

## XO-NANO Development Kit User Guide

### Introduction

This is a step-by-step guide for XO-NANO's Smartfoam development kit (dev kit). Your dev kit should include:

- (1) XO-NANO BLE Microcontroller and battery
- (2) Smartfoam Impact sensors - one insole sensor which measures the Smartfoam's response in 6 locations and one single zone 4"x4" impact sensor for additional testing
- (1) Smartfoam Pressure sensor – this is a single zone sensor that needs to be attached to a separate device (Analog Discovery, Arduino, etc.) to be measured

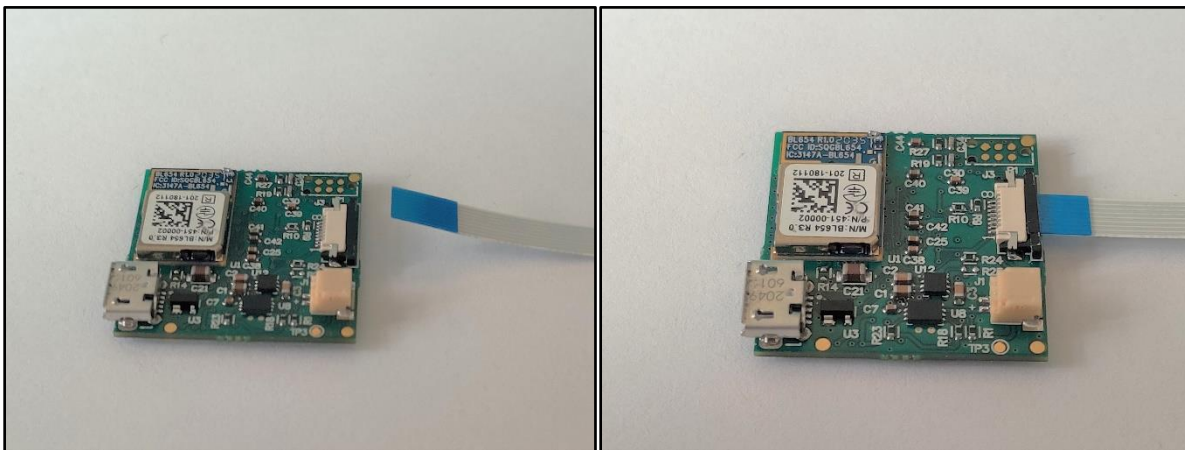
This guide will instruct you how to set up each Smartfoam sensor, connect each Smartfoam sensor to the microcontroller to begin recording data, and download the data onto a computer. This document also has contact information for any concerns or questions that may arise from using the sensors in the dev kit.

An up-to-date version of this guide is available in digital format at: <https://bit.ly/3o46gZA>

### Smartfoam Impact Sensor

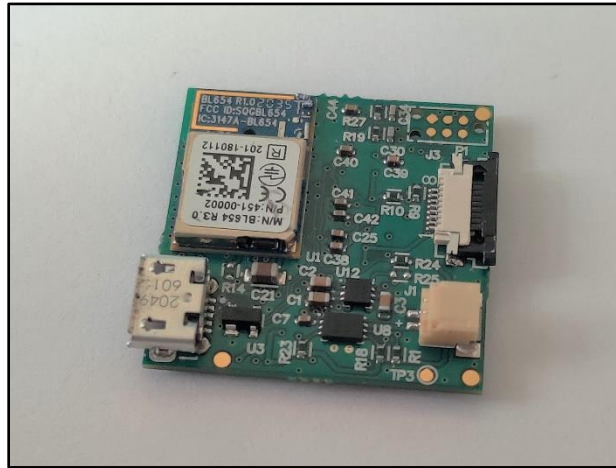
The Smartfoam impact sensor is the 4"x4" foam assembly with a single wire extending from the electrode below the foam. As this sensor is impacted, it creates a voltage that is proportional to the magnitude of impact. The electrode underneath the Smartfoam is used to measure the voltage from the Smartfoam to the microcontroller. This particular sensor is set up with one impact zone, so the location of impact on the sensor does not matter. As you impact the Smartfoam sensor, the wire coming from the electrode should be connected to the microcontroller through the ribbon cable.

The microcontroller should be powered by a battery and should be connected to a Bluetooth antenna in addition to being connected to the Smartfoam sensor. To connect the ribbon cable to the microcontroller, the metallic pins on the ribbon cable insert into the Zero-Insertion Force (ZIF) connector on the microcontroller with the pins facing down, toward the green board (see Figure 1).



