ME331 TERM PROJECT REPORT

Automatic Guitar Tuning Machine

Prepared by:

Abdullah Gedük
Alp Uysal
Ediz Ferit Kula
Elif Hacıhasanoğlu
Kevser Didar İskender
Sina Atalay

Brief Summary

This report presents the process of building a system that tunes a guitar with the help of a circuit that consists of Teensy 3.2, LCD display and sound amplifier circuit, writing algorithms for processing signals from a microphone and controlling the tension of the strings with DC motors according to the desired fundamental frequency. The main aim of this project is to tune the guitar strings with a way that is not only user friendly but also accurate. For the most crucial part of the project, that is the frequency reading precision and efficiency, Fast Fourier Transform is used.

CONTENTS

1. Introduction
- 1.1. Purpose
- 1.2. Method

2. Demonstrator
- 2.1. Block Diagrams
- 2.2. Software
  - 2.2.1 Target Frequency Selection and Directing the User
  - 2.2.2 Signal Processing
  - 2.2.3 Motor Control
  - 2.2.4 Simulation
- 2.3. Hardware
  - 2.3.1 Amplifier
  - 2.3.2 H-bridge drivers, Inverters and Motors
  - 2.3.3 Microcontroller
  - 2.3.4 Power Supply
  - 2.3.5 Board Schematic
- 2.4. Production Steps
  - 2.4.1 Design
  - 2.4.2 Simulation and Testing
  - 2.4.3 Production

3. Conclusions and Recommendations

4. References
- 4.1.References
- 4.2.Roles of Team Members

Chapter 1: Introduction

1.1. Purpose

The purpose of this project is to search the possibility of implementing a frequency detection algorithm in association with six DC motors and a user friendly target frequency selection system and to explore the accuracy of different sampling rates to achieve the most reliable tuning of the different strings on a guitar.

1.2. Method

Teensy 3.2 is used instead of Arduino Uno in order to increase the resolution of FFT method. The signals from the guitar strings are received from the microphone and amplified from the amplifier that is created from zero with different circuit elements.
3 H-Bridge Drivers and a HEX Inverter are used to control the direction of DC motors.

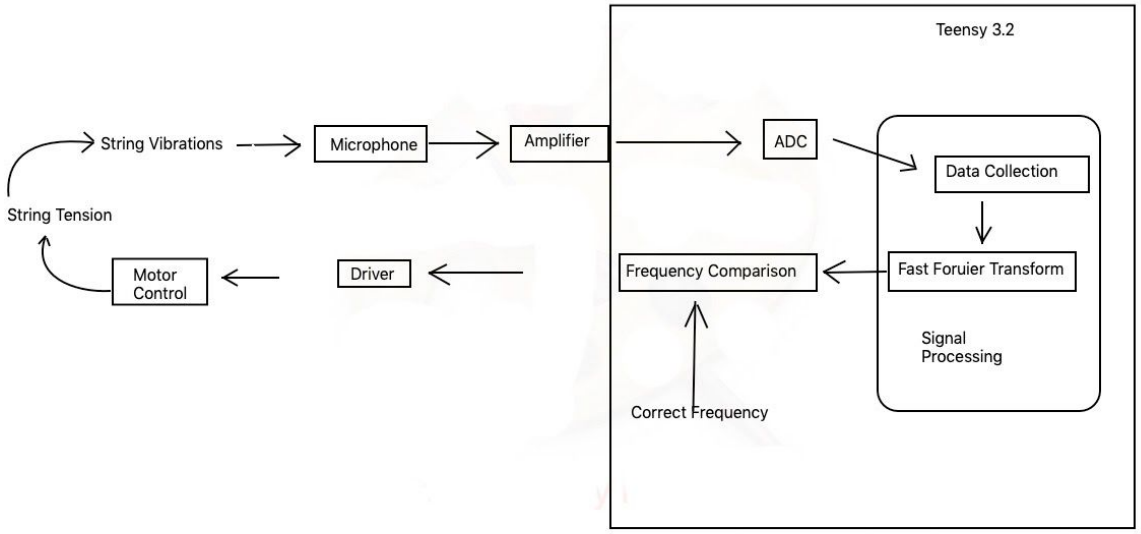
The design consists of two parts, the breadboard complex and the part that connects the motors and the guitar tuning pegs. The nozzles that are used to turn the pegs are 3D printed and the plates that are used to combine the system are cut from plastic plates with angle grinder and drilled with the upright drilling machine.

Fast Fourier Transform method is used for detecting the frequency of the signal from the microphone (magnetics of the electric guitar). Different sample rates are examined throughout the project in order to develop the coding of the system.

H-Bridge drivers are used to control the DC motors according to frequency difference detected by the code and Fast Fourier Transform.

