PX505 Final Report

eTex: Electrical signal conversion of a triboelectric fabrics into a sound or light signal

Azmi Abbessi | Sébastien Barbero | Félix Roux

Index table

Abstract **2**

Introduction **2**

Related Work **4**

Demonstrator Architecture **5**

Analog and ADC 5

Embedded Algorithmic 6

Communication 7

Results and Validation Environment **8**

Choosing the target board 8

Sensor characterization 8

Signal detection on Arduino 9

MIDI Conversion 10

Wireless connection 11

Societal challenges of their innovation **14**

Conclusion **15**

Annexes **16**

A Quick MIDI Definition 16

Figures 17

Figure 1.1 : Triboelectric sensors 17

Figure 1.2 : Setup used to test the sensors 17

Figure 1.3 : Signals captured with oscilloscope 18

Figure 2.1 : Simple and traditional MIDI recording 18

Figure 2.2 : Aftertouch test 19

Figure 2.3 : Real acquisition of a sensor excitation 19

Abstract

As the field of wearable technologies is growing with the development of new electronic textiles, the question of its application in the arts can be asked. The developed technology is a response to this question, aiming at controlling musical and scenic elements through connected textiles.

Introduction

The field of wearable intelligent textiles is an emergent, promising market. This discipline aims to improve the usability and the number of uses of clothes, simply reinventing what a piece of textile really is. A precise use of this technology takes advantage of movement, and other phenomenons such as friction, to generate electricity.

GammaO is working in the field of triboelectricity applied to the textile industry. They create high-tech and high-quality clothes that are built to harvest energy generated by athletes' efforts. Then, emerged the idea of applying this technology to the events and arts market, as follow :

1. Triboelectric electrodes made by the company GammaO are placed on a surface of clothing or furnishing textiles, used by an artist or performer. These persons, by performing with the equipped textile, would activate the electrodes by friction, thus generating an electric signal that could be then used and interpreted.
2. The generated signal would be processed and/or acquired by a digital electronic card. This chip would embed an algorithm that could interpret different components of the signal, and generate an artistically usable output.
3. The output signal, processed according to the standard communication protocols between electronic instruments, controllers, sequencers, and music softwares, would then be used to control sound triggers, light events, scene changes... All according to the performance of the persons, while being very stealthy and non-intrusive to the performers.

In other terms, we could imagine a jacket, equipped with GammaO electrodes, built for a dance performance or a ballet. As the dancer would simply dance as usual, the movements and frictions generated would be picked up, interpreted, and transmitted to a control desk, which could trigger light changes as the dance would get more and more intense. The usage of a neutral and universal protocol of communications made specifically for the music industry will also ensure cross-compatibility, and the ability of the device to control nearly anything that could be found on a scene.

Related Work

Triboelectricity is generated when two materials get in contact. Due to the momentary proximity, atoms from each material can exchange electrons, whether these electrons came from static charges already present in the material or electrons that were moved from their original atoms due to the friction movement. These exchanges to balance the charges create a temporary potential difference that can be used to generate small signals without the need of a power supply.

A first use for triboelectricity could have been energy harvesting, but due to the small voltage generated, it is mainly used to generate signals.

The applications of triboelectricity were studied after Wang Zhong Lin, a professor in nanoenergy and nanosystems, invented the TENG, TriboElectricity NanoGenerator. The TENG is a way of weaving conductive and triboelectric-friendly materials together to create a signal generator. There are two main domains currently using and innovating in triboelectricity : Smart homes and Smart clothing.

The purpose of the TENGs in these cases is mostly to detect pressure. For example, friction can be used to generate a command signal to turn on the lights, or in the wearable fields, as a sensor placed in the shoes, the number of steps can be detected while walking or running.

As some textiles can also generate triboelectricity while bending, body movements can be detected and tracked that way.

Demonstrator Architecture

This demonstrator, for easier management, and in conjunction with the company's responsible, Pascal Weber, has been separated in 3 distinct parts, all with different end goals.

Analog and ADC

As this project includes the use of pretty experimental sensors, as well as a potentially noisy and harsh environment (*f.e.* in the case of sweat, a lot of friction would be lost), we feel the need to dedicate a whole part of the project to the acquisition of the signal.

