ENS 491-492 – Graduation Project

Final Report

Project Title: Wearable Music Instrument

Group Members: Barış Ülgen

Supervisor(s):

Murat Kaya Yapıcı

Özberk Öztürk

Date: 17.05.2020

1. EXECUTIVE SUMMARY

In any kind of art, the way you interact with the medium is just as important as the topic, type, genre etc. of the piece, because a unique interaction opens a unique creativity door to us, which affects the final product significantly.

This project focuses on that point and aims to create a new musical instrument with a new playing technique. This instrument will be wearable and will use surface EMG sensors and various mechanical sensors to harness simultaneous data from human body. This said data will be generated by the physical properties of the movement of the body parts.

Several electrodes will be placed on top of a long muscle to measure the electrical activity under the muscle fibers when a contraction is made in that muscle group. The extracted voltage values which are basic analog noise signals that are obtained from the electrical activity created by the neuromuscular system, will be transferred to a microcontroller via an EMG circuit, where it will be amplified by a factor of around 3000 and filtered. The mechanical sensors will be supplied with a voltage and directly connected to the mentioned microcontroller.

After the data obtaining stage is completed, this information will be stored, classified and sent to a specific software such as MATLAB, Processing and/or LabView (in most cases, working on microcontroller's interface would just be enough for this project's purposes).

When 'meaningful' responses are obtained after the data is processed through certain desired algorithms, it is possible to map all these data to a DAW (digital audio workstation) and control any musical parameter that the DAW offers to us. Not only musical properties such as pitch, velocity etc. can be controlled but control any plugin will be created and changed by these simultaneously.

Our motivation behind all of this is not just the desire to create something new but to create a device that will boost the creativity of an artist in a unique and freeing way. So far there are several examples of such "body-motion-controllers" but they don't generally use EMG for musical purposes and the market for commercially available MIDI controllers is lacking this kind.

2. PROBLEM STATEMENT

2.1. Objectives/Tasks

The main objective is to create a device that can transfer meaningful data to a computer from your body movements in order to create, change or modulate certain musical properties, while it is mounted on a flexible and wearable fabric.

2.2. Realistic Constraints

There are mainly two topics that affected the development of the project: scientific background and resource constraints (manpower and budget).

I had no prior knowledge of EMG sensors and low noise high gain amplifiers. I had to test and see how the sensors work, how they respond with different locations on body and with different supply voltages. Practically assembling parts and getting some decent results through trial and error was not as challenging as learning the theory behind several aspects of this project, which was a bit advanced for me and time consuming.

The other thing is that I had to do this project mostly alone. Working by myself generally motivates me but the workload made me progress through this project very slowly. Also, since the COVID-19 outbreak started, I have no access to any equipment, component and working environment, which made my project nearly impossible to complete.

3. METHODOLOGY

After my previous report, I have understood these concepts:

- Obtaining analog data from neuromuscular system via EMG sensors with a low noise high gain amplifier circuit.
- Created a data set and analyzed it to decide for the best location for EMG placements.
- Classifying and filtering all the inputs with a micro controller and outputting as desired.
- Establishing a connection between the microcontroller and the desired digital audio workstation software in order to control it with our body movements.

From that point, due to my main problem which was **differentiating different movements from the same muscle group**, I gave up the idea of using our muscles as regular musical instruments like piano or a guitar and started to think of a way to implement these sensors into practical instruments. After a very long period of thinking and trying, I have concluded that my instrument's type can be one of the following:

EMG sensors

- Percussion
- Chromatic (like a theremin)
- Clip/sample trigger

Other mechanical sensors

- Chromatic
- Parameter controller

Making a chromatic instrument like theremin is not practical since you can always use a theremin! Also I haven't paid attention but even conventional instruments have several built-in effects/modifier functions on them; like generic 2-pickup electric guitars have 2 volume, 2 tone knobs and a 3-stage switch, even acoustic pianos have 3 pedals on them to control the volume and sustain. So, I have concluded that my instrument can have multiple parts, built in parameter controls and either be percussive and/or trigger kind.

I decided that I can place more than one set of EMG sensors overall to my body to trigger melodic and percussive elements through a software or use them to progress through a song and several mechanical sensors like accelerometer, stretch sensor etc. to control the parameters of the effects and modifications that the will be processed through.