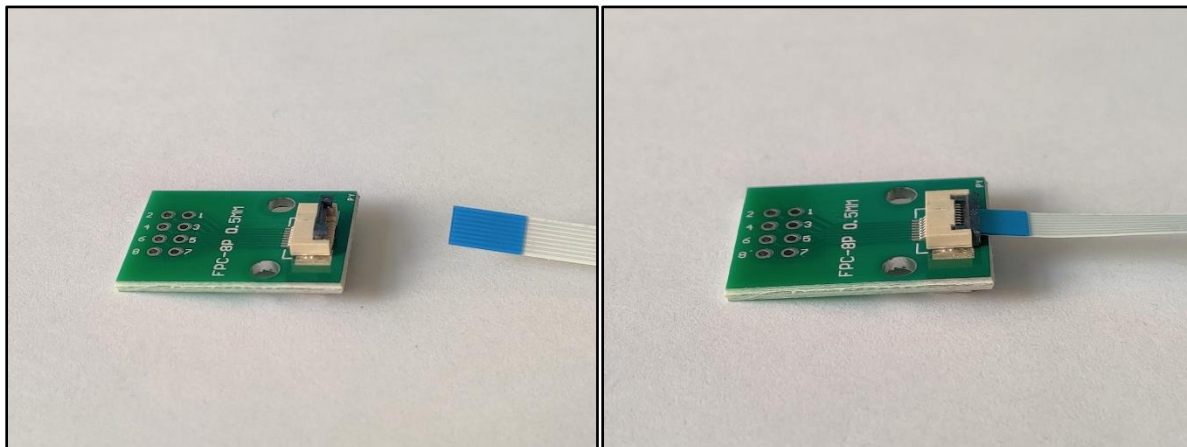
**Figure 1.** Insert ribbon connector into ZIF connector on the microcontroller board.

The ZIF connector's black tab should be gently pulled towards the outer edge of the board to unlock the mechanism, then the ribbon cable should be inserted, and the black tab should be gently pressed back into place.



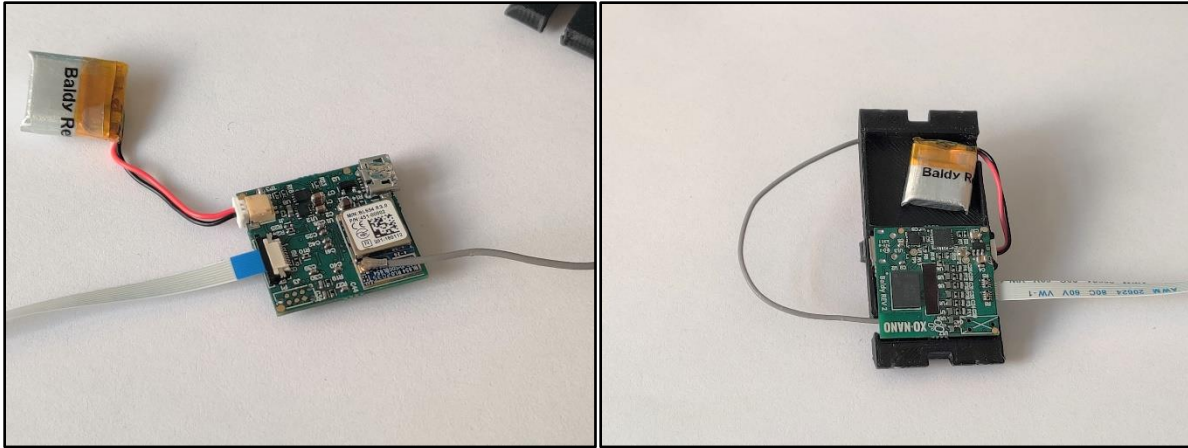
**Figure 2.** Pull black portion of ZIF connector away from the board to insert the ribbon.

The other side of the ribbon cable inserts into the breakout board's ZIF connector. This side inserts into the breakout board the same as the microcontroller: the metallic pins insert into the ZIF with the pins facing down. However, the black tab on the ZIF connector should be gently folded up 90°, instead of pulled out.



