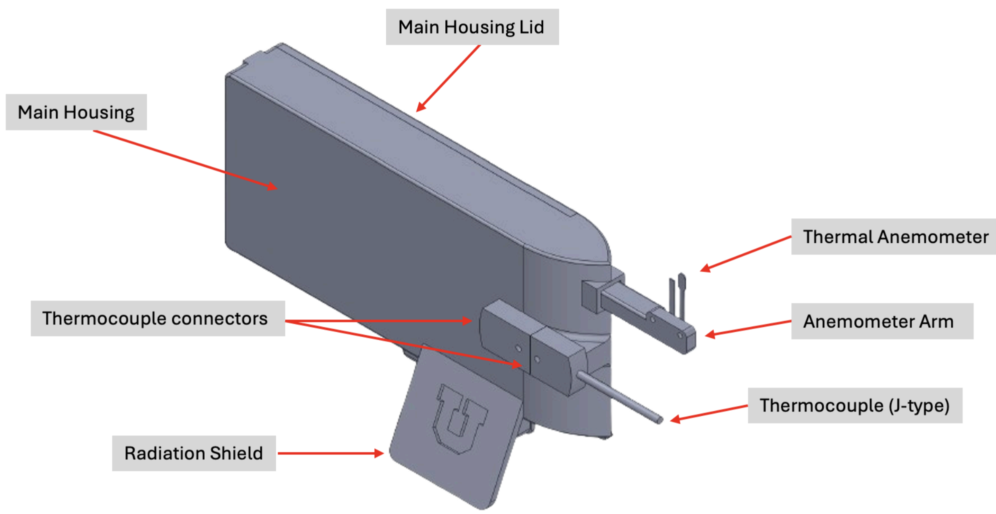

Figure B1: Full Outside Assembly

For the construction of one tethersonde, a single print of each file is needed, except for the radiation shield, which will require two prints of the same file. ASA filament is recommended as its properties are more UV resistant.

Bambu Labs' P1S and X1 Carbon were used with ASA during the design process and provided quality parts. The following settings are recommended if the Bambu Labs P1S or X1 Carbon are being used.

Filament Settings

*Factory Polylite ASA settings were modified to the following. Parameters not stated remained the same.

Recommended nozzle temperature: Min: 240 deg C Max: 260

Nozzle print temperature: initial layer: 250 other layer: 250

Texture PEI Plate: initial layer: 100 other layer: 100

Print Settings

Sparse Infill Density: 100%

Bed type: PEI textured plate

Supports: Enabled

Brim: ON w/ outer brim only

*Unstated parameters remained at the recommended settings.

Once prints are complete and supports have been removed, the assembly process will go as follows.

