Final Project - Slug Symphony

Jerret Schlosser, Eric Vetha, Clarissa Farfan,

Jose Mario Gutierez, Karla Vazquez

ECE 167

Git Commit ID: 06a52023044c15540e4b15ea799f3b699629fb25

Introduction

The goal for Slug Symphony is to be able to create a pair of gloves that can be used to create an air-rock band. We loved the idea of being able to use sensors and create pseudo instruments during the labs, so we wanted to see just how far we could push that idea. We all knew what the game guitar hero was when growing up, so we naturally drifted towards wanting to recreate that in some way. We decided to create a pair of gloves that could act like multiple different instruments depending on how they are used. The main goal was to create 5 different instruments that these gloves could recreate. We wanted to be able to play saxophone, guitar, drums, piano, and trumpet. Each instrument would have its own specific hand placements, motions, finger movements, and most importantly actual notes being played. The idea is that a user could put on the gloves, move their hands to the correct instrument position, and then just start playing. The way you would play would be like playing an imaginary version of the instrument, and then whenever the user is ready they could simply swap hand positions and play a new instrument.

Background

The background for the project is that when using the 9-DOF we could measure angle and movement, so the idea of an air guitar was the first thing we thought of. The 9-DOF would be able to measure the strumming hand, and then we realized the flex sensors could be used to recreate the other hand and make the actual notes. We then realized that you can do this same concept for other instruments, so we wanted to push this concept and see how many instruments we could recreate within the time frame.

The mechanical background would be that since flex sensors can give a wide range of values we could simply use them as rudimentary buttons so that if it passes a specific resistivity value, a flag would be raised. Using the method described above, you could get 3 individual readings of finger 1 pressed, finger 2 pressed and finger 1 and 2 pressed in unison. For the 9-DOF, we would be able to use multiple parts within it to help out our design. The accelerometer could tell us which orientation relative to gravity the hands are in to determine the position and which instrument to play. Refer to the chart below for accelerometer reading. Additionally, the gyro was able to tell us the angle and movement. We were able to tell if the 9-DOF was moving as well as what angle it was currently at. Based on what instrument you were configured in, movement of your hands and movement of your fingers would modify and output tone.

We chose these two options because the 9-DOF was the best option to capture movement, angles, and everything. We chose the flex sensors because if we were to use buttons or capacitive touch sensors or anything else, it wouldn't look like the real life of the instrument. We wanted it to be similar to the real-life instrument and the flex sensor was the only method that would work because it doesn't need to be touched at the right strength or angle or anything, it just needs to be there.

Reading from Accelerometer	Left-Hand Reading	Right-Hand Reading
Reset	-Z	-Z
Piano	+Z	+Z
Trumpet	+X	+X
Saxophone	+Y	-Y
Drums	+Y*	-Y*
Guitar	+X	+Y

*with flexing of fingers

Implementation

Implementation. What parts did you use, and how were they connected and programmed? Did you use any class libraries? Did you create your own?

The major parts that we needed for this project were gloves, flex sensors, 9dofs, the uno32, and then assorted wires and resistors. We needed to combine all of these components into one glove and program them together. We used a combination of the libraries we made in class and were provided, as well as created a variety of new ones that could be better used for us. Additionally, we made a state machine from scratch that could be used to run the entire program.

Part 1: Flex Sensors

The first thing we decided to tackle was the flex sensors. The main goal we needed was to have as many flex sensors as possible that could all be distinguishable from each other. When connecting the flex sensors to the glove, we wanted to be able to use only one uC32 board. Given that there are only six analog output ports, we realized we either had to cut down the number of flex sensors or pair them up. When pairing the six sensors up, we connected each pair in parallel. Since two flex sensors were being read to one port, we needed to be able to differentiate between each reading. This was achieved by adding a 460Ω resistor to one flex sensor in series to create a noticeable offset which created four separate values when flexing the sensors individually, together, or not at all. Two 47kΩ resistors were grounded to complete the circuit.