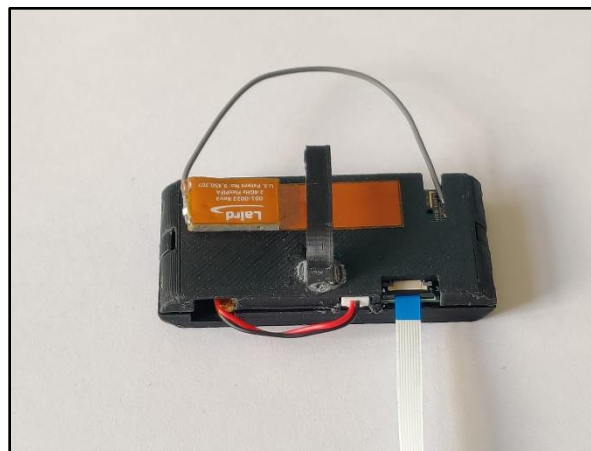
**Figure 3.** Lift black portion of ZIF connector on break out board to insert ribbon.

The microcontroller may be attached to the antenna, battery, and Smartfoam sensor while the microcontroller is in the case. You may also attach all the components before placing the microcontroller in the case.



**Figure 4.** Connect battery, ribbon, and antenna before or after inserting microcontroller into the case.

Before impacting the sensor, be sure to anchor the microcontroller to a steady surface using the clip on the case or any other method fit to stabilize it. To attach the antenna, see Fig. 10, and ignore the placement of the antenna wire in the channel in the figures. The antenna is adhered to the outside of the microcontroller case and its wire will remain outside of the case as it's attached to the microcontroller.



**Figure 5.** Attach the antenna to the exterior of the microcontroller case.

To attach the battery, see Figs. 13 and 14, again ignoring the placement of the battery wire in the figures. The battery will stay in the battery compartment with the wire extending to the microcontroller.

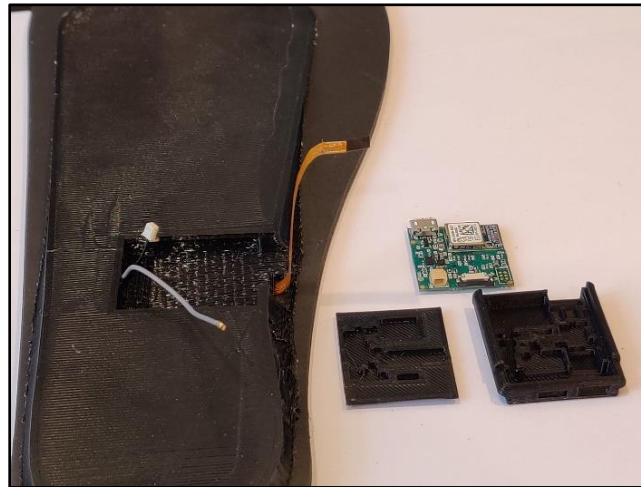
To collect data, the microcontroller should be connected to a charged battery (the battery can be charged by plugging a powered micro-USB cable into the microcontroller), in range of an iOS device with Bluetooth activated and the XOnano DevKit app open. Read the section “Connecting to the App” for further instructions.

The impact sensor is attached to the first pin on the breakout board of the ZIF connector. This pin will correlate to the “Pad 1” dataset in the recorded data. You may attach other (up to six total) wires to the breakout board and the microcontroller will record their data at the same time.

## Smartfoam Insole

The Smartfoam insole is the most complex assembly: two layers of foam, a TPU insole that supports the foam and electrode, the battery, the Bluetooth antenna, and the microcontroller case. This connection is made through an eight-pin ribbon cable (although only six are active). To assemble the microcontroller into its case, follow these steps:

1. Begin with the Smartfoam insole, the microcontroller, and the two case pieces. Ensure that there are three leads coming from the insole: the ribbon connector, the antenna wire, and the battery connector (Fig. 6).



