Final Report Group C14

Kiril Voigtländer *s1094153*
Kareem de Quillettes *s1108109*
Radboud University *24. January 2024*

---

1 User documentation

The product we created is a self playing guitar. It consists of a physical part you need to attach to a fitting guitar, and a website to control the device. When you have attached the product to the guitar correctly, and you have set up the website. You choose a song from the website or upload a new song, and the song will start playing on the website.
To use the product you first need to attach the part for the body of the guitar to the body; and the part for the neck to the neck. The part on the neck attaches with two screws at the bottom of the neck, and some simple connections of 3d printed parts. The part on the body of the guitar attaches with some screws. With the current iteration of the product, some tape is needed to fully secure the neck part of the product. To secure the product tightly on the neck some tape is used, and to stabilize the product so that the parts cannot move when the servo motors are moving.
When all is attached, you need to wire up the servos to the raspberry pi on the body. Now you need to supply power the raspberry pi and the servo controller hat on the raspberry and run start the server from the raspberry pi.
When the server is online, you can run the website. On there you can upload a midi file of a song you want to play into the database. You can also delete the songs that are in the database already. From the list of songs in the database you can select a song to play it on the guitar.

2 Software & electrical/mechanical design

Our product consists of a mechanical part and of software. Both of these can then again be split up; the mechanical part in picking and fretting; and the software in the website and playing functionality.

2.1 Mechanical part

We did not have the means to play all strings of the guitar. If we wanted to accomplish this we would have needed a lot more servos, plus a way to control that many of them. Another limitation we had was the lack of space. The strings are all very close to each other so if we needed to play them all we would have had a lot more issues with space. Because of these limitations we chose to focus on playing three strings, and nine notes per string. We now had to

choose which strings we wanted to play. There where a lot of things to consider while picking the strings. What are the notes that are used the most; would we rather play high notes or low notes; do we want strings far apart so we have more range, or close together so we can play notes that are close together more efficiently; and also very importantly, what is easier construction wise, the strings you do not use you can lean on when fretting the other strings. After good consideration we decided to use the second, third, and fourth strings of the guitar.

2.1.1 Picking the guitar

Picking a guitar refers to using something, normally a guitar pick or finger, to vibrate the strings producing a sound. Looking for a way how to do this automatically we quickly came to the idea of using servo motors. Finding out exactly how we could pick the guitar with these servos was a bit of a challenge however. Online we had found someone who had done a similar project as us. Olav Martin Kvern made a guitar that can pick itself [3]. The way he tackled this problem seemed like a very good solution, but it was hard for us to implement, because of the multiple complex 3d printed parts. So we got in contact with him, to ask him some questions about how he managed to do it and if he had any suggestions or tips for us. He was very helpful and offered some great suggestions for us. One thing he send us, was a link to a project where someone made a self playing lyre using servo motors [1]. They used the servo horns that come with the servos, attached them to the servos, and then turned the servos to pick the strings of the lyre. This way of picking the strings was less complicated to implement and we therefor chose to do it in this way.
So we use the servo horns and the servo motors to pick the strings. To do this we glued down the servos to a 3d printed frame in such a way that they align with the strings. The servo horns that attach to the servos have to be sanded down to have the right length and shape for picking a string without getting stuck.

2.1.2 Fretting the guitar

To fret a guitar is to push down a string on the neck in a place, to produce a corresponding note when the string gets picked. To fret the notes that we wanted to fret we needed a way to push down on the neck in the same spots, with enough force consistently. We chose to use servo motors for this because of a few reasons. Firstly, we already use servos to pick the strings and to control those we use a servo controller hat for the raspberry pi. This servo controller can control and power sixteen servos at a time, so by using servos to fret we do not need any additional hardware. Also with the use of servos, we can fret two notes on a string with one servo, increasing the amount of notes we can play.
We now need to figure out how to attach the servos to the guitar, and how to make the servos push down on the strings.