Chapter 2: Demonstrator

2.1 Block Diagrams



General Tuning Mechanism of Automatic Guitar Tuner

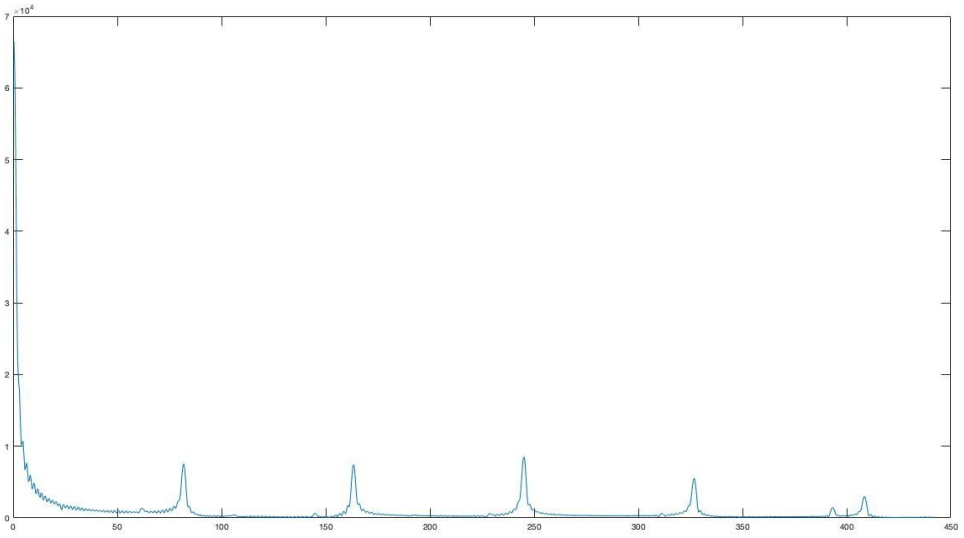
2.2 Software

2.2.1 Target Tuning Selection and Directing the User

Guitar players may prefer different tuning configurations depending on the song they play. That's why; we do not want to be restricted with the standard configuration (E, A, D, G, B, E), but instead, we increase the number of tuning configurations that the user can apply with our product. We achieved this goal thanks to the LCD and three buttons. We created tuning configurations and tuning names matrices that include the desired tunings and their names. Initially, standard tuning configuration is held within the target frequency matrix. Whenever, user pushes the buttons, the target frequency changes and goes back and forth in the frequency matrix. If the user decides which configuration is to be selected, s/he can press the main button to go on with the tuning process. During the selection process, user can see the current configuration on the LCD and if the selection button is pressed, "Target tuning is…" message is going to be seen for user to understand that tuning is going to start if strings are played.

While tuning process goes on, LCD shows the current situation of tuning. We start with the first string and the process goes through the sixth string one by one. LCD shows the current frequency and the target frequency while the motor is rotating. Current frequency value is updated till targeted value is reached. When the string is tuned, user is informed via the message “String is done, play next!”. After all the strings have finished, the user gets the message “Tuning is done!”. It can be seen that usage of LCD provides both multitasking in frequency configurations manner and user friendliness with its capability of directing and informing the user during selection and tuning processes.

2.2.2. Signal Processing



FFT graph of the first string

Microcontrollers that are used in the project cannot read values unless it is digital data since they can only see levels of voltage. Analog to digital converting is an important process in which an analog variable changing continuously is turned into a multi-level signal. Digital signals are sequences of discrete values that depend on sampling rate.

At first, the microphone (magnetics of the electric guitar) extracts the sound signal. Then it goes to the amplifier to be increased in amplitude manner to provide the analog pin to read the signal. ADC starts converting that analog signal to digital signals by sampling the signal. Then, signals are quantified and binary values are set. The sampling rate represents the number of samples the system collects within a second.

Digitalized data goes through the FFT process. Fast Fourier Transform is an algorithm which converts a signal from its original domain to a representation in frequency domain. In our case, original domain means time domain. The question is why do we use it? Theory of sound states that any signal that we can imagine is a superpositioned form of sinusoidal signals and those sinusoidal signals have constant frequency throughout the signal. When we play a string, a signal with a complicated shape forms. That signal is the superpositioned form of the natural frequency and also harmonics of the natural with different amplitudes. We need to extract the natural frequency from this composed signal for using through the tuning process. By FFT algorithm, complicated shaped signal can be divided into its sinusoidal components. Those components are represented in a frequency domain graph and the natural frequency can be seen easily due to its highest level of amplitude in the expected region. Harmonic product spectrum is used to detect the natural frequency via eliminating the harmonics.

Resolution of frequency domain depends directly on sampling frequency and the number of samples. As the sample number increases and sampling frequency decreases, we get more accurate frequencies. We can not randomly pick any sample number due to memory limitation, which is 2048 samples in our case. We need at least 0.45 Hz accuracy which requires our sampling frequency not to be bigger than 920 Hz. 883 Hz is picked to be more precise. However, sampling 2048 points with 883 Hz would take 2.32 seconds and it is a very long duration of tuning process. We searched for solutions for this problem and decided to only sample 512 points and assign all other 1532 points as zero. Results were promising; sampling and running FFT time reduced to 0.58 seconds which is enough for us.

Also, as an alternative to FFT, Autocorrelation method was considered. Autocorrelation method is a time-domain algorithm that finds the similarity between the signal and the shifted version of itself. Trial of this algorithm gave an erroneous result and we did not focus on it in order not to waste time, therefore we decided to proceed with our trials with FFT.