**Figure 6.** Required components to assemble insole.

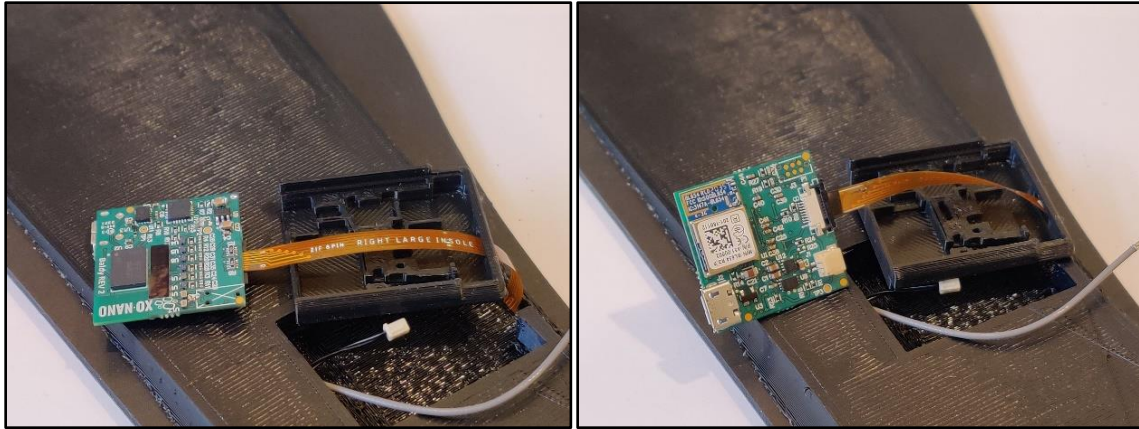
2. Thread the ribbon cable through the mid-line slot in the larger half of the microcontroller case (Fig. 7).



**Figure 7.** Thread the ribbon connector through the case.

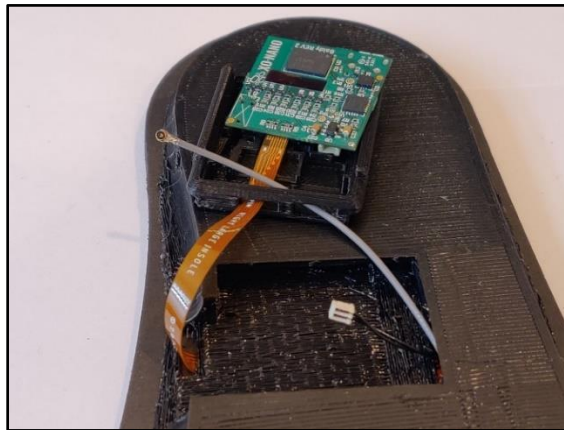
3. Connect the ribbon cable to the microcontroller. The metallic pins on the ribbon cable insert into the ZIF connector on the microcontroller with the pins facing down, toward the green board (Figure 8).





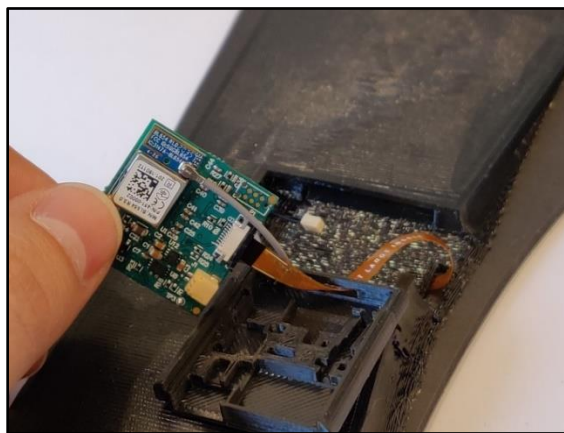
**Figure 8.** Insert ribbon into ZIFF connector on the microcontroller.

4. Thread the antenna wire through the lateral slot in the back of the case. Note: It is easier if the antenna wire passes over the ribbon connector (see Figure 9), but this is not necessary.



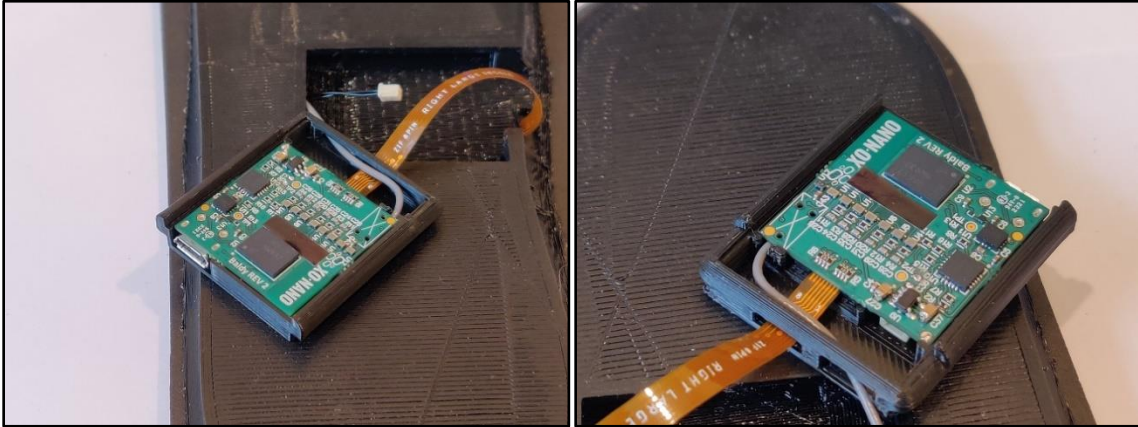
**Figure 9.** Thread the antenna wire through the case.