There is definitely a need to heavily filter the input signal to the Arduino board, as the sensor seems very sensitive to noise, and especially to 50Hz perturbations. We also noticed that a lot of the signal was actually useless to the artistic interpretation, as no fast vibrations were meant to be captured (we only want to capture the touches and frictions of the human body with the sensors, and no dancer will ever "rub" the sensor at 1000Hz !). We can also question the use of vibration-sensitive sensors in an environment where there is potentially very loud music, and heavy basses that could easily interfere with the system. These sound waves carry a lot of energy, and are within the range of the useful information we want to capture out of the sensors. However, this could not be tested in a real situation yet.

We also chose a demo-board with well-suited ADCs, that makes for an easier measurement of the signals. We are also constrained in the number of sensor cells we can put on, depending on the number of ADCs our card has. This can vary as we could choose to implement our system on a different but compatible card, but as of now, we use 4 out of the 6 ADCs our card has, on a custom "monster" (a prototype) vest which has 4 sensors embroidered to it, meant to be worn by a dancer.

M. Weber also gave us a homebrew 6-channel analog 50Hz Notch filter, that we can use in our final architecture to avoid the sensor's signals being too sensitive to its RF surroundings.

This is the part that is the most exciting, where we could innovate the most. We are capturing the signals that are collected from the sensors, and treating them to be able to output MIDI messages that are usable for other controller's softwares or hardwares.

As we're trying to capture the whole dimension of a performance and to extend it *via* our sensors, we thought of multiple algorithms and ways of translating movements picked up by the sensors. The version implemented as of now seems to be the most usable, and robust one out of our idea pool :

- If the sensor is in an *inactive state*, the associated MIDI note is deactivated. The algorithm waits for an input strong enough to wake up the MIDI note.
- If the threshold is reached, a MIDI Note On message is sent, with a velocity proportional to the strength of the input. This event starts a timer, and the note will remain active for that amount of time.
- During that time, if the sensor is manipulated or interacted with, this will generate MIDI Mod Wheel messages : that allows us, on the receiving end, to modulate certain elements of the response of the MIDI messages.

  For example, we can assign the volume of a synth to the Mod Wheel : that way, the output sound when manipulating the sensor will be evolving with time.

- Lastly, when the timer is finished, a MIDI Note Off message is sent, and the algorithm can go back to the *inactive state*.

Note how this algorithm can be repeated over an infinite amount of sensors : we just need to be able to have a decent sampling frequency for all the sensors, enough memory to save all the useful samples as we go… But that allows us to just apply this algorithm to the four sensors on our final device.

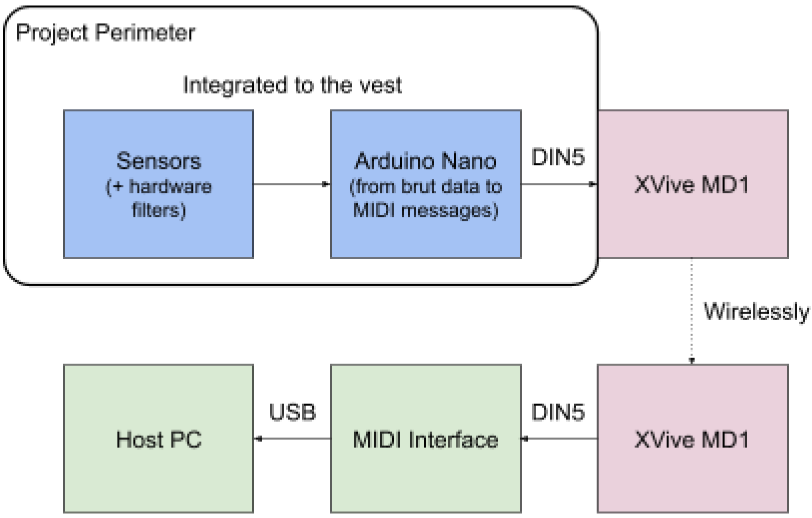
The very nature of the MIDI protocol also allowed us to eliminate the need for a driver on the client side of things : as it is so flexible and is just a "meaningless" information conveyor, we could just assign the algorithmically fixed messages to different events to fit the needs of the artistic operation.

Communication

If we were to develop a textile that would be meant to stay put (*f.e.* a carpet, rug, or table cloth), we could use a simple MIDI DIM5 connection to link the device to the rest of the MIDI ecosystem.