To attach the servos to the guitar we needed a frame that attaches to guitar securely and on which we can glue the servos. We chose to make a 3d printed frame that partly screws on to the guitar. We made the frame this way because it seemed like the easiest, and fastest way. This frame can be put on and removed from the guitar by loosening some screws and undoing some connections from the 3d print. This way it can be taken off and put on the guitar easily. Then we glued the servos on the frame and on top of each other, to align above the strings between notes on the neck.
Finding a good way to push down on the neck was quite difficult. We first tried using the servo horns that come with the servos to push down on the strings 1. What we would do was attaching multiple horns to each other with small screws, and the attaching the whole construction to the servos. The problem with this was that with this construction, there was too much play in the arms. This meant that it was hard to properly align them with the strings. It also made them unreliable, sometimes missing the strings or not pushing down hard enough on the strings.
We then thought a better way to fret the strings1. We could cut out some pieces of plastic to the exact shape and size to fret the strings. To these pieces of plastic we can then attach small servo horn to the bottom so we have more service area to catch the string. These pieces of plastic can then be screwed on a servo horn and that can be attached to the servos. This construction is less flexible and can not move once fully screwed down unless the servo move. This makes this the more reliable and overall better option so this is what we used.

2.2 Software

We ran all of our software from a Raspberry Pi 4. We chose this device because it had the needed power to run the code for controlling the servos, while also running the website. Some weaker micro controllers would not have been powerful enough to handle this. Another reason for using the Raspberry Pi 4 was the compatibility with Python. This proved very useful because it was easy to find Python libraries for all our needs. It was also very useful that we had the servo controller that plugged onto the Raspberry Pi.

2.2.1 Playing midi files

To play a song with our device we needed some code to drive the servos. We wanted our device to be able to use a midi file to play song. We wanted to have this functionality because the midi format is very versatile and there is a lot of support for it. This also meant that there was an easy to use Python library we could use to load midi files into a python program and translate the files to something we could use. The library we used for this is called Mido [2]. With this library we can decipher the information about each note and feed them into our own functions to fret a string and then pick it.

2.2.2 Website

We decided to create a website using the Django Framework, which we already worked with in the course R&D Project, while working in that course we saw the advantages of Django and hence decided to use it here too. On the website the user has the option to either add, delete, edit or play a file. All of those four actions are done on a separated web link as we thought that this will be nicer and easier to overview then if we would have listed them all out on one site. We could have also worked with a drop out menu instead of creating a web link for each one of these activities, but we didn't manage to implement it and on top we already have done a similar listing for the last project were we also used Django and hence we were already familiar with how it should be looking. With the adding one has the option to add any midi file they want to the database of the project.

3 Sustainability

Making our project sustainable was a big challenge we faced. There were quite some options which we looked into where the use of servos could potentially be reduced, but on the other hand the power consumption would be increased and on top of that most of these options included too expensive parts or were above our knowledge. Eventually we came up with a way on how to make the fretting of the strings sustainable. Instead of using a servo for every single option to fret we decided to instead use a servo and turn it to the left and to the right 3. Doing so we managed to half the amount of servos needed in the implementation of the project. Furthermore it also added a challenge to the project about which we will talk a bit more in section 2.1.2.
Before deciding on which servos we will use we tried searching up the torque needed to play a string and to fret a string. Our goal in doing so was that we will not use too powerful servos for a task, which can also be fulfilled by a weaker servo. This also had a positive effect on our power consumption and hence on sustainability.