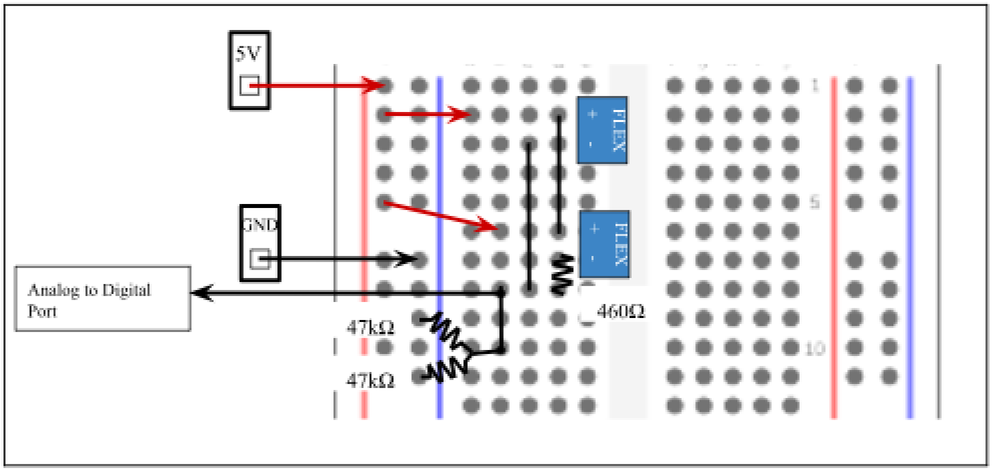

Figure 1. Flex Sensor Schematic.

The flex sensors as well as the power and ground rails were connected to a perf board shown below. The output was then connected to the uC32 board and MPLAB X IDE was used to program the board and read each sensor output via analog output ports. The figure below shows the flex sensor connections for an entire hand.

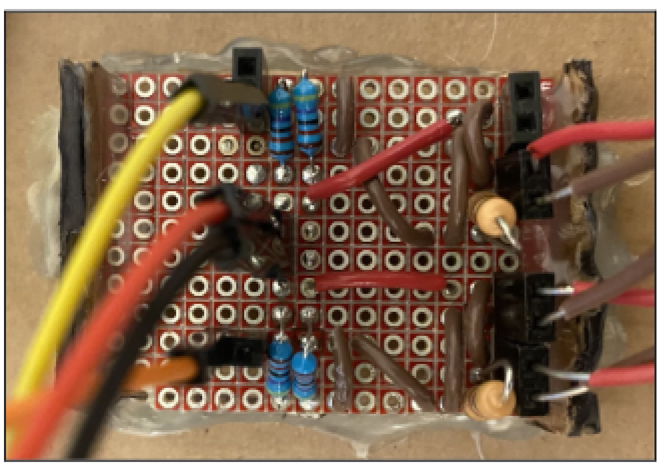

Figure 2. Flex Sensor Circuit on PerfBoard.

Part 2: 9-DOF

To differentiate between instruments, we utilized the readings from the 9dofs. We needed to be able to detect which direction the hands were being held. The main method we decided would work for this is by using the accelerometer to determine which hand position we were in. The reason we did it this way was because the accelerometer can detect gravity and when it is flat on a table it makes it very easy to get accurate readings that would consistently read correct no matter how we started or anything such as that. In order to achieve this we decided to mount the 9dof on the palm of the hand. This made it so that it could get accurate readings of the hand.

Part 3: Glove Construction

Figure 3. Gloves.

In order to construct the gloves we initially assumed that we would just attach the flex sensors to the outside fingers; however, once starting this process we found this was incorrect. One of the main things we found out the hard way was if the flex sensors were attached to the glove in an inconsistent way where it wasn't a clean bend, the sensor would give inaccurate readings and sometimes when we would test them several times in an incorrect configuration they would break. After several design iterations, we concluded the best way to do this was to input the flex sensors on the inside of the gloves rather than the outside. This improved our design because it allowed the user to have a firmer grip on the flex sensor and thus provide us with accurate and consistent readings. We also cut the fingers off the tips of the gloves to make the design more breathable and allow the flex sensors to not be as restricted as they were.