As you may have noticed, our application case is to be worn by a dancer, so that immediately excludes the cable's solution. We identified different solutions to that problem, and certain of these have been realized :

- Firstly, we can still include a DIN5 connection to our prototype, but use a commercially-available solution to wirelessly transmit MIDI messages from the Arduino board. This solution was found in the shape of the MD1, made by XVive, and has proven to be really effective, all the while being very low-latency.
- We could have also built a direct Bluetooth connection from the Arduino board to the receiving computer, as Apple computers are compatible with that technology. However, this caused quite a few problems, the first of which being that we don't have an Apple computer on our hands to test this solution, and it was excluded to buy one for the project due to its price.
- Lastly, if the sensors were to be used in a more static environment, for example as a MIDI controller under the hands of a pianist (that doesn't move), the cable solution can be completely appropriate. We could even use the built-in USB serial connection in that case, completely freeing us from the DIN5 connector.



Demonstrator Architecture

Results and Validation Environment

Choosing the target board

We decided to turn to Arduino boards because they are easy to program and are smaller than others like STM32 or Rasberry. Notably, the Arduino Nano boards are ideal for the use we want since the embedded electronics will be worn with the jacket by the end user.

Sensor characterization

A first step to study the sensors was to determine the best material and structure that would give us exploitable results.

Each sensor is like a patch embroidered on a piece of cloth. They vary in the type of material weaved to make the contact surface, and in the shape of that contact surface. We firstly had a round patch made of one material, then squared, with two wire and alternative bands of two different materials.

Figure 1.1 : Triboelectric sensors

Here are some examples of the sensors we used on Figure 1.1. To test them, we used an oscilloscope to see the form of the signals generated. We connected the sensor under test to a 50Hz filter board provided by M.Weber. We were then able to proceed with the testing.

Figure 1.2 : Setup used to test the sensors

Above, Figure 1.2 shows how everything was connected. For each sensor, we observed the response to the following challenges :

- Friction, with one finger, then with the whole hand, moving along with the stripes then perpendicular to them for the square-shaped patches

- Pressure, by gently hitting the surface with one finger then with four fingers

These tests were made with bare hands and wearing gloves of 3 different textiles.

These tests allowed us to choose one type of sensor with the best exploitable results. Given that the digital 50Hz filtering will be perfect, the response signals we can use are of two types : a series of small peaks for a friction movement, a higher peak for a hitting movement. The generated signals are under 100mV, and are considered valid above 30mV for reliability.

Figure 1.3 : Signals captured with oscilloscope

These are the types of responses we obtained while assessing one material with the different actions previously mentioned.

Signal detection on Arduino

Following the tests results, M.Weber provided us with a new sensor to use on the target board, Arduino nano. Still filtered with the analog 50Hz filter, one sensor was connected to the board, on an analog input.

These analog inputs work with a 10-bits converter. The range is defined between a maximum that can be one of the internal power supply or an external voltage, and between 0V. We used the 0.6V internal as reference to have the best precision possible without using external power supply. This means a maximum value of 1023 for 600mV measured, 0 for 0V measured. A 30mV detection threshold means a value of 51.

Using a simple algorithm that displays a message and the measured value when the threshold is reached, we tried to detect.

Unfortunately, and for some undetermined reasons, the threshold varies from 150 to 180 depending on the temperature and hydrometry in our project room, making amplitude measurement imprecise. But peaks can be detected. Since amplitude measurement is unreliable to make a distinction between friction and hitting movement, a first idea was to count the number of detected peaks during a set period of time after a first detection.

This solution works as is but the main disadvantage is that it creates latency between the movement of the operator and the command to be processed by the main program, which is highly undesired for musical and scenic applications. A general rule of thumb is that any delay superior to 10ms will begin to be perceivable by spectators and performers, and we would have to go way beyond that threshold to count peaks.

MIDI Conversion

As our project suggests, we will try to generate MIDI data. The following screenshot displays a typical recording of a MIDI controller on our target software, Ableton Live 10 :

Figure 2.1 : Simple and traditional MIDI recording

In blue, we have the notes recorded in the forms of blocks, and in red we have their velocity, the strength at which they were pressed. Note that these recorded notes have no evolving capacity : once a key was pressed, no other element could change the volume, or other characteristics of the sound played.