4 Reflection on implementation

The implementation of the project went more or less good. First we wanted to come up with a way to pick the string in the front. The problems encountered doing that are talked about in the section 2.1.1. After we finished that up we started to think about the fretting of the strings as this was the order in our plan we made at the beginning of this course. We then came up with a 3d print, which was supposed to hold the servos for the fretting part, additionally we also ordered some more parts. While waiting for those parts to arrive and for the 3d print to finish we started working on the website, which we claimed to be easy with the voice control and hence left it for the last week in our plan. We decided to use Django as our designated website host and developer as we

already made some good experiences with it. While working on the website we encountered numerous bugs and problems such as that Django didn't want to start/couldn't be installed on the raspberry pi, that the website couldn't be reached, that the files weren't saved properly in the data base and many more. Eventually we ended up spending roughly three weeks working on the website, which wasn't as bad as it sounds as we were still waiting on the 3d print and on the ordered parts. After the Christmas break we then received the ordered parts and started on the fretting, which we planned to do in three weeks in our original planning as it was the most complicated part of our project. We then first had the idea of making the fretting parts out of the servo arms, but we quickly realized that this isn't going to work 2. Then on the Tuesday of the week were we needed to present we came up with a new idea on how to make the fretting parts 1. This leads to the next bug in our implementation. Here we measured the length, which was needed without considering that the whole construction will be moving when the servos are turned. Hence, we needed to stabilize the whole construction with tape, which led to the fact that the whole construction where the servos where on top got moved a bit to the back, which ended up causing the problem that most of our fretting arms were too short to reach the string now. In the future we could improve on this problem by creating a new construction for holding the servos, which is more stable and then after some more test runs and with good results glue the servos on and then start measuring the length needed for the arms. This could improve the problem, but the margin for errors is thin as if you make the arms to short the might never touch the string (problem that we have now at some arms) or if you make them to long they will "silence" the string.

5 Reflection on the learning process

The main thing, which we would do differently if we had the chance to redo this project again would be the model for the fretting of the strings. We would spend more time in trying to work out a more stable way of the construction, which wouldn't lead to us having to use a lot of tape or other external materials to stabilize the construction. We would also start working on the fretting arms the way we did it in the end as we are confident that with more time we would have managed to make them the right size and to get them working. In our procedure on how we would work on the project we would first work on the fretting before we start working on the website, but this was already the plan for this try too, but because of circumstance out of our reach we were not able to implement it that way. The main reason behind doing the fretting first would be that it is the hardest mechanical part of the project and it needs the most time and roughly one week is not enough time to make the construction stable and to figure out the right length of the fretting arms. So in conclusion we would say that working on the designated order would have lead to a better result probably if we would have considered the final way of constructing the fretting arms in the beginning. If we would do the project again we are confident that

we would be able to produce a better result as we would be able to work out the fretting of the strings completely, which is one of the most important part of our project. Besides that our main idea didn't change while working on the project we always followed the same goal to pick and to fret three strings on the guitar. We did change one of the possible extensions we had planned to do, which was the voice control, because we ran out of time for it and decided to get the other parts work fully before committing to another sub problem.