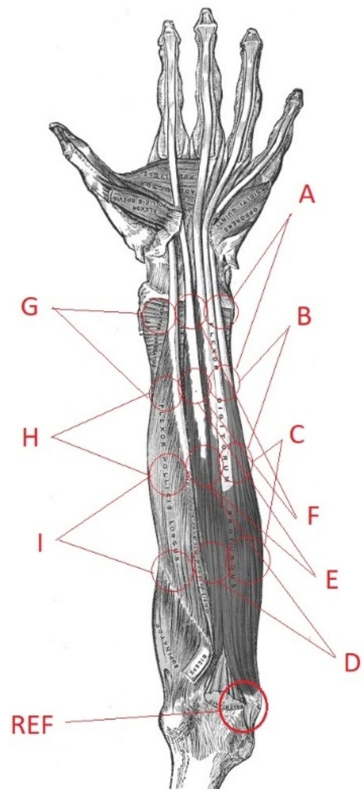

Figure 1: Muscles in forearm

I experimented the best hand/arm locations to collect EMG data since the signal amplitude level is very low and just a pulse of noise. The muscle which allows us to flex our index, middle, ring and pinky fingers is called "flexor digitorum profundus" which is the muscle located on the forearm's close side to our body. There are two main nerve groups that play major role in the movement of the fingers which are **median** and **ulnar** nerves. They are located side by side along the arm and they supply the flexor muscles, even though the workforce is distributed among these two, they act together for the movement of each finger. There are more than 20 muscles in the arm, but my main focus will be around forearm and mostly flexor digitorum profundus.

By placing the electrodes as pairs (an additional reference electrode placed around elbow which is same for all sets) around the forearm with 9 different locations and for 4 finger contractions (thumb not included), I obtained 36 different data sets (which is accessible in the appendix). After evaluating the test results, I concluded that the best position on the forearm is "C" and the most visible difference between the relaxed and tightened position can be obtained by moving the ring finger.

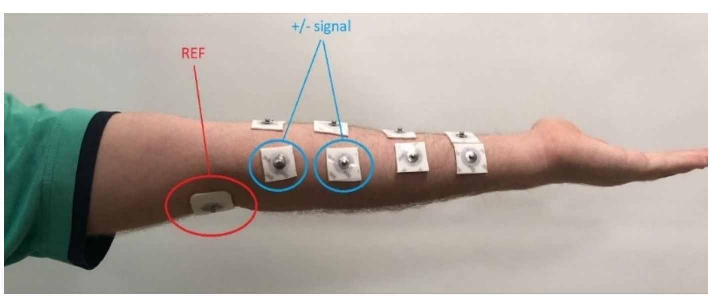

Figure 2: Optimum electrode placement on arm

The placement of the electrodes can be seen on the picture to the left as I placed 13 electrodes to my left arm to do this experiment. Like I said above, the dataset is available in csv format with a sample rate around 10kHz for a detailed examination.

When plotted the data for the location “C” with the ring finger, which I moved my finger 8-9 times in a 10 second time range, I observed the voltage spikes when the muscle is contracted, with an amplitude around approximately 1.4 to 1.6 volts.