Let's compare that to a test recording of a MIDI generation of our device :

Figure 2.2 : Aftertouch test

In black in the background, we have a note sequence playing, and in blue, we have an Aftertouch modulation. This is exactly what we're trying to achieve, as we can still keep the original velocities of the notes (not displayed here), but also make the characteristics of the played note change over time. Here, we could assign the volume of our synthesizer to the blue envelope, and we would have notes that would fade in and out !

For the following example, a sensor was set up, along with the algorithm presented in our Architecture. This is a complete reading of one excitation of a sensor :

Figure 2.3 : Real acquisition of a sensor excitation

We can see that we generated a high excitation at the beginning, followed by a small rest and another excitation. The values before and after the notes are artifacts generated by the end of a note, and are not harmful because this modulation is useless without a note to… Well, modulate.

Wireless connection

To achieve the final goal of project which is to find a wireless solution to send MIDI messages so that the player or the dancer will have more flexible moves

1. solution using XVIVE MD1

So a research part is really important to make our goal come true, we studied the newest technologies that are used in such subjects and finally we found that Xvive MD1 could be a good solution for wireless communication.

Xvive MD1 is a wireless MIDI device that allows you to transmit MIDI messages wirelessly between devices. It uses a 2.4GHz wireless connection to transmit and receive MIDI data, and it has a range of up to 30 meters (100 feet). To use the Xvive MD1, you will need to connect it to your MIDI device using a MIDI cable. Once connected, you can use the Xvive MD1 to transmit and receive MIDI messages wirelessly. The device has a built-in display that shows the current connection status, and it also has a button that allows you to easily switch between different MIDI channels. To transmit MIDI data wirelessly, you can use the Xvive MD1 in combination with a MIDI controller or other MIDI device. When you play a note on the controller or send a MIDI message from the device, the Xvive MD1 will transmit the message wirelessly to any other MIDI devices that are connected to it. Overall, the Xvive MD1 is a convenient and easy-to-use wireless MIDI device that can help you to transmit MIDI data between devices without the need for physical cables.

To connect the Xvive MD1 to an Arduino using the MIDI library (midi.h), you will need to follow these steps:

- Connect the Xvive MD1 to your Arduino using a DIN5 connector. The Xvive MD1 has a MIDI input and a MIDI output, so you will need to connect the MIDI output of the Xvive MD1 to the MIDI input of the Arduino using a DIN5 connector.

- Include the MIDI library in your Arduino sketch by adding the following line at the top of your sketch: **#include <MIDI.h>**

- Initialize the MIDI library by calling the MIDI.begin() function in the setup() function of your sketch. This will initialize the MIDI interface on the Arduino and prepare it for transmitting and receiving MIDI messages.

- In the loop() function of your sketch, you can use the MIDI.sendNoteOn(note, velocity, channel) function to send a MIDI "note on" message to the Xvive MD1. This function takes three parameters: the note parameter specifies the pitch of the note to be played, the velocity parameter specifies the loudness at which the note will be played, and the channel parameter specifies the MIDI channel on which the message will be sent.

2. Solution using MODULE BLE arduino

If we want to connect our MIDI device with an macOS pc to receive the MIDI signals via the software we will be using a Bluetooth module connected to the Arduino

To do so we should include SoftwareSerial library so we can send and receive Bluetooth data and obviously set the RX , TX pins and the data rate.

In a second step we should set our computer to receive the MIDI message. To do so we will need to use HAIRLess MIDI software. Here's a general outline of the steps we'll need to follow:

- Download and install the HAIRLess MIDI software on your PC.

- Use the SoftwareSerial library to communicate with the Bluetooth module and the MIDI library functions to send MIDI messages to the PC. For example, to send a note on message, we can use the sendNoteOn() function.

- Pair the Bluetooth module with your PC.

- Open the HAIRLess MIDI software and select the Bluetooth module as the MIDI input device.

- Use a MIDI software on the PC to receive the MIDI messages from the Arduino.

We propose also in case to work on ESP32 instead of an arduino and a bluetooth module that the cost of the product will be lower.

To ensure that this solution work correctly we have to include these two libraries <BLEMIDI_Transport.h> and <hardware/BLEMIDI_ESP32.h> and leave the code as it is because when we call the function MIDI.sendNoteOn(sensorNote2, 100, midiChannel) it will directly send it via bluetooth.

