Sparkfun OpenLog Artemis IMU
Calibration Guide:

ROS Package (by: Francis Le Bars): https://github.com/lebarsfa/razor_imu_9dof
(The package was built upon the previous Razor model, so do not worry about the "Razor" name, it will work for the Openlog Artemis :))

Original Calibration Guide*:* http://wiki.ros.org/razor_imu_9dof

This guide is written to improve and better clarify the calibration steps in the guide above based on my own experience calibrating the OLA Artemis IMU.

Calibration:

1. Install the Processing Application (necessary for magnetometer calibration and helps in visualizing the calibration process): https://processing.org/download/
2. Clone the following repo https://github.com/Razor-AHRS/razor-9dof-ahrs/tree/master/Processing

3. Open the .pde files from the Processing folder from the above repo. The Magnetometer_calibration.pde file will allow you to calibrate the magnetometer using the extended magnetometer calibration, which is far more accurate than the standard calibration. The Razor_AHRS_test.pde file will allow you to visualize the 3d calibration of your IMU which can help to verify the calibration.

   You can also visualize using the ROS package's display node (this has only worked so far using Ubuntu 20.04 and ROS noetic so far due to problems with the python-visual to vpython migration). You also have the magnetometer_calibration.pde already installed in the ROS package from Francis Le Bars here: (https://github.com/lebarsfa/razor_imu_9dof/tree/indigo-devel/magnetometer_calibration)

4. Open the my_razor.yaml file from your ROS package if possible on your Jetson, since it will be easier to quickly record the calibrated values into this file. Make sure to set the extended magnetometer calibration to true to actually use your calibrated magnetometer values! (we have personally forgotten to do this more than once I'm afraid :/, which caused some confusion)

   my_razor.yaml

   a. `calibration_magn_use_extended: true`

To calibrate the Sparkfun OLA Artemis, we will need to calibrate the accelerometer, gyroscope, and magnetometer. As of now we will be doing this with an external machine that can run the Arduino IDE and Processing applications.

Make sure your IMU can be held stable and is within the environment it will reside for best calibration results. Have the USB-C port face downwards and towards yourself if you are holding it. This is the default setting for the x-axis to the primary “forward” direction. (This can be changed within the configuration, reference steps at end of guide)

Accelerometer Calibration

1. Open the Arduino IDE serial monitor with the appropriate port to the OLA Artemis IMU.
2. First, we are going to calibrate for the max/min values of gravitational force on each respective axis. Use the command #oc in the serial monitor to set into calibration mode for the accelerometer.
3. Hold the IMU stable, then remember to make all movements as **slow and careful** as possible because it is extremely sensitive to changes in acceleration.
4. Point the IMU downwards slowly, you should see the x-max growing (second value). Record this max value. Reset using #oc if the process was not clean (it takes some time to get used to this so #oc frequently if needed)
5. Reset using #oc and tilt upwards to get the x-min value.
6. Repeat these steps except tilt to the side for the y-max and then opposite side for the y-min (similar to rotating a skewer). Record the values
7. For the z-axis, slowly move the board up and down and repeat in a similar fashion to the x and y axes calibration if needing to reset. Record the values
8. **“CAUTION:** You have to be really careful when doing this! Even slightly tapping the board with the finger messes up the measurement (try it!) and leads to wrong calibration. Use #oc very often and double check your min/max values)” (good advice from the original guide)
9. Input all the values into the my_razor.yaml configuration file you created in the ROS package you previously installed.

Gyroscope Calibration

***This was well written from the original guide so it did not need to changed**

1. Lay the *Artemis AHRS* still on the table.
2. We’re still in *calibration mode* for the accelerometer. Send #on **twice**, which will move calibration past the magnetometer to the gyroscope.
3. **Wait for 10 seconds, and do not move the *Artemis AHRS*.** It will collect and average the noise of the gyroscope on all three axes.
4. You should now have output that looks like this:

```
gyro x,y,z (current/average) = -29.00/-27.98  102.00/100.51
-5.00/-5.85
```

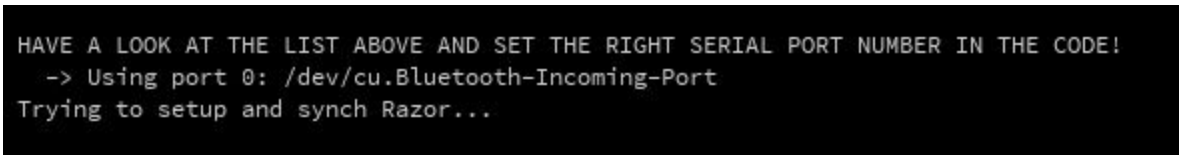
If you think you messed up the measurement by shaking or moving the board, you can reset by sending #oc.