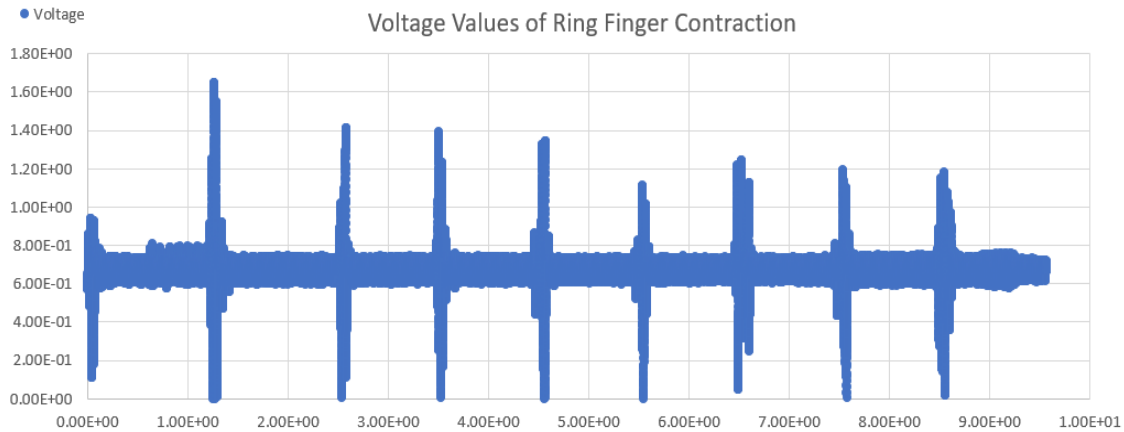

Figure 3: Left ring finger voltage spikes

nclude the thumb because there are three thumb-specific muscles which are responsible for its contraction and the neural activity is more visible in them when moving the thumb rather than forearm muscles. There are 9 muscles around and three of them are the main muscles which approximately start at the edge of the wrist and end at the base of the thumb are called **abductor pollicis brevis**, **opponens pollicis** and **flexor pollicis brevis**. The largest is called **abductor pollicis brevis**, it can be observed as the one that cover some of the part in our palm. It also produces the best EMG signal response for thumb movement.

In following part, I will try to explain the basics of my setup which you can also observe from the diagram.

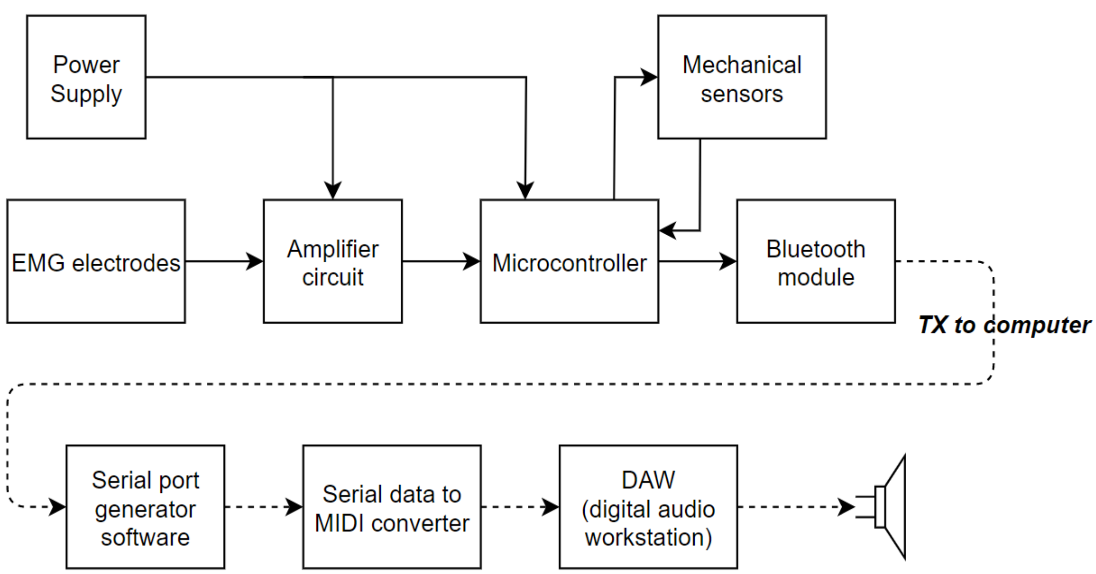

Figure 4: The basic setup of the instrument

There are two main parts to my setup. The blocks that are on top of the dashed line are the **hardware section** and the three block that are below the dashed line are **software section**.

HARDWARE SECTION

<u>Power Supply:</u> The amplifier circuit needs to be supplied by + - 9 volts. For the mobility purpose, I will use two 9-volt batteries.

<u>EMG Electrodes:</u> I have been using foam electrodes with gels but in the future, I want to use reusable gel.

<u>Amplifier Circuit:</u> This is a low noise high gain amplifier. Since the muscle potential signal has a range of 5-10 mv, and dominant frequency range between 50-150 Hz, there are a lot of noise between these ranges, especially around 50-60 Hz. The gain of the circuit will be around 2000.

<u>Microcontroller:</u> I am currently using an Arduino uno but for the mobility purposes I am planning to switch to micro but because of the lack of communication outputs I may use Arduino nano.

<u>Mechanical Sensors:</u> In addition to EMG sensors, I am planning to use accelerometers in order to measure the gravitational forces in two axes, x-axis being pitch, y-axis being roll. This will give me the ability to control continuous (at least 2 different) parameters per sensor with the small movements of my arm.