6 Images

Figure 1: The new and final way of the fretting arms

Figure 2: The first try of making the fretting arms

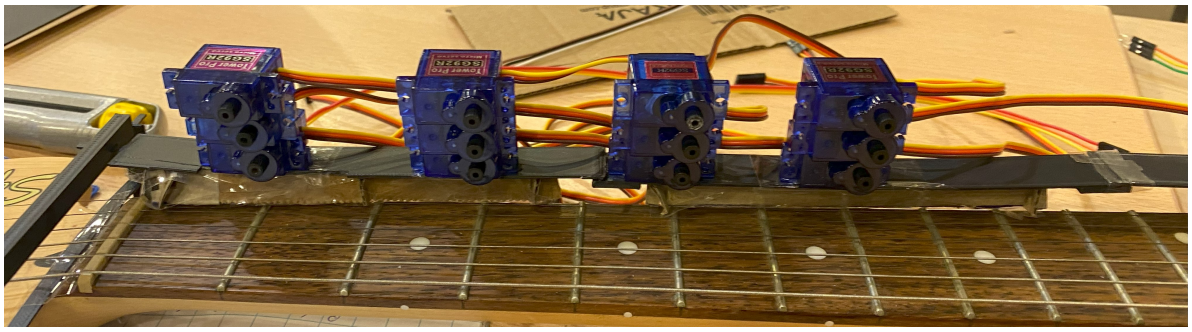

Figure 3: Placement of the servos for the fretting

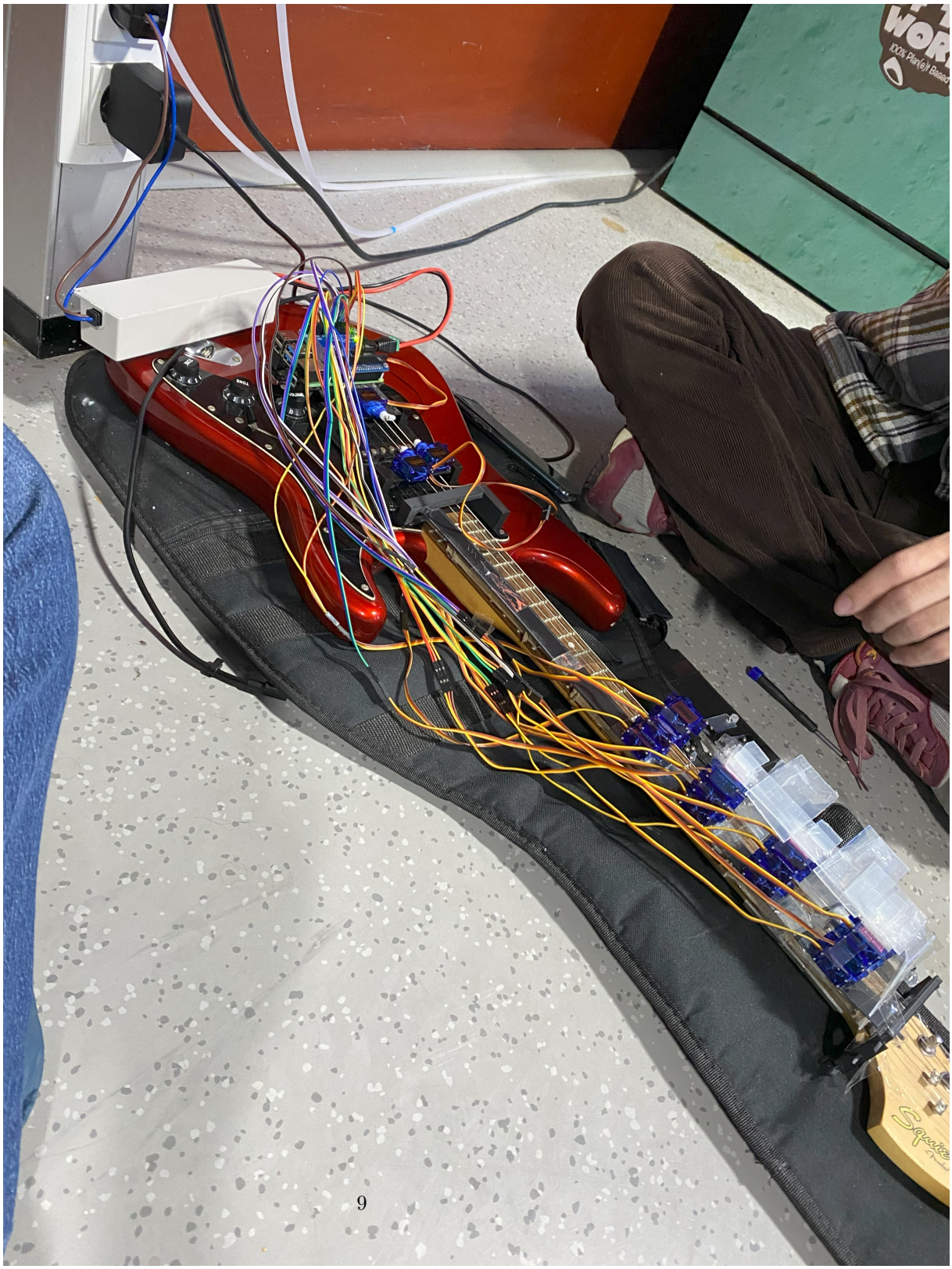