Gyroscope Calibration

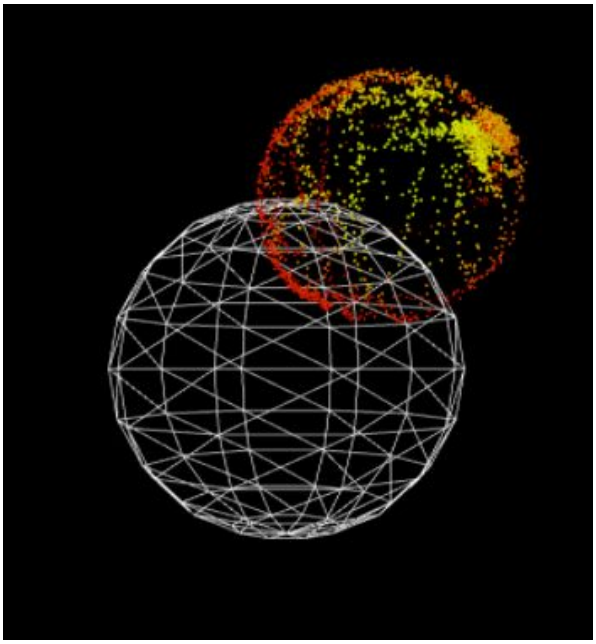
1. Close all the Arduino IDE serial monitor applications to open the port to be read by Processing application
2. Open the $(find razor_imu_9dof)/magnetometer_calibration/Processing/Magnetometer_calibration.
3. Follow the steps at the top of the .pde file to download the appropriate EJML jar and place it in a created libraries/EJML/library/EJML.jar folder.

> IMPORTANT: You have to install a library, before this sketch can be run!
> We're using EJML for matrix math, because it's really fast: http://code.google.com/p/java-matrix-benchmark/ Also, it's released under LGPL, which fits well with our GPL. Get the library from: http://code.google.com/p/efficient-java-matrix-library/ (you only need the .jar file), find your Processing "libraries" folder (normally this is Processing/libraries in your user documents folder). Create a folder "EJML" inside "libraries",create a folder "library" inside "EJML" and put the .jar inside. Rename to EJML.jar. So you should have "libraries/EJML/library/EJML.jar". Restart Processing and you're good. More info on installing libraries in Processing http://wiki.processing.org/w/How_to_Install_a_Contributed_Library
>
> Tested to be working with EJML 0.17 and 0.23.

4. Run the .pde file and then close it. Scroll up in the messages and figure out what number port the IMU is listed on. (This picture has the wrong port set so the error message would appear).



5. Set the Port number on this line at the top of the file
    a. `final static int SERIAL_PORT_NUM = 13`
6. Run the sketch and rotate the IMU around in all orientations in all 3 axes. You will start to somewhat of a circular shape form somewhere in the sketch, the location is dependent on the magnetic influences in the area you are calibrating within. Make sure to calibrate closest to the most realistic environment possible in which the IMU will be mounted to best account for these magnetic forces. You should end up with something like the following:

a. 

7. Hit space bar to end, and record the values printed at the bottom of the console. Input these values into the my_razor.yaml file. Remember to set the following to actually see changes with your calibration: `calibration_magn_use_extended: true`

Now your OLA Artemis IMU should be properly calibrated and outputting reliable values! Restarting the IMU is always a good first step if further issues arise. Also, recalibrating based on the environment the IMU is key in ensuring accurate readings. Please reach out for further questions! I will be improving this guide as we continue this process ourselves and from your feedback. Thank you.