<u>Bluetooth Module:</u> I don't have any experience at RF communication, so I decided to use a HC-05 module which has a serial port protocol. I'm using a small Arduino shell for the module.

SOFTWARE SECTION

<u>Serial Port Generation Software:</u> The data which is received from the Bluetooth module of the instrument is serial which means bit-by-bit. So, in order for my computer to recognize these, I am using a software called **Hairless MIDI**. This software helps me to transfer serial data to MIDI ports, which is explained further in the next component.

<u>Serial Data to MIDI Converter:</u> MIDI, by definition, is a technical standard that describes a communications protocol that connects a wide variety of electronic musical instruments, computers and audio devices for playing, editing and recording music. It contains every musical aspect such as pitch, volume, velocity and other utilities. For this purpose, I am using a software called **Loop MIDI**. Its purpose is to create a MIDI port without any information in it. The **Hairless MIDI** software just creates a loop between the incoming serial information and the created midi port. This is important because any digital audio workstation is **mostly and only** compatible with MIDI protocol for any digital instrument, controller or any external audio device.

DAW (digital audio workstation: This is the end of the setup, before hearing the sound from a speaker. The software that I use for this purpose is called **Ableton Live 10 Suite**. It is a user-friendly environment for recording, creating and/or modifying any sound and it enables me to map the MIDI data that I obtain from previous steps to any parameter of any digital device, plugin etc.

To make it more clear and make it easy to understand, here is what I am planning to do with this device:

I will place **4 sets of electrodes** on both of my arms. Two of them will be on my forearms and other two will be on my thumbs. The purposes that I can assign for these can be several things. For example, you can assign different percussive elements to these and trigger them with movements. But this doesn't mean that you can repeatedly and only play 4 sounds. There will be a threshold set to understand that you contracted a muscle and want to play that note BUT the continuous data will also be used for measuring the desired velocity.

I want to draw your attention to the difference between VELOCITY of a sound element and VOLUME. Increasing both of them will surely increase the dB output but the thing about the velocity is that it corresponds to an organic aspect of the sound. In real life, you can hit a drum soft and hard. The hard hit will sound louder than soft but besides the volume, **all the transients will change**. That is the way that our ears can **distinguish digital sound from real life.** By utilizing this and certain 3rd party plugins, I will achieve an organic sound closer to the real life, depending on the time AND the strength of the contractions of different muscles. It is possible to apply this to melodic features as well as percussive sounds.

Besides the EMG sensors, mechanical sensors can be used as continuous parameters for both generating the sound and modifying it but I have experienced that it is more practical and easier to assign sensors like accelerometers to mainly effects parameters such as filter cutoff frequencies, wet/dry mix percentage, pitch/sample sweeping etc. For example while playing different sounds through several muscles with EMG sensors attached to them, you can simultaneously tilt one of your hand to left or right to increase the effect of a certain sound FX and the other hand slowly changing the frequency of a low pass filter applied to the overall mix. The reason and advantage for this is that you can control the desired parameters, without interrupting the instrument that you are playing. And it also contributes to the before-mentioned **organic sound** matter as no sound in real life can be produced exactly same as before.

THE DESIGN

The basic design will be like an "evening glove". The mechanical sensors will be placed on the hand and EMG sensors will be placed inside of the palm/forearm whereas the microcontroller and power supply will be on the opposite side of the forearm since that part doesn't move at all when you contract your fingers, when you tilt your hand or when you twist your hand from your wrist. It would also be an unnecessary weight to your hand. The locations of the components are shown below. They will be mounted/sewn into a lightweight flexible fabric.

