Inventory

* ESP32 Development Board (x1)
* FlexiForce Piezoelectric Resistor (x1)
* Cheaper Piezoelectric Resistors (x10)
* Neoprene Rubber (6 in x 6 in x 6 mm) (x10)
* Op Amps (x5)
* Breadboard (Electrical Parts Kit)
* Connectors (Electrical Parts Kit)
* Resistors (Electrical Parts Kit)

Verify Resistance-Force Chart Provided by FlexiForce (Drum Recognition Accuracy)



Chart 1: FlexiForce Resistance vs. Force Graph



Image 2: FlexiForce Recommended Circuit Diagram

Needed Materials: FlexiForce Piezoelectric Resistor, Oscilloscope, Resistor (RF = 500kOhm?), OpAmp, Power Supply (or ESP32), Connectors, Breadboard.

This test has 2 objectives:

1. Verify the sensor measures force by changing resistance.
	1. On Success: The resistance measured via oscilloscope will change as different forces are applied.
	2. On Failure: The resistance isn’t changing when force is applied. Verify setup is correct (power plugged in, correct circuit, etc.) and if still non-functional seek out a new sensor ASAP.
2. Verify the Force-Resistance Relationship matches the chart
	1. On Success: Measurement of 10 (or so) different forces match with the expected resistances seen in the chart. Since the relationship should be continuous, we can assume the intermediate points will at least be near their expected values.
	2. On Failure: Measurement of 1 or more test forces produces an unexpected resistance (not within reasonable error of expected). From here, we’ll need to measure a larger subset of forces (every 5 lb?) and graph our own force-resistance graph. Afterwards, assess whether this new chart is still satisfactory for our needs.

Testing Steps:

1. Setup the recommended circuit for the FlexiForce sensor (Image 2)
2. Hook the Oscilloscope to Vout
3. Place at least one Neoprene sheet over the sensor (to avoid direct contact and damage to sensor)
4. Measure the voltage at terminal Vout every 10 lb. Starting from 0 lbs and going to 90 lbs.
5. Calculate the corresponding resistance for each of these voltages
6. Compare these resistances to the expected values based on Chart 1
7. Assess Accuracy of the values in Chart 1

ESP32 Analog Signal Recognition (Latency)

Test 1

Needed Materials: Power Supply, ESP32, Connectors, Breadboard.

* The only objective of this test is to verify the ESP32’s ability to recognize analog signals as expected based on its datasheet.
	+ On Success: The ESP32 can recognize analog signals on its GPIO pins and say with reasonable accuracy the voltages on those pins.
	+ On Failure: The ESP32 doesn’t respond in any form to analog voltages being placed on its GPIO pins. From here, verify the setup matches our expectations and determine If the ESP32 has any means of communicating the presence of the analog signal externally.

Testing Steps:

1. Using the ESP32 Arduino IDE, program the ESP32 to send a simple signal to a serial monitor upon receipt of an Analog signal (above a certain threshold).
2. Verify the code works as expected using a digital signal first, then modify it for analog signals.
3. Connect the power supply to the measuring GPIO pin of the ESP32 using connectors and a breadboard if necessary.
4. Gradually move the power supply voltage up until it passes the analog threshold specified in the ESP32 program.
5. Verify that the ESP32 has or hasn’t sent the serial signal.

Test 2

Additional Needed Materials: FlexiForce Piezoelectric Resistor, Resistor, OpAmp.

* This test will verify the functionality of the entire hardware pipeline, from force being applied to the sensor to the ESP32 sending a signal.

Testing Steps:

1. Using the setup described in the “Drum Recognition Accuracy” test as well as the setup described in Test 1 above, combine the two by having the Vout pin of the “Drum Recognition Accuracy” test connect to the GPIO pin of the ESP32.
2. Modify the ESP32 code so that the “no signal” state matches Vout when no force is applied the voltage threshold needed to trigger a signal is between that resting voltage and the Vout voltage measured when a (moderate?) force is applied.
3. Apply the (moderate?) force to the sensor
4. Verify that the ESP32 has or hasn’t sent the serial signal.