Societal challenges of their innovation

The direct impact that this project will have on society is a change in the way the artists can perform their show. Imagine this : the performer, by simply rubbing their hand against their clothes, will be able to change the ambient sound, light, or trigger light effects, fireworks, etc. This will make shows even more incredible, looking as if magic was at use.

But it is also a way to promote triboelectricity, as it is an energy harvesting technology. Private investors might see interest in this technology, which might help it spread and grow. This will result in a good environmental impact since the expansion of energy harvesting means less use of the energy production means currently in use and known for their big and negative environmental impact.

Solutions using these products generate little voltage and far too little current to be of any harm to people using it or getting in contact with someone equipped with it. However, they are an interesting field of research, and experimenting with them might lead to new sources of renewable energy !

Regarding environmental impact, as these clothes integrate electronics, it raises the questions of repairability, reusability and end of life of the products. How can they be maintainable, recyclable, given their intricacy ?

Conclusion

GammaO is a company that creates high-tech clothing using triboelectricity to harvest energy generated by movement and friction. They are considering applying this technology to the events and arts market by creating a jacket that can pick up and interpret the movements and frictions of a dancer, and transmit them to a control desk to trigger light changes.

This project requires experimental sensors and may be used in a potentially harsh environment, so a focus of the project will be on signal acquisition.

GammaO has considered using a DIN5 connection with their prototype, but instead will use the commercially-available MD1 by XVive to wirelessly transmit MIDI messages from the Arduino board. This solution has proven to be effective and low-latency. Another option that was considered was using a direct Bluetooth connection from the Arduino board to the receiving computer, which is compatible with Apple computers.

Annexes

A Quick MIDI Definition

From the official MIDI 1.0 documentation : "MIDI, the Musical Instrument Digital Interface, was established as a hardware and software specification which would make it possible to exchange information (musical notes, program changes, expression control, etc.) between different musical instruments or other devices such as sequencers, computers, lighting controllers, mixers, etc.

This ability to transmit and receive data was originally conceived for live performances, although subsequent developments have had enormous impact in recording studios, audio and video production, and composition environments.

This document has been prepared as a joint effort between the MIDI Manufacturers Association (MMA) and the Japan MIDI Standards Committee (JMSC) to explain the MIDI 1.0 specification."

Note : the MIDI protocol in its current shape has remained untouched and is still in use since… February 1996 !