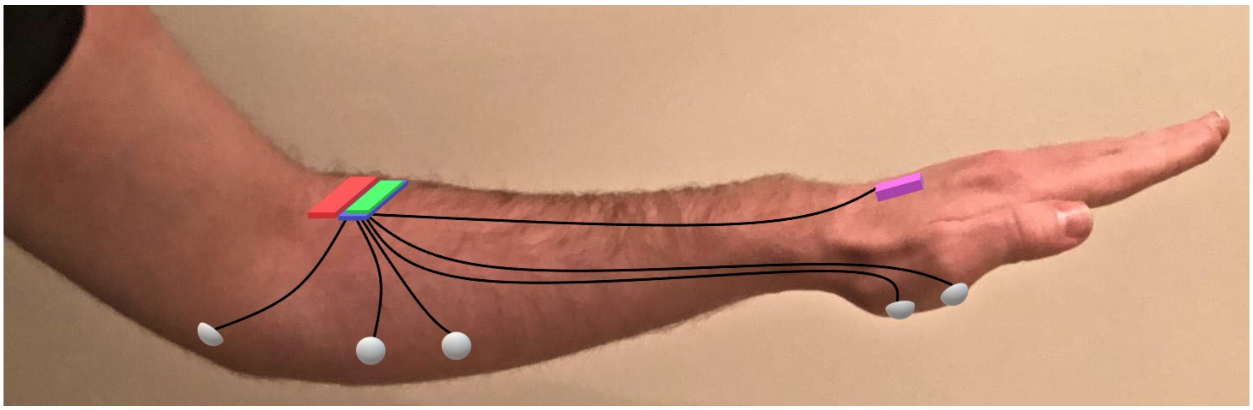
Figure 5: Placement of the components

<u>COLOR CODING</u>
Red: Power supply
Green: Microcontroller/modules
Blue: Amplifier circuit
White: Electrodes
Purple: Mechanical sensors

In the future, I can imagine that this will be part of my setup which consist of many different instruments, digital controllers and audio devices. I would love to use this instrument simultaneously with my other equipment/instruments.

THE CIRCUIT

I have been using a predesigned circuit for my applications and experiments, but I have designed one for myself with LTspice, in order to use it if the project continues beyond this point. The design is inspired by an amateur prosthetics maker named Nitya Dhanushkodi.

It takes around 5-15mV input and converts it around 1.5 volts. I will use **TL072** as the input amplifiers because it is a low noise JFET op-amp and **INA106** for the rectification and filtering stages as it is a differential amplifier which can accurately maintain gain.

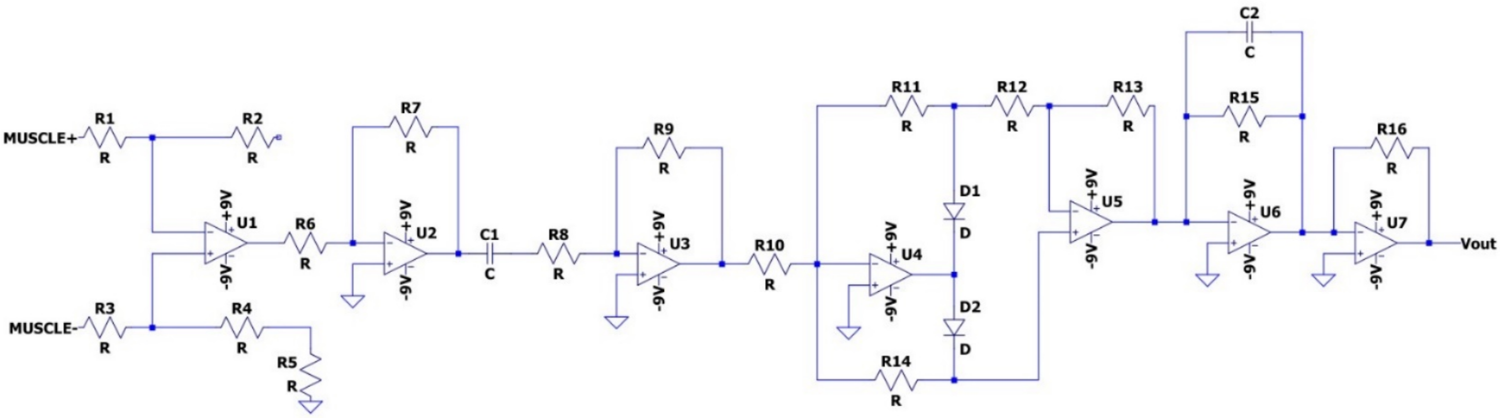

Figure 6: The amplifier circuit, designed with LTspice XVII

4. RESULTS & DISCUSSION

Initial objective was to create a piano like instrument which the fingers would be different keys on the piano. But differentiation of the EMG signals which are generated via the movements of different body parts that are attached to the **same or close long muscle group**, requires an advanced solution to implement, which I cannot manage to accomplish. Therefore, I came up with the idea of a hybrid sensor instrument, which is still playable as an instrument, practical and does not require a machine learning or data analyzing type of solution.