Figure 4: Image of the finished guitar fully wired up

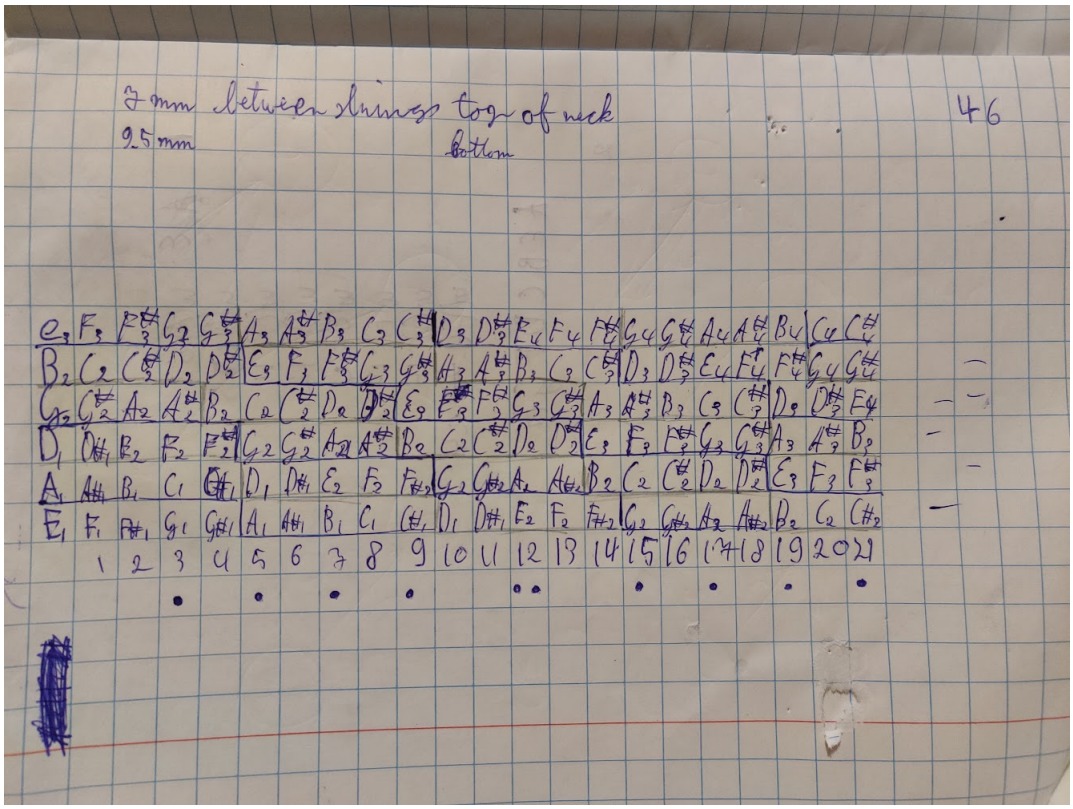

Figure 5: All notes on the fretboard of a guitar(Note: the notes have the correct letters but the numbers do not correspond to the standard notation here)