2.2.3 Motor Control

Logic of motor control depends on taken frequency datas. Two steps should be decided to control the motor: Direction of rotation, duration of rotation between each frequency measurement. Whether frequency differences are positive or negative, direction of rotation is defined.

$$\text{Frequency Difference} = |\text{Measured Frequency} - \text{Desired Frequency}|$$

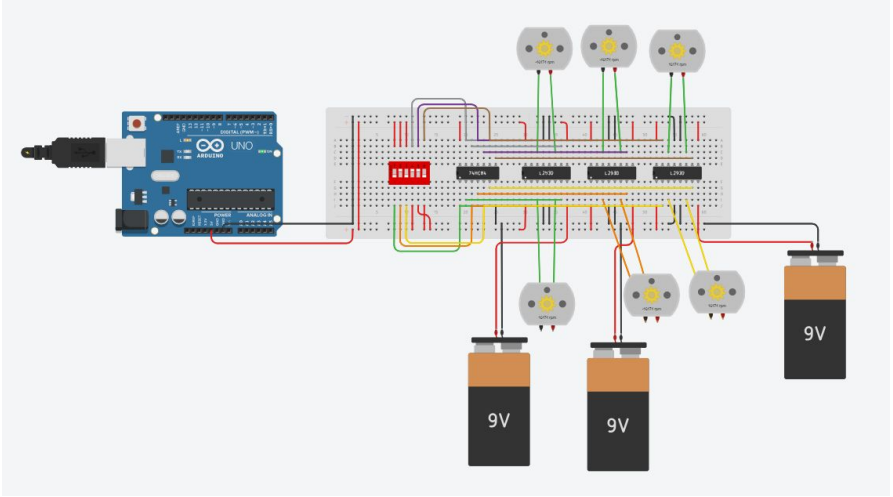
If the difference between measured and desired frequency is greater than zero, string should be unstretched. Opposingly, string should be stretched if the difference is smaller than zero. We defined a frequency tolerance value as 0.9 for stopping the motor. If the difference gets within the tolerance value, the motor stops.

Duration of the rotation motor between each frequency measurement calculated by closeness to correct frequency value. By decreasing the frequency difference, duration between each measurement is also decreasing linearly. Getting close to the correct frequency requires more attention for being accurate. That's why; duration is wanted to be decreased if frequency difference is small.

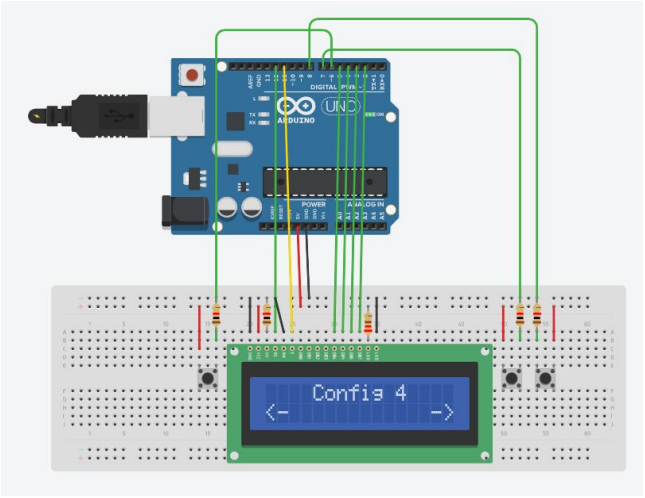
Optimized motor control is achieved by testing the system with different duration of rotation. Our aim was to approach critically damped states and prevent oscillations that may occur in the underdamped state. Also being in the overdamped zone would increase tuning time dramatically. A function which provides both a good settling time and low oscillation is generated to overcome these issues and implemented in the code.

2.2.4. Simulation

Before the implementation of a physical circuit, testing is done by simulations for LCD and Motor Control Circuits. Seeing the outcome of the designed circuit enabled us to foresee possible errors that may occur in our physical circuit. Also, with the help of simulation it's seen that any possible error does not stem from hardware components.



Simulation of motor circuit



Simulation of LCD circuit

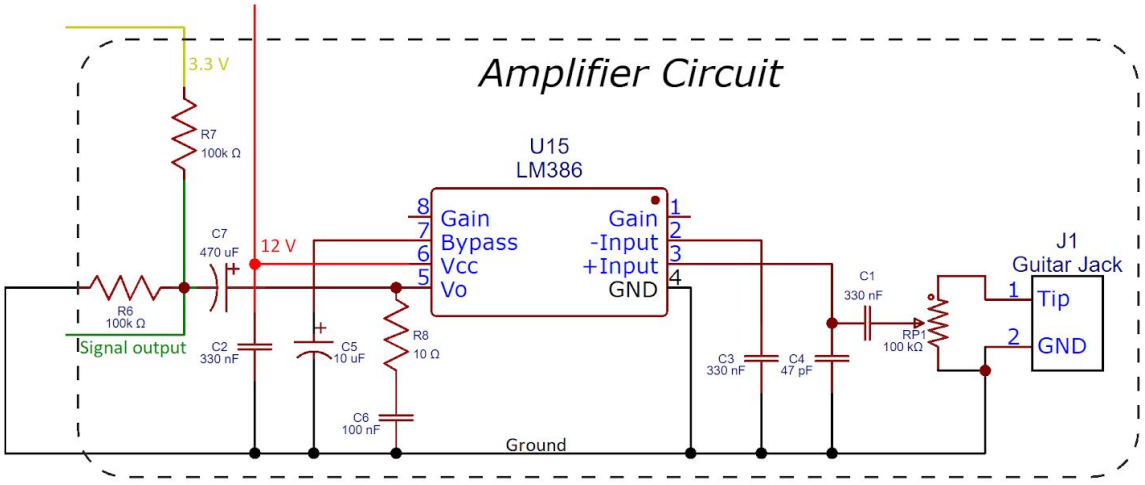
2.3. Hardware

2.3.1 Amplifier

Sound signals are received from an electric guitar which can transform the mechanical sound wave to an electrical analog signal with the help of magnetics. However, since the electro guitar is not supplied with any external voltage, this analog signal's power is not strong enough for Teensy to detect. The power of the sound signal must be increased with the help of an operational amplifier.