I have encountered some problems which I have not thought that I would face when I started this project. Here are the problems in a couple of short items.

- <u>Time:</u> The topic of my project has turned out to be too comprehensive. I love multidisciplinary work in any kind but the limited time I have until the deadline and all the other projects that I am involved has made it hard to reach to the final product for this project.
- <u>Competence</u>: Like I mentioned above, multidisciplinary work is needed to conduct this project. At some points during this project, I have either spent a lot of time learning a single topic or found out that I do not have enough knowledge with a topic that I thought I already knew.
- <u>Lack of equipment and components:</u> I have progressed easily and fast with the trial and error method at the beginning of the project but because of the inaccessibility of certain electronic equipment, the cost of certain components, the scarcity of the parts that I already have and the current disease outbreak which is keeping me away from the lab combined, is making it really hard to progress.

- Team: I learned that making a project by yourself can make some process faster but dividing the workload for a project to team members is really important, since it makes the progression stable and reduces the pressure on members.

I do not think that I have successfully completed this project, but I have created a playable new instrument and I contributed to this unique area. I couldn't get past my prototype design, but I have successfully managed to control a digital audio software with movement of my fingers and played a couple of simple notes, changed certain parameters. I say this is a success for me and in the future, I will certainly work and create in this area.

5. IMPACT

I think that this project has a potential of being a base-DIY product. You have the ability to tweak and adjust the instrument as you want. I have installed "MAP" buttons on the prototype for users to assign different body movements to different notes/parameters on the digital audio workstation, "BYPASS" buttons to arm and disarm certain functions as you continue to perform and so on. There will not be any Freedom-to-Use (FTU) issues since the ideology of this project is to art, contributions and alterations are more than welcomed.

6. ETHICAL ISSUES

In order to create my prototype, I have used several software and plugins, designs etc. Even though most of them were free to use and had a public use license, commercial distribution of these "as a package" would not be possible. But finding alternatives is possible for these or even creating alternatives is an option, it was just not practical in my case.

The idea of making a wearable EMG instrument is only possible with wearable and reusable EMG electrodes. Generally, the top companies like 3M, manufacture Ag/AgCl electrodes surrounded by a conductive gel and these are single-use items. This may be logical due to hygiene purposes but the consumption of these are excessive. The said top companies will not choose to develop a re-usable electrode since they can easily monopolize this nonstop demand of traditional electrode. I would have used a re-usable and wearable electrode, provided by my supervisor if the current conditions would have let me continue my project.

7. PROJECT MANAGEMENT

I have realized that production process is much shorter than research and development. I think that the best way to conduct these kind of projects is produce/buy a template device, modify it easily as you continue gain new knowledge and when you know what to do, how to do it and which components to use, start from scratch and build the device.

8. CONCLUSION AND FUTURE WORK

I have pointed out the problems I encountered and the topics that I was not adequate yet. But I will continue this project since this is what I would like to do for a living in the future. Now that I know the theory behind this project better, I would start by constructing hardware first, for my new prototype, and then the coding would be after that. Focusing on the software end has enabled me to get results quickly but every change that is made on the hardware, made it really time consuming and complicated to alter the software and coding.

9. APPENDIX

The measurement data in CSV format:

https://drive.google.com/drive/folders/16ffRopGgEhgKgZGRMQIRO3blHpX57Jfk?usp=sharing

BASIC MIDI CONTROLLER ARDUINO CODING

```
#include <MIDI.h>

int currentVal0 = 0;
int prewVal0 = currentVal0;

int currentVal2 = 0;
int prewVal2 = currentVal2;

MIDI_CREATE_DEFAULT_INSTANCE();

void setup() {
  MIDI.begin(MIDI_CHANNEL_OMNI);
  Serial.begin(9600);
}

void loop() {
  int sensorValue0 = analogRead(A0);
  int val0 = map(sensorValue0, 0, 1023, 0, 127);

  currentVal0 = val0;

  if (currentVal0 != prewVal0)
  {
    sensorValue0 = analogRead(A0);
    val0 = map(sensorValue0, 0, 1023, 0, 127);
    currentVal0 = val0;
```

```
    MIDI.sendControlChange(1, val0, 1);
    delay(20);
  }
  prewVal0 = currentVal0;


  int sensorValue2 = analogRead(A2);
  int val2 = map(sensorValue2, 0, 1023, 0, 127);

  currentVal2 = val2;

  if (currentVal2 != prewVal2)
  {
    sensorValue2 = analogRead(A2);
    val2 = map(sensorValue2, 0, 1023, 0, 127);
    currentVal2 = val2;
    MIDI.sendControlChange(2, val2, 1);
    delay(20);
  }
  prewVal2 = currentVal2;

  //Serial.println(currentVal2);
  delay(20);
}
```