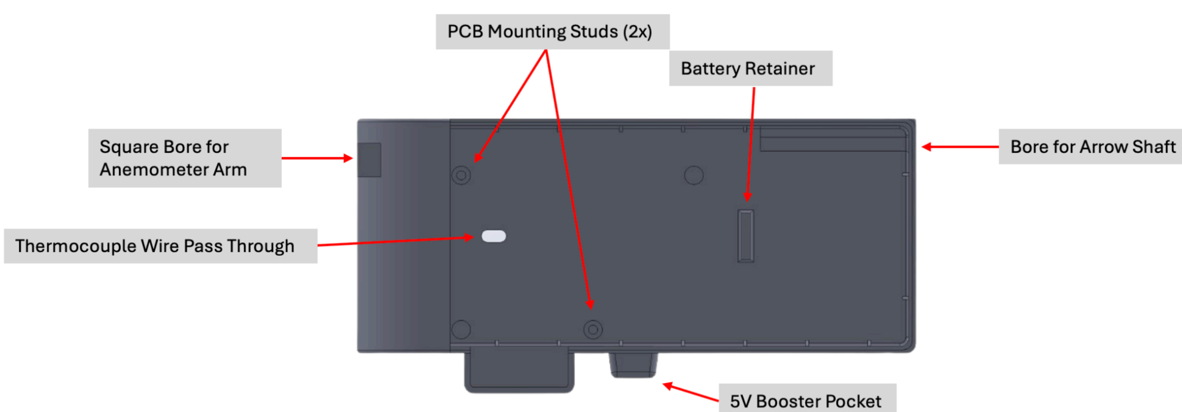

Figure B2: Main Housing

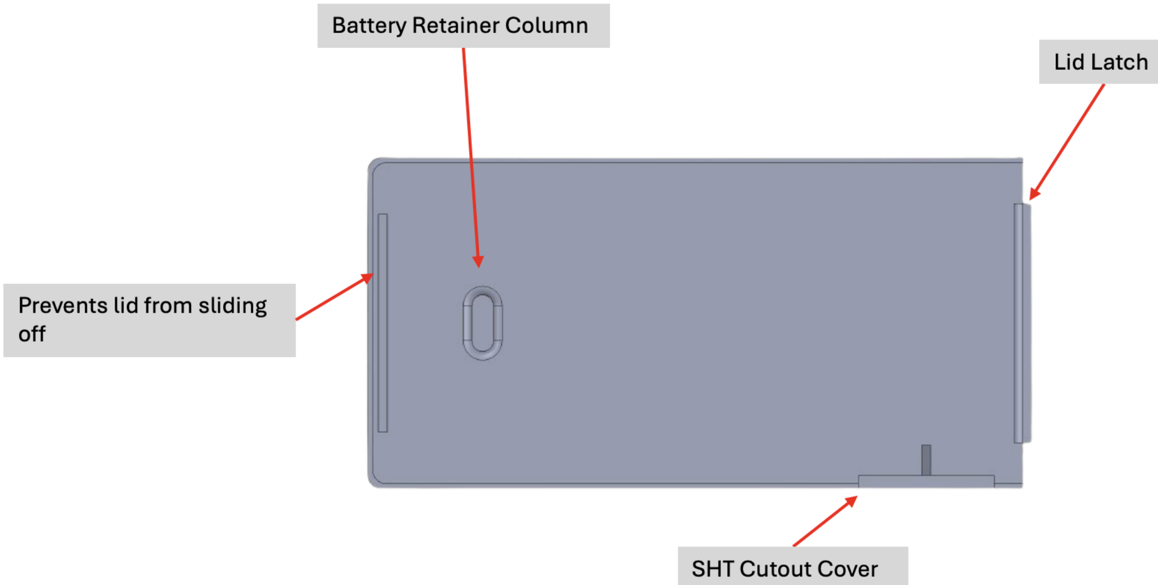

Figure B3: Main Housing Lid

1. **Attaching Radiation Shields**
   a. **Main House**

Painter's tape and super glue are recommended. The radiation shield's bottom edge of the interface will be aligned with the bottom edge of the housing lid, and the front vertical edge of the shield's interface will be aligned with the parting line of the lid. Figure B4 shows the proper alignment.

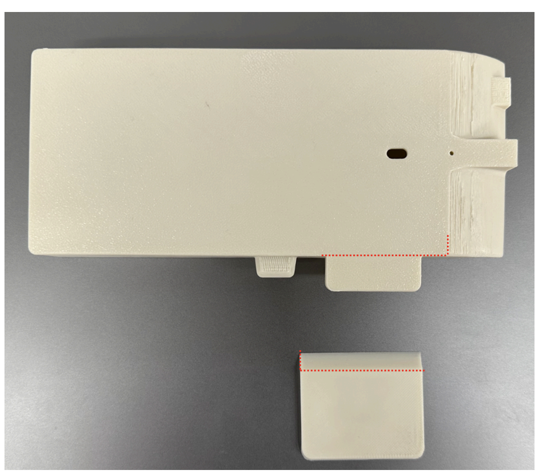

Figure B4: Main housing and radiation shield alignment

b. **Main Housing Lid**

Painter’s tape and super glue are recommended. The radiation shield’s bottom edge of the interface will be aligned with the bottom edge of the housing, and the front vertical edge of the radiation shield’s interface will be aligned with the nose cone and plat interface. Figure B5 shows the alignment and the edges that are to coincide.