In our first trial, TL082 integrated circuit was used and we managed to amplify our signals. However, our frequency measurements were not good enough. We suspected noises which are very common for amplifier circuits. In order to analyze the quality of our amplifier circuit, we connected our amplifier's output to a speaker, so that we can hear the signal and check if it's good enough. We concluded that TL082 is not suitable. We decided to conduct our research.

In conclusion, we created an amplifier circuit with LM386 with a lot of external circuit elements to reduce noises. In our final setup, we could even listen to music when we connect our phone as input and connect a speaker as output without hearing any annoying sounds. Capacitors are used to filter some noises and fluctuations.



A 330nF capacitor is added to the input signal to avoid any DC signal. We don't receive any DC signal from the guitar, however it is added to be sure that the system does not amplify any DC signal. Next 47 pF capacitor is added across the input to short circuit any high frequency signal that we would like to avoid. It acts like a low-pass filter and keeps the high frequencies out of the amplifier. A 470 uF capacitor is added to output in order to avoid any DC signal to pass, again. Moreover, we add a 320nF capacitor between 12V supply voltage and ground. Because the same 12V supplies DC motors and other components, 12V can deflect time to time. With the help of a capacitor, we can short some of the AC supply, and receive pure DC 12V supply.

The other important thing is, LM386's output signal oscillates about 0V, however, teensy cannot read signals below 0V. So we have to make that signal oscillate about 1.65V since teensy can read values between 0V and 3.3V. We can achieve this with a voltage divider and it is made from two 100k resistors wired in series from the Teensy's 3.3V supply to ground. Since the resistors have the same resistance, the voltage at the junction between them equals 1.65V. This makes sure that amplifier output oscillates about 1.65V. We have to adjust the potentiometer so that maximum voltage does not exceed 3.2 volts or 0 volts.

2.3.2. Motor Selection, H-bridge Drivers and Inverters

1. Motor Selection

For tuning process to be successfully achieved, two features are expected from motors:

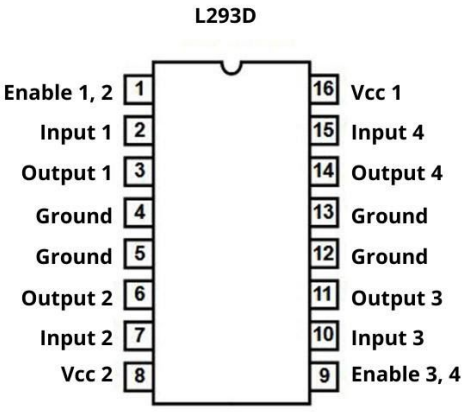
- High level of torque
- Continuous rotation
- Low rotation speed

Since servo motors provide mostly 180 degrees of rotation, they need to be converted to fully rotational motors by cutting the small metal / plastic bulge inside of them. This process would cause us to waste time and seemed risky. We started our trials with a 107.9 N*cm, 360 degree rotational servo motor to see what happens. The motor couldn't rotate the peg of the guitar. That experiment led us to buy a stronger, fully rotational motor. We tried the 147.15 N*cm, 360 degree rotational (continuous rotation) servo motor and got a successful result. However, servo motors with that specialities were expensive. It was suitable for one motor design but not so much for six motor design. On the other hand, DC motors with high gear ratios. could provide high levels of torques at low speed and this was a crucial point for the tuning process. Therefore, we decided to go with DC motors since they were both cheaper and stronger at low speeds. The model (JGA25-370) that we are using now stands up to a torque of 196 N*cm which provides enough torque to turn the peg.

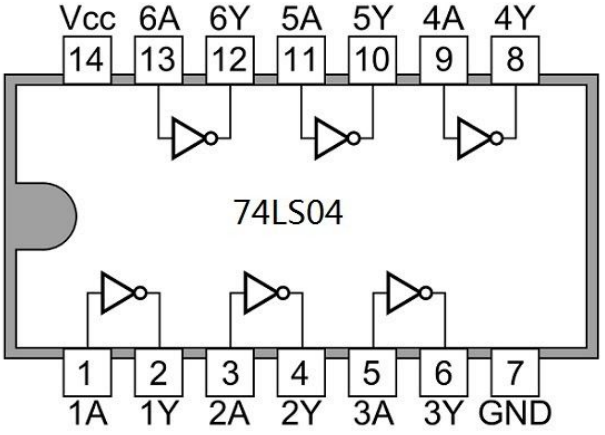
Current Model : 12V 26 RPM Brushless DC Gearmotor