ACCELEROMETER, BLUETOOTH AND EMG COMBINED ARDUINO CODING

```
#include <MIDI.h>
MIDI_CREATE_DEFAULT_INSTANCE();

#include <Wire.h>  // Wire library - used for I2C communication
int ADXL345 = 0x53; // The ADXL345 sensor I2C address
float X_out, Y_out, Z_out;  // Outputs

float mapfloat(float val, float in_min, float in_max, float out_min,
float out_max) {
    return (val - in_min) * (out_max - out_min) / (in_max - in_min) +
out_min;
}
int temp0;
int temp1;

void setup() {
  MIDI.begin(MIDI_CHANNEL_OMNI);
  Serial.begin(9600);
  Wire.begin(); // Initiate the Wire library
  // Set ADXL345 in measuring mode
   Wire.beginTransmission(ADXL345);  //  Start  communicating  with  the
device
```

```
  Wire.write(0x2D); // Access/ talk to POWER_CTL Register - 0x2D
  // Enable measurement
   Wire.write(8);  //  (8dec  ->  0000  1000  binary)  Bit  D3  High  for
measuring enable
  Wire.endTransmission();
  delay(10);
  // This code goes in the SETUP section
  // Off-set Calibration
  //X-axis
  Wire.beginTransmission(ADXL345);
  Wire.write(0x1E);  // X-axis offset register
  Wire.write(1);
  Wire.endTransmission();
  delay(10);

  //Y-axis
  Wire.beginTransmission(ADXL345);
  Wire.write(0x1F); // Y-axis offset register
  Wire.write(-2);
  Wire.endTransmission();
  delay(10);

  //Z-axis
  Wire.beginTransmission(ADXL345);
  Wire.write(0x20); // Z-axis offset register
  Wire.write(3);
  Wire.endTransmission();
  delay(10);
  pinMode(2, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
}
void loop() {
  // === Read acceleromter data === //
  Wire.beginTransmission(ADXL345);
  Wire.write(0x32); // Start with register 0x32 (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(ADXL345, 6, true); // Read 6 registers total, each
axis value is stored in 2 registers
  X_out = ( Wire.read()| Wire.read() << 8); // X-axis value
  X_out = (X_out/256)+0.95; //For a range of +-2g, we need to divide
the raw values by 256, according to the datasheet
  Y_out = ( Wire.read()| Wire.read() << 8); // Y-axis value
  Y_out = (Y_out/256)+1;
  Z_out = ( Wire.read()| Wire.read() << 8); // Z-axis value
  Z_out = Z_out/256;

  if (Y_out < 0){
    Y_out = 0;
  }
```

```
if (X_out < 0){
  X_out = 0;
}
digitalWrite(2, LOW);
digitalWrite(7, LOW);
digitalWrite(8, HIGH);

int val2 = analogRead(A2);
int val3 = analogRead(A3);
int val4 = analogRead(A1);
bool check2 = false;
bool check3 = false;
bool check4 = false;

if (val2 > 1000){
  check2 = true;
  delay(1000);
}

if (val3 > 1000){
  check3 = true;
  delay(1000);
}

if (val4 > 1000){
  check4 = true;
  delay(1000);
}
while(check2 == true){
  digitalWrite(2, LOW);
  digitalWrite(7, HIGH);
  digitalWrite(8, LOW);
  int val4 = analogRead(A1);
  MIDI.sendControlChange(1, 50, 1);
  delay(500);
  MIDI.sendControlChange(1, 51, 1);
  delay(500);
  if (val4 > 1000){
    check2 = false;
    digitalWrite(2, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    }
}
while(check3 == true){
  digitalWrite(2, LOW);
  digitalWrite(7, HIGH);
  digitalWrite(8, LOW);
  int val4 = analogRead(A1);
  MIDI.sendControlChange(2, 50, 1);
  delay(500);
```

```
  MIDI.sendControlChange(2, 51, 1);
  delay(500);
  if (val4 > 1000){
    check3 = false;
    digitalWrite(2, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    }
}
int val0 = mapfloat(X_out, 0, 2, 0, 127);
if (val0 > 127){
  val0 = 127;
}
if (val0 != temp0+1 && val0 != temp0-1 && val0 != temp0){
  MIDI.sendControlChange(1, val0, 1);
  }
delay(20);

temp0 = val0;

int val1 = mapfloat(Y_out, 0, 2, 0, 127);
if (val1 > 127){
  val1 = 127;
}
if (val1 != temp1+1 && val1 != temp1-1 && val1 != temp1){
  MIDI.sendControlChange(2, val1, 1);
  }
delay(20);

temp1 = val1;
bool startStop = false;
int val5 =analogRead(A5);
if (val5 > 1015){
  digitalWrite(2, HIGH);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  delay(1500);
  startStop = true;
  while (startStop == true){
    delay(1500);
    val5 =analogRead(A5);
    if (val5 > 1000){
      startStop = false;
      delay(1500);
    }
  }
}
/*Serial.println(val2);*/

/*Serial.print("Xa= ");
Serial.print(val0);
```

```
  Serial.print("   Ya= ");
  Serial.print(val1);
  Serial.print("   Za= ");
  Serial.println(Z_out);*/
}
```