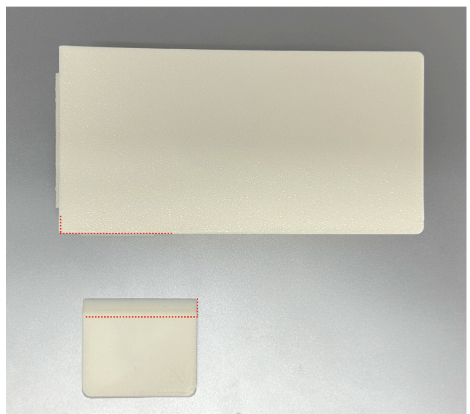

Figure B5: Main housing lid and radiation shield alignment

2. **Attach Arrow Shaft**

The friction between the arrow shaft and the dedicated slot is often enough to hold the entire tethersonde package together; however, if the fit has clearance, it is advised that the user lightly apply adhesive to the shaft before inserting it into the sonde.

3. **Installing Thermal Anemometer**

a. **Attaching to Anemometer Arm**

The anemometer node can be attached with 2x 2mm screws at a length of 5mm. Ensure the correct orientation as seen in Figure…

b. **Inserting into the main housing**

The thermal anemometer's arm is extended away from the body. Due to varying FDM printer resolutions, the addition of adhesive to the interfaces may be required for secured positioning. Before inserting the arm, ensure the JST wire is inside the arm and the male port makes its way into the sonde before applying the adhesive coating.

4. **Thermocouple installment**
    a. **Female Connector/ Thermocouple Wire**

    Begin by attaching approximately 150mm worth of thermocouple extension wire to a female plat pin connector. With the female connector on the outside of the body, feed the extension wire through the designated pass-through port, refer to Figure B2. Fasten the connector using a 2 mm flat counter sink screw at the provided hole location.

    b. **Male Connector/ Type E Wire**

    Create a small loop at the junction with the bare type E wire, leaving 50-75mm of wire on both ends. Connect the thermocouple wire to the male flat pin connector. Before closing the connector, feed the wire through a coffee stirrer cut to length, exposing approximately 5mm of the loop. This will help protect the very frail wiring. Thermocouples require correct polarity connections; the **red circular sticker indicates which end of the thermocouple is negative**.

5. **Installing PCB**
    a. **Cautions**

    The PCB is a very tight fit with all of the components attached. One should position the RevP, SHT, and LED into positions first. Here, you can slide the PCB down from the top of the housing into the correct position where the BMP and the IMU are seated correctly with 3mm of clearance from the top. Removing the SHT can also aid in the installation process. The RGB LED may require movement in order to reach the fastener hole.

    b. **Plug-ins**

    Before fastening the PCB to the housing, connect the RevP and thermocouple to their respective breakout boards. The RevP uses a unidirectional JST port and plugs to ensure they are correctly connected. For type E thermocouples, in the USA, the **purple wire is positive,** and the **red is negative**.

    c. **Fastening**

There are two locations where the PCB can be fastened to the housing: the top left corner and a hole underneath the LED. A 3mm ∅ screw that is a maximum of 5mm long. The threads of the screws act as a tap for the hole due to the soft ASA filament.

6. **Inserting Battery**
    a. **Tips and Tricks**
        i. It is recommended that a battery cable extender be permanently fixed to the PCB. This extends the battery JST, so plugging in the removable lipo battery is simpler.
        ii. Adding a pull tab to the battery can aid in removing the battery from the housing.
7. **Placing the Lid On**

    Step 1: Place the front lid lip underneath the nose cone.

    Step 2: While maintaining step 1, slowly close the lid until contact between the lid's SHT cover (shown in Fig B3) and the main housing wall touches. Push the lid's SHT cover into the body and or push the main housing wall away from the body so that they overlap.

    Step 3: Continue closing the lid until fully seated with the housing. By now, the SHT cover should be in its proper place.