2. H-Bridge Drivers and Inverters

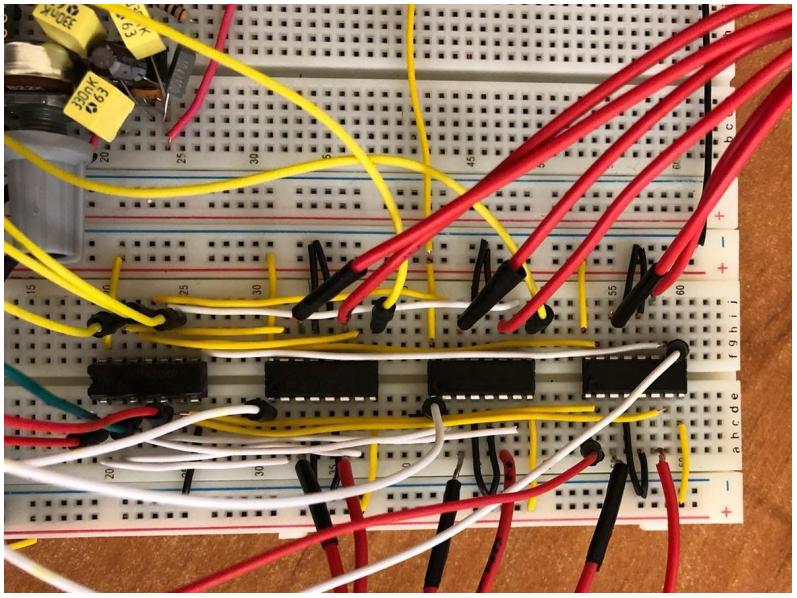
L293D motor drivers are used for controlling DC motors since we are familiar with the functioning of it. L293D motor driver's input pins control the rotation of motors and enable pins provide functionality to drivers. One of the input pins should get a signal of high voltage, and the other a signal of low voltage. Normally, two pins are used for controlling those signals. However, six DC motors are being used in our project and using eighteen pins total (12 input + 6 enable) poses a problem for us. Therefore, we created a simple 1-Bit Decoder circuit with the use of NOT gates to decrease the total pin usage by implementing the logic for each motor. For each motor, we can use only one pin with the help of NOT gate. If the value of the input signal is 1, the other input pin gets the value of 0 which comes from the NOT gate. Opposingly, if the value is 0, the other pin gets the value of 1. As a result, we can control six DC motors with twelve pins (6 input + 6 enable) thanks to 1-Bit Decoder. That reduction is important for us because of the scarcity of the number of pins available in Teensy board.



Pin Arrangement of L293D Driver



Pin Arrangement of 74LS04 Hex Inverter



Physical Representation of The Motor Controlling Circuit

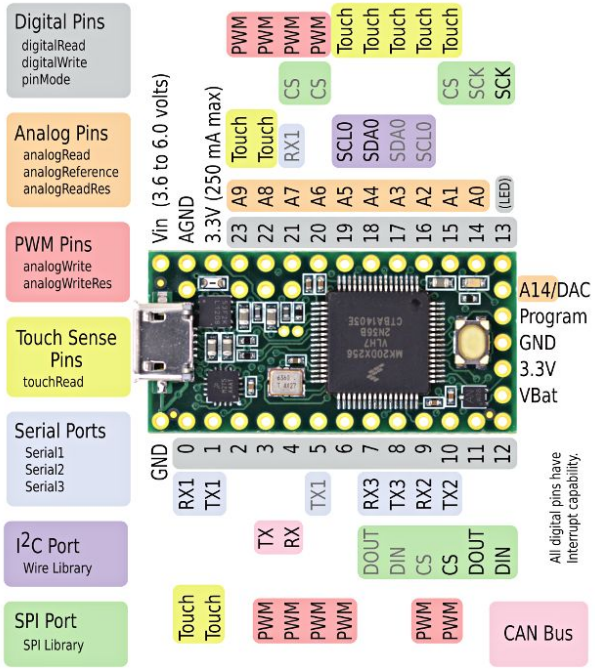
2.3.3. Microcontroller

a. Arduino Uno

Arduino Uno is a microcontroller based on ATmega328P. It has a flash memory of 32KB, 2KB of RAM ,14 digital inputs (6 of them can be used as PWM outputs) , 6 analog inputs. Arduino Uno was planned to be used in the project since it is a microcontroller we are familiar with. However as we proceed on the project, our research on fast fourier transform showed us that we will need more memory than the Arduino Uno provides us since Arduino won't work for samples higher than 128 due to lack of memory. This situation causes the accuracy of frequency detection to decrease, which is a crucial problem that affects our tuner's regulation.

b. Teensy 3.2

Teensy 3.2 is a USB controlled microcontroller development system with a high performance and a small size which doesn't take up so much space on the breadboard. It is compatible with arduino software and libraries. The most important technical spec of Teensy 3.2 that will benefit our project is it's 64KB RAM. This memory allows us to take 2048 samples that increases our frequency resolution to 0.4 Hz which is a great resolution for guitar tuning.