10. REFERENCES

- DIY Muscle Sensor / EMG Circuit for a Microcontroller. (2017, November 1). Retrieved from https://www.instructables.com/id/Muscle-EMG-Sensor-for-a-Microcontroller/
- Farina, D., & Merletti, R. (2000, September 28). Comparison of algorithms for estimation of EMG variables during voluntary isometric contractions. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S1050641100000250
- Flexor pollicis brevis muscle. (2019, September 20). Retrieved from https://en.wikipedia.org/wiki/Flexor_pollicis_brevis_muscle
- Forearm. (2020, March 9). Retrieved from https://en.wikipedia.org/wiki/Forearm
- Serial communication. (2020, March 15). Retrieved from https://en.wikipedia.org/wiki/Serial_communication
- A Portable MIDI Controller Using EMG-Based Individual Finger Motion Classification. (n.d.). Retrieved from https://ieeexplore.ieee.org/document/4463328.
- Abdullah, Rahim, A., Saad, M., Weihown, & Jingwei. (2019, February 22). EMG Feature Selection and Classification Using a Pbest-Guide Binary Particle Swarm Optimization. Retrieved from https://www.mdpi.com/2079-3197/7/1/12/html.
- Neuromuscular system. (n.d.). Retrieved from https://www.healthdirect.gov.au/neuromuscular-system.
- Mills, K. R. (2005, June 1). The basics of electromyography. Retrieved October 20, 2019, from https://jnnp.bmj.com/content/76/suppl_2/ii32.

- projects, C. to W. (2019, September 27). Electromyography. Retrieved from https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvRWxlY3Ryb215b2dyYXBoeQ.
- Raez, M. B. I., Hussain, M. S., & Mohd-Yasin, F. (2006). Techniques of EMG signal analysis: detection, processing, classification and applications. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1455479/.
- Smith, L. H., & Hargrove, L. J. (2013). Comparison of surface and intramuscular EMG pattern recognition for simultaneous wrist/hand motion classification. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3914139/
- A. Hiraiwa; K. Shimohara; Y. Tokunaga (1989) EMG pattern analysis and classification by neural network Retrieved from
- https://ieeexplore.ieee.org/abstract/document/71472.
- Gokgoz, E., & Subasi, A. (2015, January 24). Comparison of decision tree algorithms for EMG signal classification using DWT. Retrieved from https://www.sciencedirect.com/science/article/pii/S1746809414002006.
- Reaz1, M. B. I., Hussain1, M. S., & Mohd-Yasin1, F. (2006, December 1). Techniques of EMG signal analysis: detection, processing, classification and applications. Retrieved from https://biologicalproceduresonline.biomedcentral.com/articles/10.1251/bpo115.