Part 4: State Machines

With a fully built set of gloves, we shifted towards programming the gloves and making them functional. The main program we needed to make this run was a main state machine. This consisted of having a main state where it could figure out what instrument position we were in. Once the gloves were moved to a new position it would transition into the individual instrument states.

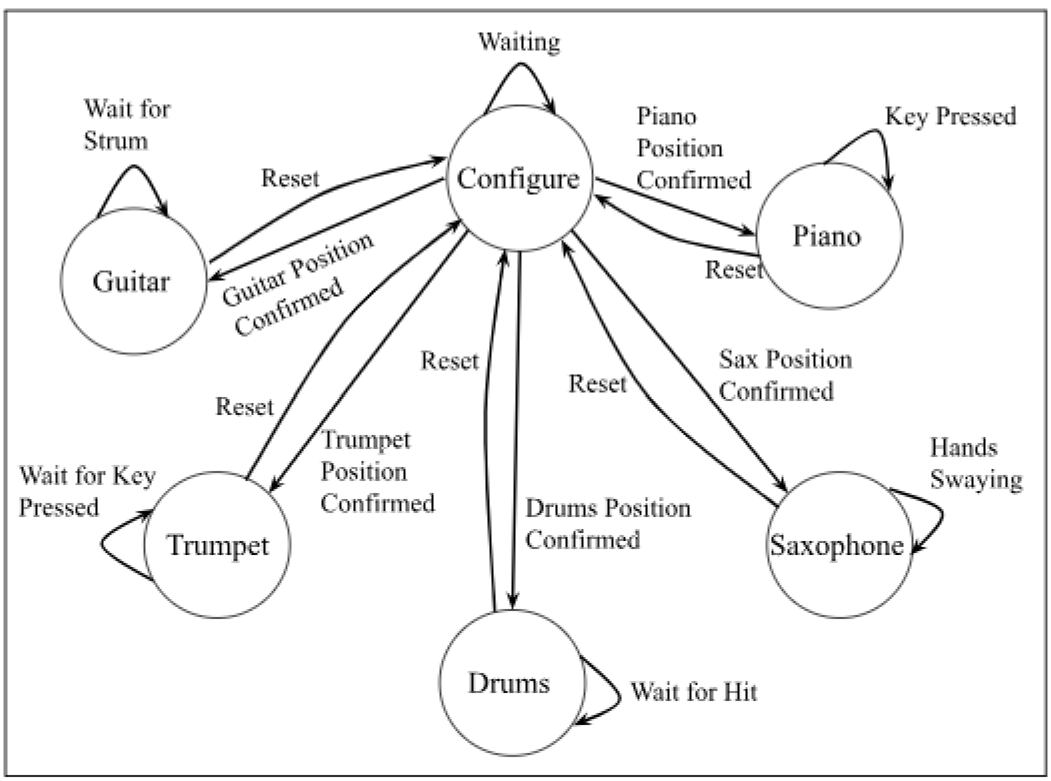

Figure 4. Instrument State Machine.

After figuring out this larger state machine, we then needed to have individual smaller state machines that could be run in order to make each instrument play music and react appropriately. Below are the different state machines for each instrument.