5. Attach the antenna wire to the Bluetooth module on the microcontroller as shown in Figure 10.



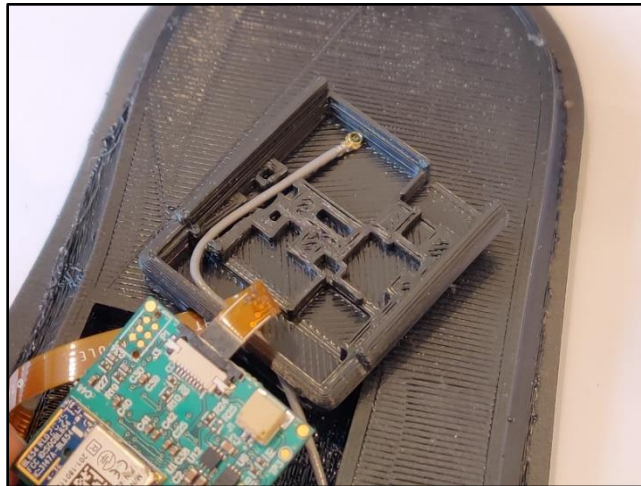
**Figure 10.** Attach antenna wire to microcontroller.

6. Place the microcontroller into the case such that the antenna wire and ribbon connector are exiting on one end and the USB connector is facing out on the opposite side.



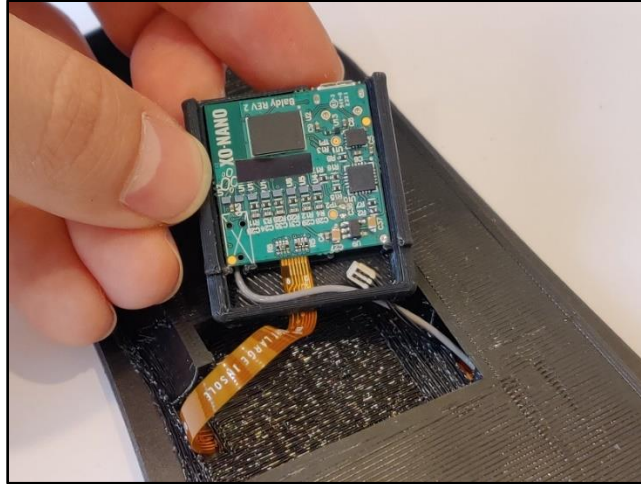
**Figure 11.** Place microcontroller in the case.

NOTE: The antenna wire rests in the channel under the microcontroller shown in Figure 12.



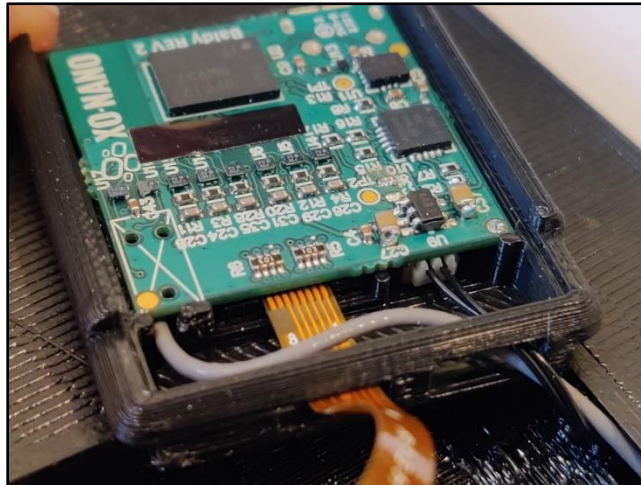
**Figure 12.** The antenna wire lays in this slot in the case.