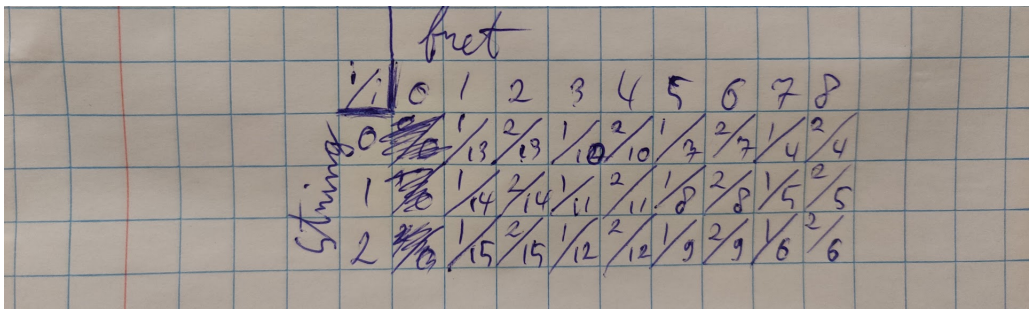

Figure 6: Representation of what servo corresponds to which fret on each string.

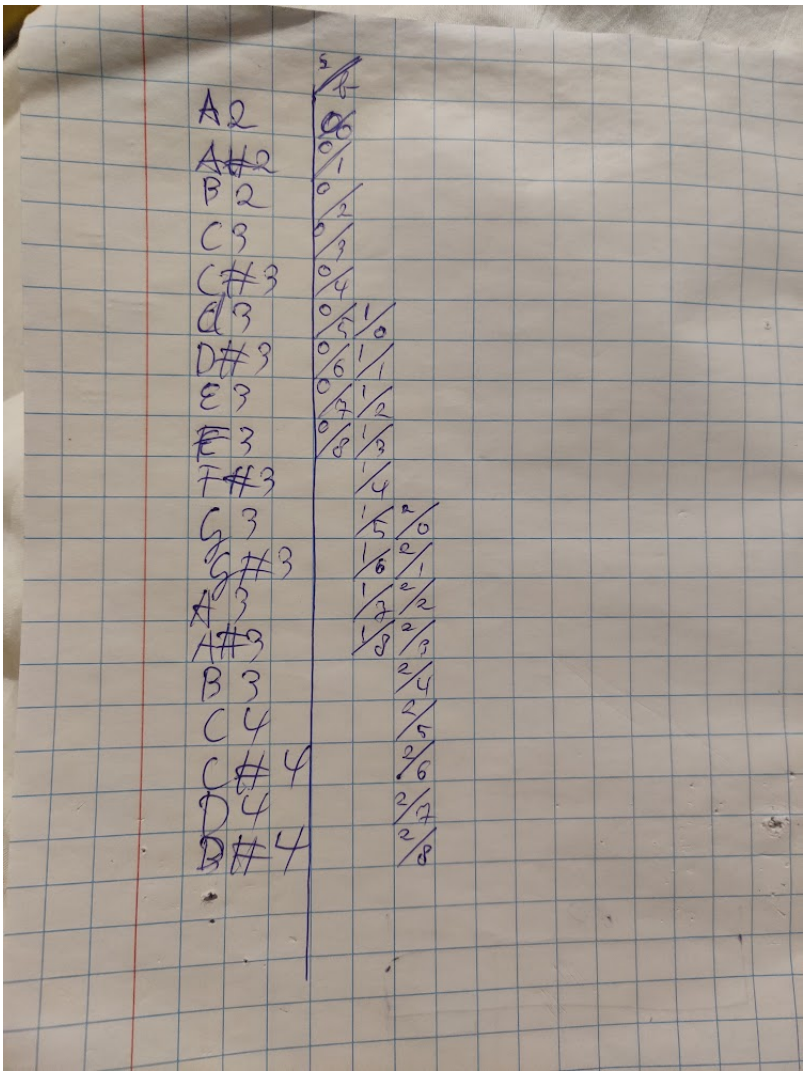

Figure 7: A mapping of notes onto their possible string/fret combinations

References

[1] Liz Clark. https://learn.adafruit.com/midi-controlled-robot-lyre-with-circuitpython/overview.

[2] GitHub. https://github.com/mido/mido.

[3] Olav Martin Kvern. https://makezine.com/article/craft/music/how-i-built-a-guitar-picking-robot/.