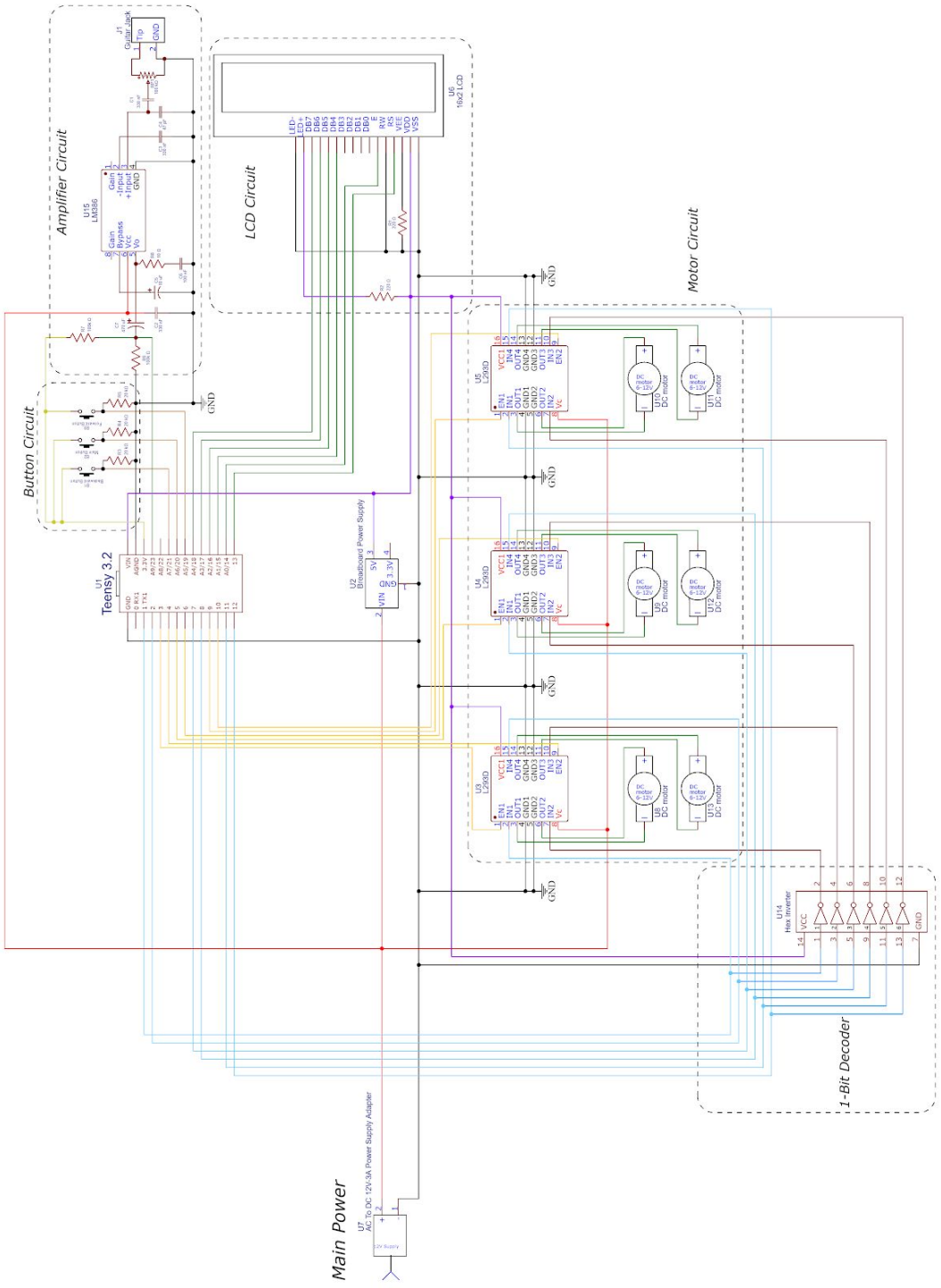


Pin Arrangement of Teensy 3.2

2.3.4 Power Supply

Our circuit consists of elements that require different voltage values. 5V powers Teensy 3.2, Hex Inverter, motor drivers and LCD, 12V powers the amplifier and DC Motor Circuit. The Breadboard power module is used as a power regulator and converts 12V to 5V. The 3.3V that is supplied by the Teensy 3.2 is used for the inputs of the motor drivers. The circuit is arranged so that the only needed power supply is the 12V adapter which is connected to the city electric.