7. Thread the battery connector through the lateral slot (the same hole as the antenna wire).



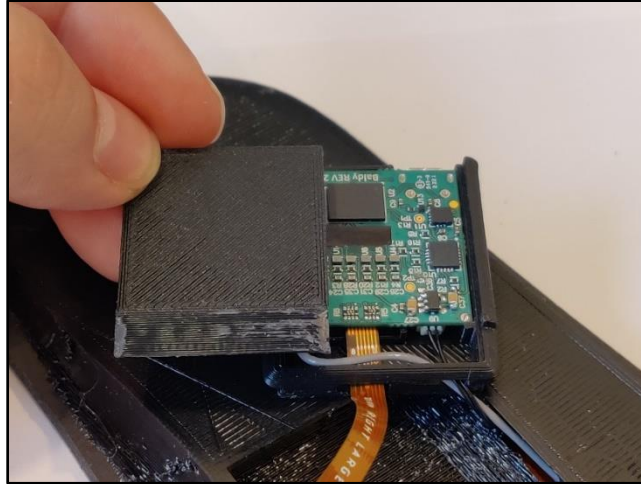
**Figure 13.** Thread the battery connector into the case.

8. Plug the battery connector into the microcontroller. Note: Tweezers may be helpful in this step.



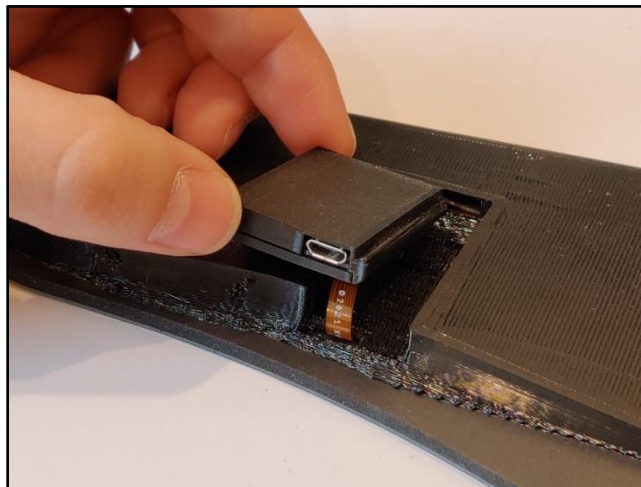
**Figure 14.** Plug the battery into the microcontroller.

9. Place the top cover on the case such that the slanted edge is over the ribbon and the battery and antenna wires.



**Figure 15.** Place the cover on the microcontroller and case.

10. Align the case with the insole so that the USB connector is facing out and the ribbon folds back on itself under the case.



**Figure 16.** Align the case in this orientation to the insole.

11. Place one side of the case into the insole so the lip on the case fits into the corresponding cavity in the insole.





**Figure 17.** Push one side of the case into the insole.

12. Slightly flex the insole to allow the other lip on the other side of the case to fit into the opposite cavity in the insole.

**WARNING:** Be careful not to bend or crimp the ribbon cable in this step.



**Figure 18.** Flex the insole and push the other side of the case into the insole.

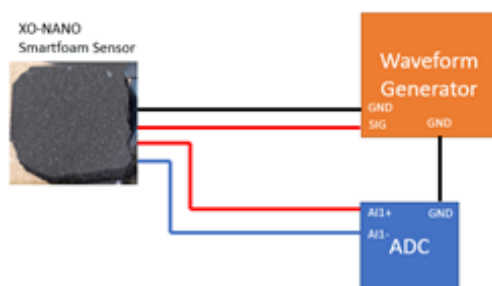
Once the insole is assembled, charge the battery by attaching a powered micro-USB cable into the charging port on the microcontroller. After several hours of charging, it should be ready for use. If the battery is charged and plugged into the microcontroller, it should connect to an iOS device in range with the XOnano DevKit app open. Read the section “Connecting to the App” for further instructions.

### **Smartfoam Pressure Sensor**

The Smartfoam pressure sensor is available as a 1”x1”, 2”x2”, or ½”x½” foam sensor with four wires extending from the electrode below the foam. We can measure the static pressure of objects using this

sensor while it is attached to an Analog Discovery device, an Arduino, or other data collection methods. The Smartfoam and the electrodes form a capacitor whose impedance changes with pressure.

Figure 19 shows the generalized connection scheme for the Smartfoam pressure sensor. The shortest black wire connects the ground between the waveform generator and an ADC, and the longer black wire grounds the pressure sensor. The long, single blue wire should be connected to the negative analog input of an oscilloscope or some other ADC. Either of the two red coupled wires should be attached to the supply of the waveform generator and the other to the positive input of the ADC. We prefer an Analog Discovery as both the waveform generator and ADC automatically share GND and the same internal clock when measuring pressure.