Figures

Figure 1.1 : Triboelectric sensors

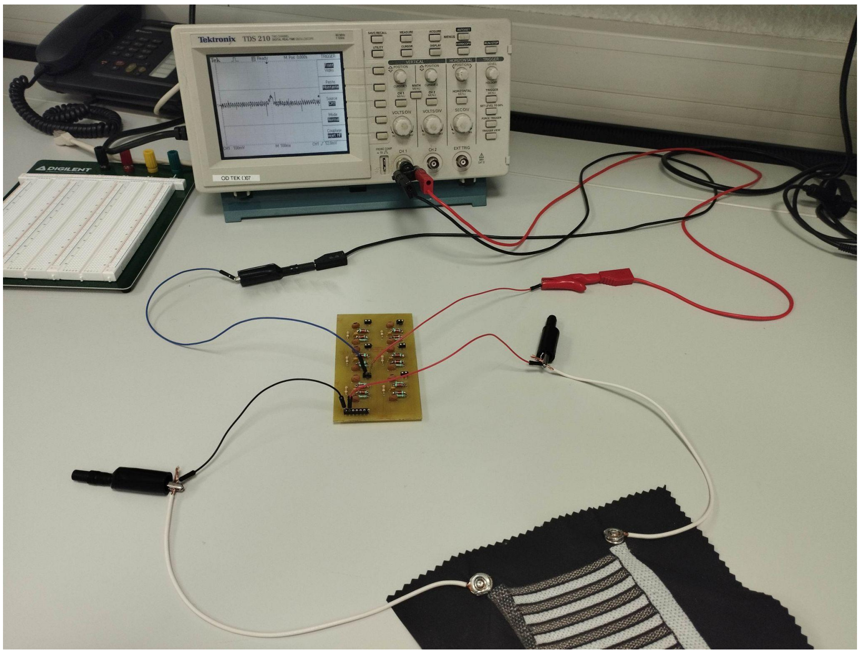


Figure 1.2 : Setup used to test the sensors

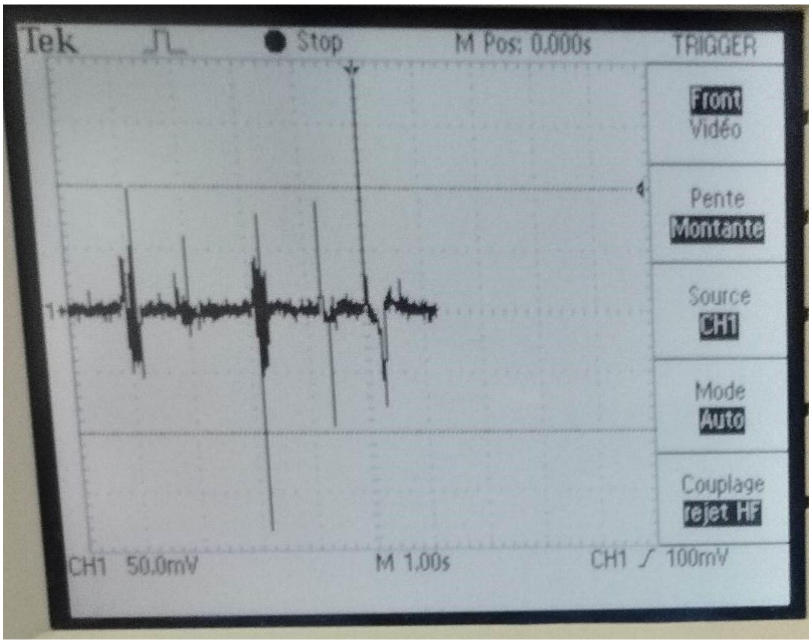


Figure 1.3 : Signals captured with oscilloscope