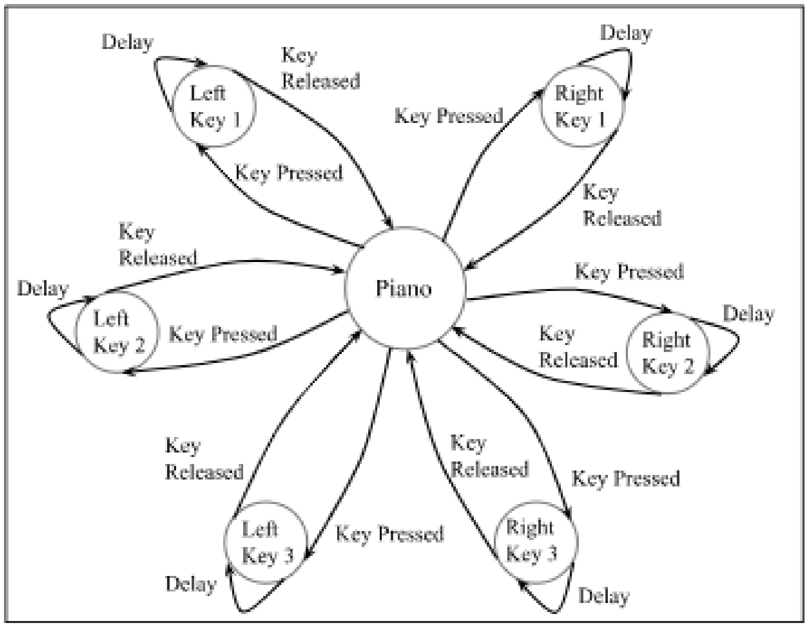

Figure 5. Piano State Machine.

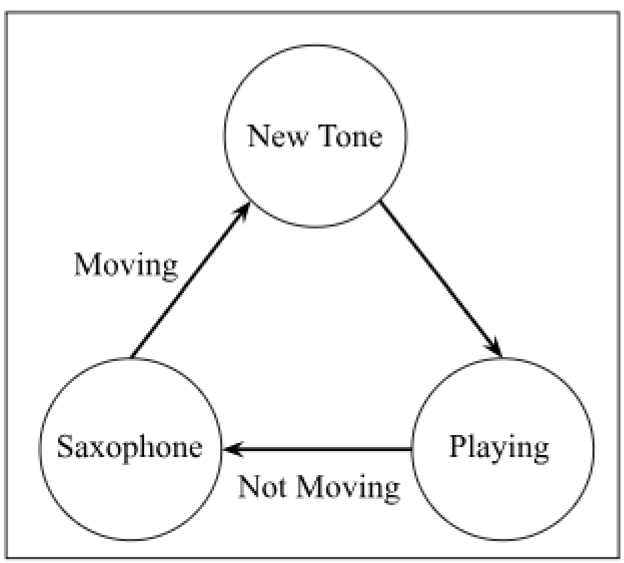

Figure 6. Saxophone State Machine.

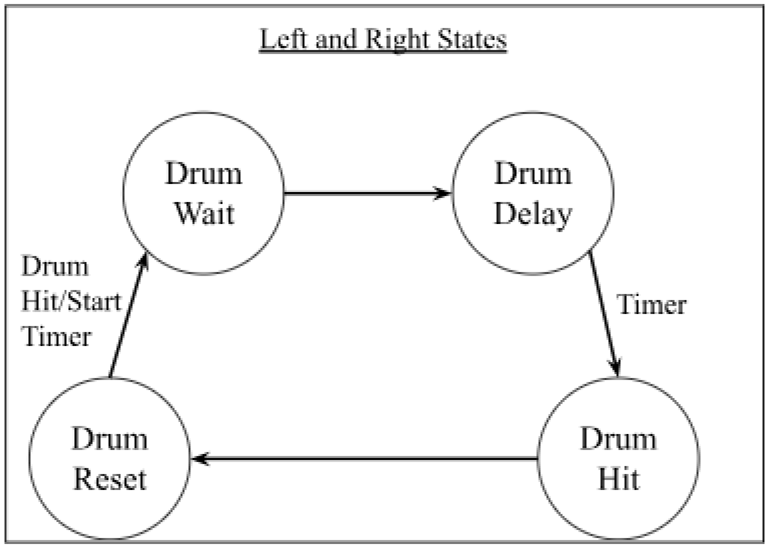

Figure 7. Drum State Machine.