**Figure 19.** A generalized schematic for the Smartfoam pressure sensor. For the best results, the Waveform Generator and the ADC should use the same clock.

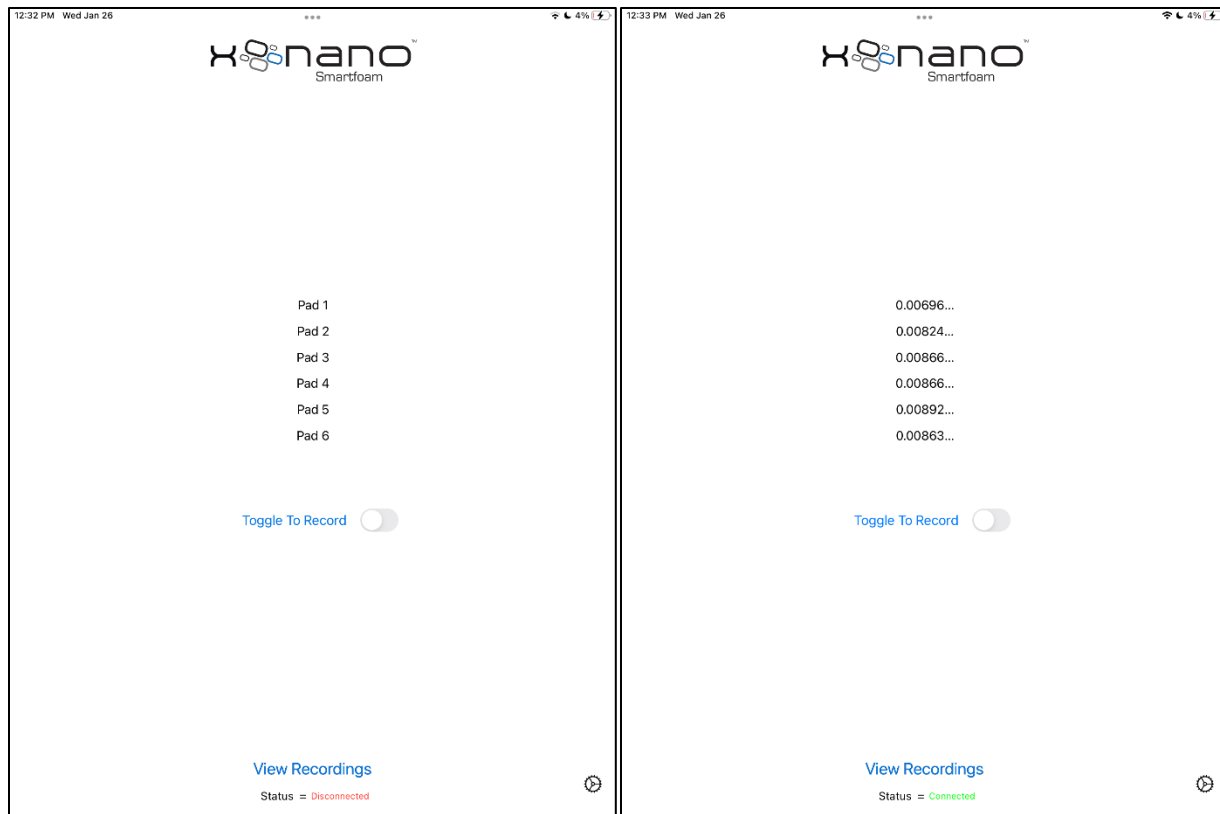
We suggest having the waveform generator power an AC current in the form of a 1kHz square wave with 2.5V offset and 2.5V amplitude (a 5V PWM). The ADC sampling frequency should be at least double the supply's frequency and should calculate the root-mean-squared (RMS) of the measured voltage. The RMS of the voltage will change proportionally to the pressure applied to the foam.

For detailed instructions for setting up the pressure sensor with an Analog Discovery 2, an Arduino Uno, or other method, please see the "Pressure Sense Instruction.docx" document on [XO-NANO's Github page for the Dev-Kit](#).