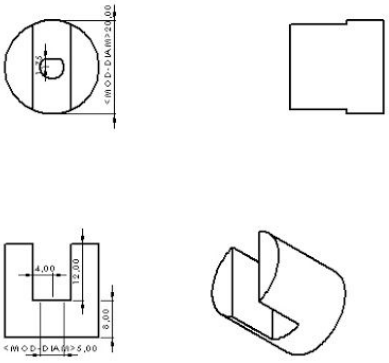
2.3.5 Board Schematic

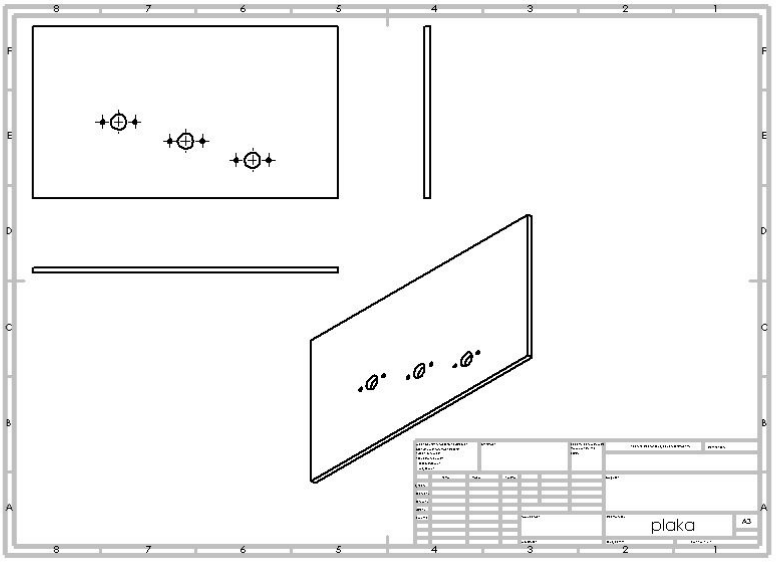


2.4. Production Steps

2.4.1 Design

First, our automatic guitar tuner consisted of a single 3D printed nozzle that was designed to fit the tuning pegs of a guitar and the shaft of a DC motor. A hand-held, non-guitar-integrated piece was drawn through Solidworks. These parts were designed to fit the motor shaft (4 mm diameter) so that it turns smoothly. Later, since we wanted to create a more functional part, a 6-motor combined system was designed. The guitar that the tests are conducted on throughout the process has different tuning peg sizes and nonlinear centerline for these pegs. For this reason, attention was paid to differ the lengths of the parts that hold the pins of the designed parts. In conclusion, the lengths of the printed parts designed as 32 mm, 28.5 mm and 20 mm. Second part of the design process is to design the side plates that hold the motors and the base plate. It is crucial to measure properly so that the gimlets are aligned. Two sides of our guitar differ in length and angle, so we shifted and mirrored the locations of the motor shaft holes on one plate. Each plate is supposed to hold 3 DC motors of 90 gram each. This led us to think of a design that allows the guitar to be tuned on a support with an additional plate that will connect the other two plates that will hold and stabilize DC Motors.



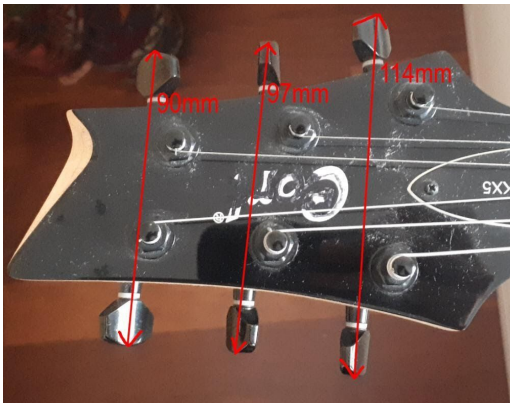


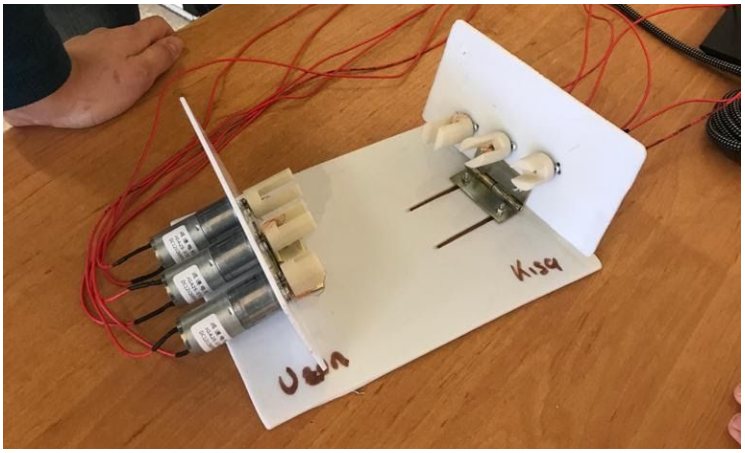
SolidWorks Drawings of the Nozzles(1st Figure) and Plates(2nd Figure)

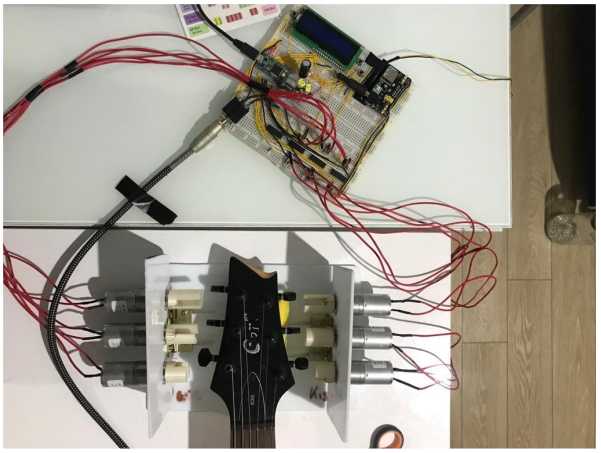
2.4.2. Production

The first step in the production process was to set up a 3D printer to print the drawn parts. As a result of our research on the internet, we agreed with the 3-D printer center named 3Piksel in Topçular Business Center. After the first prints were ready, we recognized that there was a problem of measurement between the first drawings and the engine. Since the width of the part where the part needs to enter the engine is smaller than expected, we developed the idea of using the part by heating and expanding it. However, as we thought that this solution could cause problems in the long run, we revised the drawing and printed 6 pieces of engine head parts again. Again in the second prints of the part we came across with a problem that they were now slightly wider than the expected engine head parts. In order to solve the problem, we increased the inner width by wrapping wire inside the parts. These parts, which provide the connection between the motors and the tuner pegs, were combined one by one for 6 motors, and then we started the production of plastic plates. We used the upright drilling machine in the BURST (Boğaziçi University Robotics Systems Team) workshop to cut the plastic sheet. First, two rectangular areas were created for the side plates, then suitable holes were created for the motor head parts and gimlets.