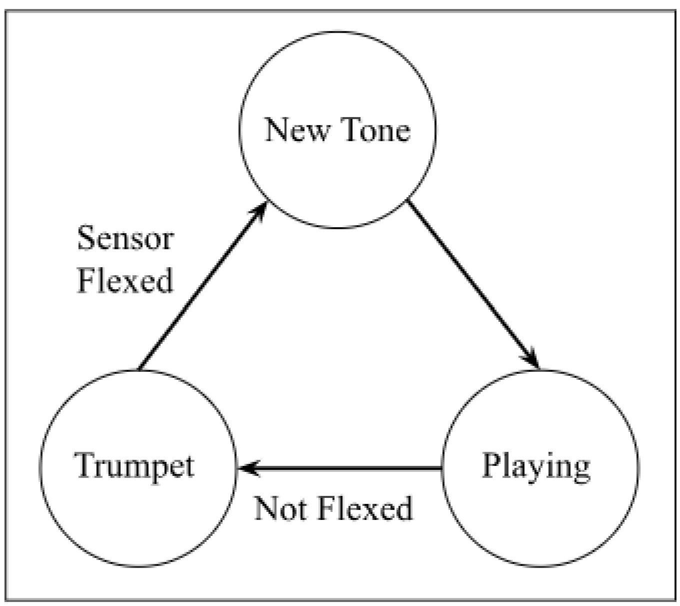

Figure 8. Trumpet State Machine.

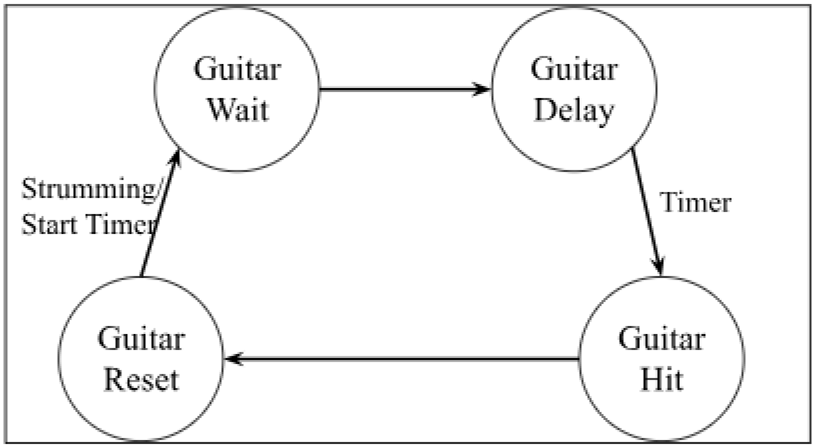

Figure 9. Guitar State Machine.

Part 5: Instruments

After all the state machines were complete, we now needed to make sure the instruments fully worked correctly. In order to do this we needed to create a way of producing the music we needed. We did this by creating a python program that could create a ui that would take print statements from the terminal and play the associated tone. This meant getting individual mp3 files for every single instrument and their individual notes. With all of the files, we then simply made them correspond to all of the different movements and instruments.

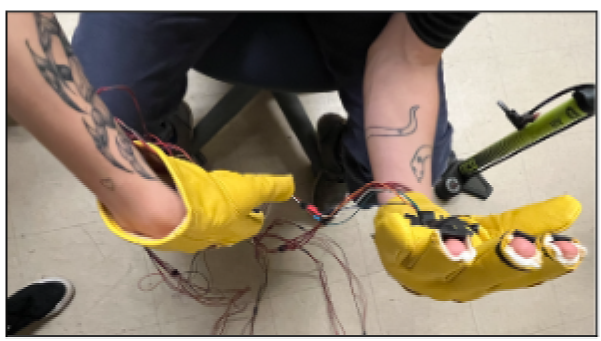

Figure 10. Guitar Hand Position.

- Left palm pointing upwards; Right hand emulating how one would hold a pick, with the palm facing toward the user.
- Finger tapping on the flex sensors on the left-hand modifier the tone, simulating chord changes.

- Motion on the right hand triggers the sound, simulating strumming.