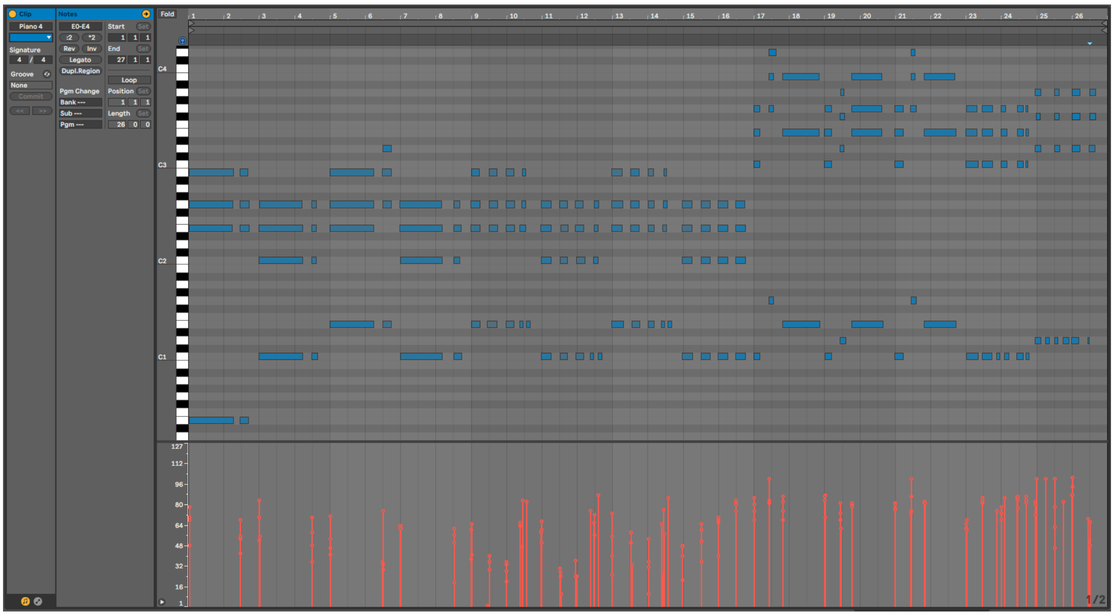


Figure 2.1 : Simple and traditional MIDI recording

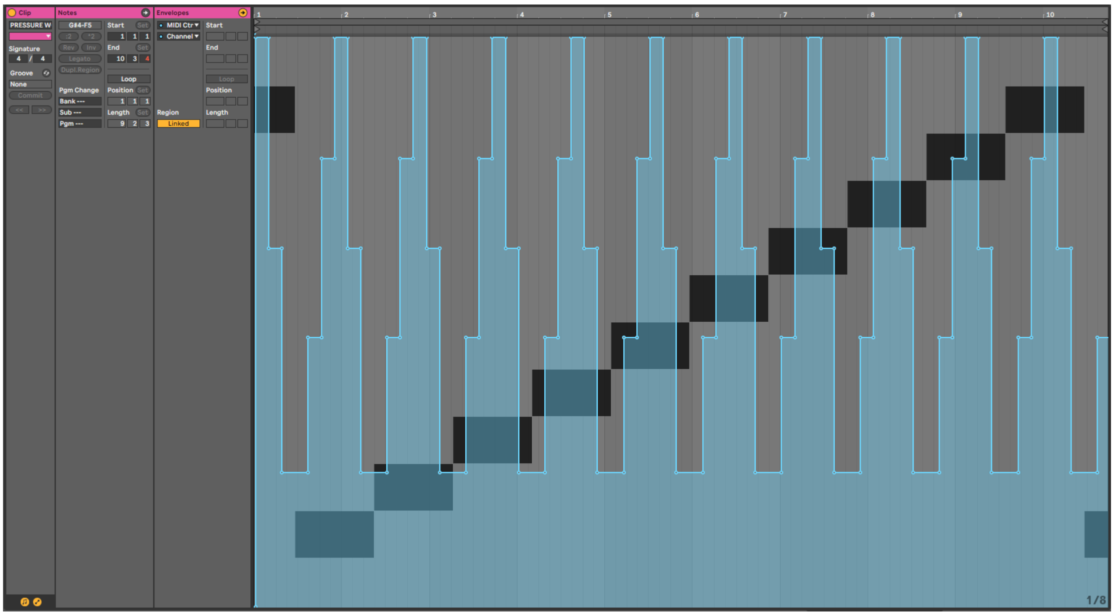


Figure 2.2 : Aftertouch test

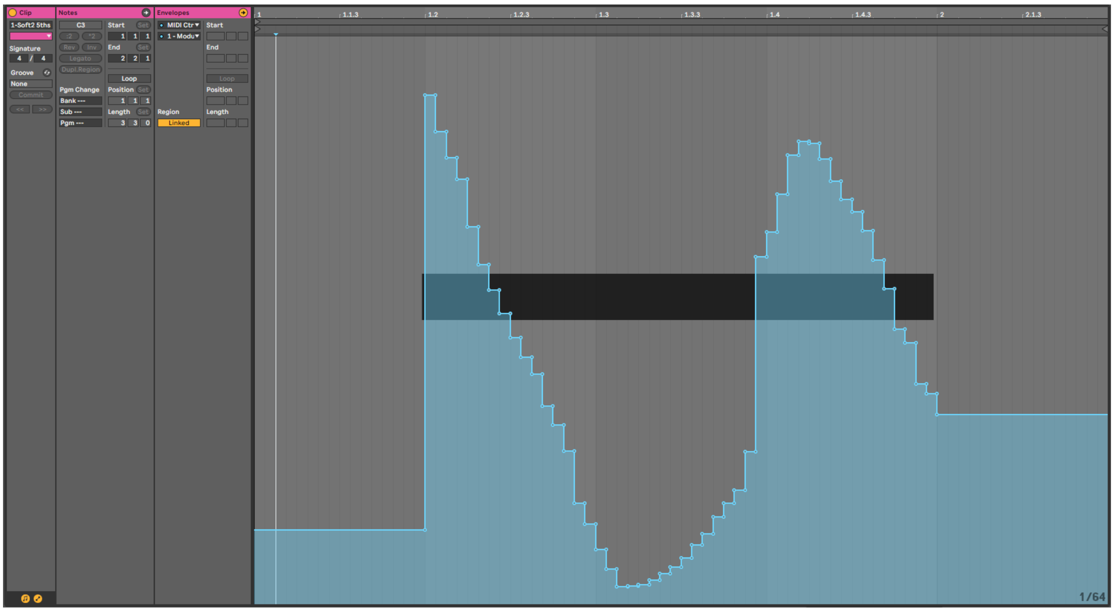


Figure 2.3 : Real acquisition of a sensor excitation