It was decided to establish the connection between the side layers and the base with a rail and hinge system in order to place the system easily according to the gimlet angles. A sponge has been placed under the guitar to obtain the proper height to connect the nozzles and the guitar pegs. In spite of the fact that it is difficult to obtain accurate measurements of the gimlets, a proper design for the base plate is achieved







Chapter 3: Conclusions and Recommendations

In spite of the fact that we have come across several problems throughout the project, with the help of teamwork the project has ended up successfully. We showed that a guitar can be tuned effectively for desired configuration via a mechatronics system and we provided a portable design which could help musicians who are away from their computers or cellphones. As it is mentioned before, implementation of Teensy 3.2 is a major advantage for data collecting. Thanks to its much more memory capacity than typical Arduino Uno, Teensy 3.2 has provided us with an effective way to get the strings tuned. For the design part, it is not wrong to say that obtaining measurements accurately is crucial to get the most long-lasting and useful design.

More work can be conducted on the methods of getting rid of the environment noise and to obtain more accurate results. In addition, a system in which all the six strings can be tuned at once. In order to do that we need to eliminate the harmonics of each string. To use the Harmonic Product Spectrum method for six strings, we need more resolution. More resolution could be achieved via more memory. Since Teensy 3.2 has limited memory, we weren't able to handle six strings at the same time. Even though this work can be done in systems with higher memory capacity, there could be ways to manage it by using real time FFT or algorithms other than FFT to analyze the frequency we read. We have proceeded with our research about methods in frequency analyzing and eliminating harmonics throughout the project, but we didn't include these studies into our project. Further research and studies could be made in these fields.

Chapter 4:

4.1. References

1. Teensy Technical Specifications
   https://www.pjrc.com/teensy/index.html

2. Arduino UNO Specifications
   https://store.arduino.cc/usa/arduino-uno-rev3

3. Pitch Detection Algorithms
   https://sound.eti.pg.gda.pl/student/eim/synteza/leszczyna/index_ang.htm#a3

4. FFT Features
   https://en.wikipedia.org/wiki/Fast_Fourier_transform

5. Market Analyzing
   https://stanford.edu/~mmhou/Guitar_tuner_instructable.pdf
   https://www.instructables.com/Arduino-Guitar-Tuner/
   https://www.roadiemusic.com/
   https://www.ece.ucf.edu/seniordesign/su2017fa2017/g21/AutoTunerConference.pdf

4.2. Roles of Group Members

Throughout the project, we have made weekly meetings via Discord. In the first month of the project, we searched different topics in order to get the most efficient outcome of the project. We assigned weekly goals to carry out a proper project. We had discussions and have found solutions to the problems all together. For the reporting part, the group members took responsibility for what they have specialized in. Although the fact that all the group members have a part in all the processes of the project and have knowledge about the related topics, every group member has a different area of specialization.

Abdullah Gedük:
Research and software of frequency analyzing and Fast Fourier Transform
Design, simulation, implementation and software of LCD and user friendly-interface
Design, simulation, implementation and software of motor circuit
Composite circuit design and proper power supply management
Schematics of composite circuit
Research of cost-effective ideas (External SRAM chip, 360 degrees Servo Motor etc.)

Alp Uysal:
Research of frequency analyzing and Fast Fourier Transform
Market analyzation for different type of tuning machines
Design, software and simulation of LCD and user-friendly interface
Research and software of motor circuit
Research on motor alternatives
Reporting

Ediz Ferit Kula:
Mechanical design
Production and assembly of mechanical parts
Testing the functionality of the product and strength of the materials
Design of the motor driver circuit and implementation
Participation in motor controlling software
Research about frequency analyzing

Elif Hacıhasanoğlu:
Research of Harmonic Product Spectrum
Technical drawing via SolidWorks
Production and assembly of mechanical parts
Design and schematics of the circuit
Implementation of the motor driver circuit
Implementation of LCD and user-friendly interface
Reporting

Kevser Didar İskender:
Research for 3D-Printing process
Technical drawing via SolidWorks
Research of Harmonic Product Spectrum and FFT alternatives
Design, production and assembly of the mechanical system
Research on motor alternatives
Reporting

Sina Atalay:
Research and software of frequency analyzing and Fast Fourier Transform
Design, testing and implementation of the amplifier circuit
Research of cost-effective ideas (DC Motor, 360 degrees Servo Motor, Op-Amp ICs etc.)
Implementation, testing and software of motor circuit
Testing other frequency reading algorithms
Implementation, testing and software of LCD and user-friendly interface
Reporting