### Connecting to the App

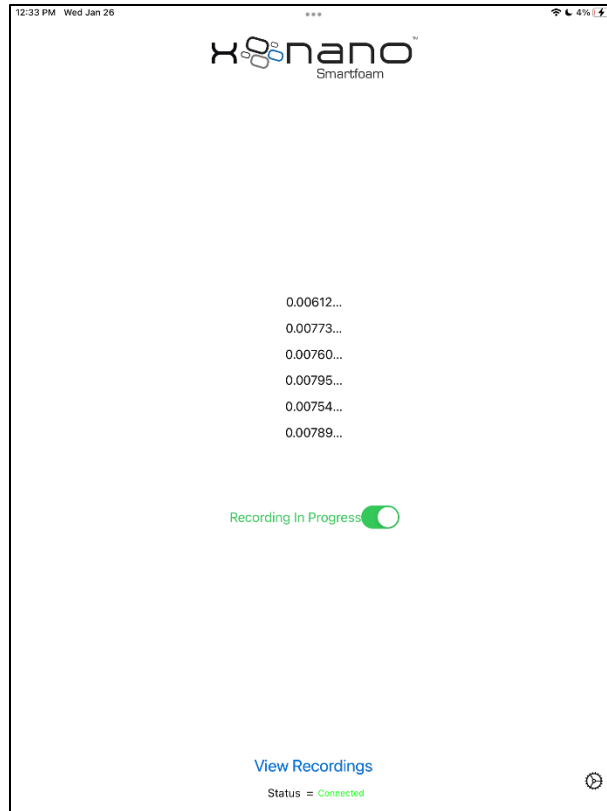
To download the Beta release XOnano DevKit app onto your iOS device, visit the following link on your iOS device: <https://testflight.apple.com/join/mG4E2b9c> and join the beta program. This app is completely local to your iOS device and will not download any data to a cloud service.

Once the battery is plugged into the microcontroller, the microcontroller will begin transmitting data. After the microcontroller is plugged into an antenna and a charged battery open the XOnano DevKit app on your iOS device. Once the app has been opened and the microcontroller is powered by a battery and in range, then the "Status" at the bottom of the app interface will change from "Disconnected" in red, to "Connected" in green. For the best results, keep the iOS device less than ten feet from the microcontroller. See Fig. 20 for an example of the iOS device screen before and after connecting a microcontroller.



**Figure 20.** The iOS device screen before and after a microcontroller is connected. For a microcontroller to connect, it must be plugged into a battery and it must be in-range of an iOS device with Bluetooth enabled and the XOnano DevKit app open. Once it is connected, the “Pad 1...” labels will change to a display of the current voltage value, and the Status will read “Connected” in green, instead of “Disconnected” in red.

Once it is connected, press the toggle button labelled “Toggle To Record” if you wish to record data. When the toggle button lights up green, then the app is recording data to a newly created folder called “XOnano DevKit” in the iOS device Files app. Figure 21 shows the screen of the app while recording. You can easily reach this folder by pressing the “View Recordings” label on the home screen of the XOnano DevKit app.



**Figure 21.** The iOS device screen while the microcontroller is connected and recording. If recording is in process, then the six numbers above the toggle switch will be constantly changing, the toggle button will light up green (as shown) and will say “Recording in Progress”.

Once you press the toggle button and the app starts recording data, run the test you desire. When the test is complete, press the toggle button again and the recording will be saved as “Test#1” or “Test#x”, where “x” is the test number run. Each time starting a new recording will create a new .csv file that is named one number higher than the previous test. This recording is a .csv file saved to the previously mentioned folder. The test numbers reset each time the app is restarted and will begin rewriting the previous tests. So, if you must restart the app for any reason, either export the desired .csv files or rename them.

If the six numbers on the screen ever stop changing at a high frequency, then the recording will stop. However, the status will still read “Connected” in green. When the six numbers stop changing, it means that the antenna or battery have become unplugged, the microcontroller is out of range of the iOS device (try to stay closer than ten feet), or the battery is out of power. To solve this, make sure the battery is charged, the battery and antenna are connected, and the microcontroller is within ten feet of the iOS device. Restart the app and export your desired data, then begin the process of recording again.

To access the recordings, export the .csv files from the XOnano DevKit folder to your computer via email, Air Drop, or any other online file exchange system.



## Contact Us

If you have any concerns, questions, or need help troubleshooting one of the sensors in the XOnano DevKit, please reach out to one of our engineers. Below is contact information for help with the impact sensor, the insole, and the pressure sensor.

### *Impact Sensor and Insole:*

Trevor Christensen

Email: [trevor.christensen@xonano.com](mailto:trevor.christensen@xonano.com)

### *Pressure Sensor:*

Max Tree

Email: [Maxwell.tree@xonano.com](mailto:Maxwell.tree@xonano.com)