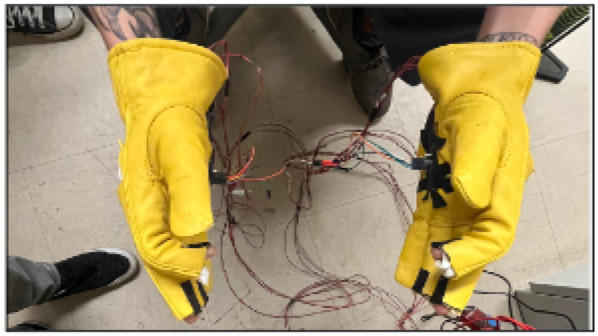

Figure 11. Drum Hand Position.

- Both hands pointed out in front of user with index fingers curled in.
- The speed of movement will affect the rhythm and intensity of the tone.

Figure 12. Saxophone Hand Position.

- Hands straight out in front of user
- Tapping and moving fingers controls note changes.
- To emit a tone, the user will rhythmically move their hands as if they were dramatically performing a solo. (Moving their hands side to side in unison).

Figure 13. Piano Hand Position.

- Both hands with palms facing down.
- Finger tapping on the flex sensors produces individual tones.
- Movement of your hand in the x-plane will modify the pitch.

Figure 14. Trumpet Hand Position.

- One hand in front of the other in the x-plane.
- Tapping and moving fingers controls tone changes.

Combining all these elements, we were able to program the gloves and achieve the desired musical outputs. From the five figures above, each instrument can be played and seamlessly transition from one instrument to the other. While the program had no issues transitioning to the desired instrument, it would sometimes take longer than five seconds to transition.

Evaluation

In order to evaluate the success of this project, we quantitatively check if our gloves passed certain requirements and worked at a 100% rate. We needed 5 instruments, each with a full set of tones corresponding to different finger flexing; as well as, there needed to be little to no error whenever these instruments were in use. Additionally, we needed to be able to have seamless transitions between instruments. After coding all of the instruments, we knew we had complete sets of instruments with tones. We also had the ability to change instruments on the fly without any issue. We tested there was no error with this by repeatedly trying to change instruments and if it was able to change 10 times without fault then we knew it was good. We were able to achieve this so we were satisfied with the change. We also used this method to make sure that all of the tones played correctly. We would check each finger 10 times and at the same times as others and if it all played correctly we were good. After a very long amount of time and testing, we were able to get to the point where we were 100% accurate with the production of music. We also tested the gloves by giving them to strangers or friends and asking them to test them out and see if they worked for them. After doing this 2 times, we found both people were successful in understanding and working with the instruments. Overall, we found that we were extremely successful in making gloves that were accurate and responsive.

Conclusion

Our Group was capable of creating 5 unique instruments capable of emitting different tones and modifying them at will. We were also capable of meeting our stretch goal of being able to transition smoothly between instruments.

The two largest issues our group encountered were the durability of the flex sensors and their accuracy. The durability seemed to be an issue of how the flex sensors were fixed to the glove, as this was an issue primarily in the right glove and not the left. The sole difference being the right glove was fastened lower on the sensor causing occasional kinks and less than ideal bending of the sensor. Due to the shortage of flex sensors, this involved having to modify the range that would be considered pressing your finger. So fastening and limiting movement at the base appears to be the solution to this problem. Another issue that came up often was poor connections causing data lossage. We opted for having long wire connections so that the user could move their hands with ease however the SDA/SCL cables would come loose. This would require all pins of the 9dof to be disconnected and reconnected or it would be unusable.

Overall, we feel that our end goals were accomplished and we were able to combine the knowledge we learned in this course by utilizing the features provided to us by the flex and 9DOF sensors to be able to produce a working project that we all found to be interesting and useful. If we were to attempt this again I definitely think we would spend a lot more time refining our glove design as well as our code so that both would be a more cohesive and smoother working product, however given our time constraint as well as our limited supplies we feel as that we came up with a very innovative solution.