Acoustic and electric guitar tuner

Name: Pavaloiu Bianca-Anastasia

Group: FILS 1221A

Introducere

Introducing my Arduino project aimed at creating a Guitar Tuner - the perfect solution for tuning your guitar with ease and accuracy.

This innovative project can be used with both acoustic and electric guitars, either through a microphone or by plugging directly into an amplifier jack. With automatic detection of each note and string as they are played, the tuner provides on-screen indications for tuning the instrument.

What sets my project apart is its ability to tune any guitar string without the need for special buttons. This makes it more cost-effective and easy to use compared to other projects that require multiple button presses.

In the past, I have faced issues with phone applications that are slow to open and often not as accurate as they should be. My Guitar Tuner solves this problem by providing fast and accurate tuning.

I believe that my Guitar Tuner is not only a valuable tool for musicians of all levels but also an affordable and easy-to-use solution that makes tuning your guitar a breeze. Try it out for yourself and experience the convenience and accuracy of our Guitar Tuner.

General Overview

This device is designed to help guitarists tune their instruments with ease and precision. By using a microphone to capture the sound of a guitar, the device can detect the frequency of each note played. If the correct frequency is reached, the device will indicate it, but if not, it will suggest adjusting the note through the LCD screen.

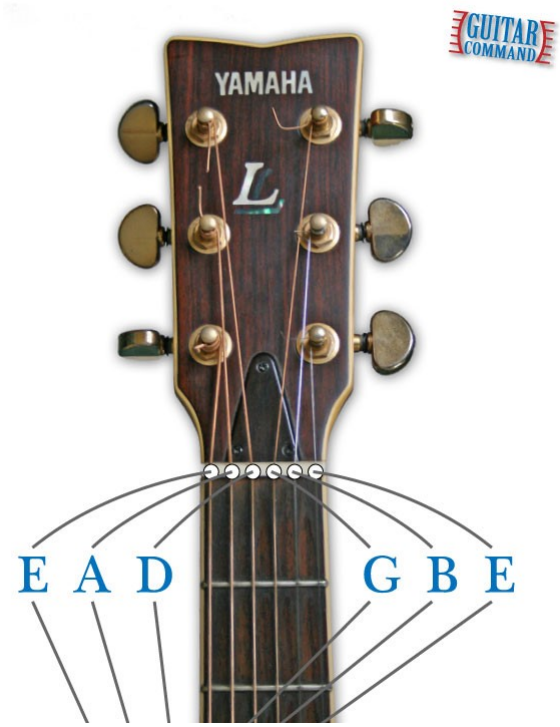
The process of sound analysis involves collecting discrete sound data every 1ms and storing it as voltage.This is possible due to the MAX9812 mic. module which has high sensitivity and low noise. Its integrated circuitry is specifically designed for low-voltage applications. Additionally, the MAX9812 features a built-in preamplifier that boosts the signal from the microphone, allowing for accurate and reliable sound detection. Its small size and low power consumption make it easy to integrate into a compact device like a guitar tuner.

The other input method for electric guitar is also really accurate. The process is similar, but the hardware is quite different. An LM386 amplifier with a jack provides a way for us to connect an electric guitar. The LM386 is a versatile and widely used amplifier that is commonly used in guitar amplifiers. After we take this signal we can input it into the Arduino using the analogRead() function to read the voltage level on the analog input pin. Then I use this value to determine the frequency of the guitar string being played, using the FFT (Fast Fourier Transform) algorithm.

The data is analyzed sample by sample to determine the peaks, which are used to calculate the frequency. By comparing these frequencies to the reference values of the six strings of the guitar, the device can provide accurate tuning suggestions for each string.

Its user-friendly design and advanced technology make tuning your guitar fast, accurate, and hassle-free.

Down below you can see a photo with the strings of a guitar.



Hardware Design

My project includes:

- MAX9812 microphone module with built-in amp 3.3-5V
- LCD screen with I2C
- 9V battery connector
- Audio jack
- LM386 Audio amp(for electric guitar)
- LED ON/OFF button

- Cables

Software Design

In order to tune a guitar, we have to get the sample of the frequency of the chord being played and compare it to the notes frequency we are trying to achieve. In order to make an accurate tuner, my project approaches this issue by implementing the tuner using FFT (Fast Fourier Transform). The core functions of my Tuner code are:

- FindFrequency() this function computes the frequency of our input signal(microphone of jack with amp) using the FFT algorithm available in arduinoFFT.h library

```
float FindFrequency() {
int GUITAR_IN = isAcousticMode ? ACOUSTIC_GUITAR_IN : ELECTRIC_GUITAR_IN;
for (int i = 0; i < SAMPLES; i++) {
  microseconds = micros();
  int sample = analogRead(GUITAR_IN);
  while (micros() < (microseconds + sampling_period_us));
  vReal[i] = static_cast<double>(sample);
  vImag[i] = 0;
}
```

```
FFT.Windowing(vReal, SAMPLES, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
FFT.Compute(vReal, vImag, SAMPLES, FFT_FORWARD);
FFT.ComplexToMagnitude(vReal, vImag, SAMPLES);
float peak = static_cast<float>(FFT.MajorPeak(vReal, SAMPLES,
SAMPLING_FREQUENCY));
```

```
Serial.println(peak);
return peak;
```

```
}
```

As you can see from the code snippet, the code first checks if is in acoustic mode or electric mode, in order to determine the input method. The program takes 128 samples(the maximum he can get according to the Arduino requirements) and stores them in a vector. Because we need also the imaginary part, that in our case is always equal to 0, we always initialize vImag with 0. Then we perform FFT analysis on the signal. Then we determine the frequency corresponding to the highest peak.

- displayNoteAndTune() based on the note frequency and tune will provide helpful instructions in order to tune the guitar

void displayNoteAndTune(float frequency) {

```
int note_idx = frequency_to_note(frequency);
String note=notes[note_idx];
```

```
lcd.clear();
```

```
lcd.setCursor(0, 0);
lcd.print(isAcousticMode ? "Acoustic Guitar" : "Electric Guitar");
lcd.setCursor(0, 1);
lcd.print(note);
```

```
if(frequency<noteRanges[note_idx][0]){
  lcd.setCursor(7, 1);
  lcd.print("Go Higher");
} else if (frequency>noteRanges[note_idx][0]){
  lcd.setCursor(7, 1);
  lcd.print("Go Lower");
} else {
  lcd.setCursor(8, 1);
  lcd.print("Good");
}
```

} }

- loop method

In loop method we first check for the type of input we have (in which state button is), then if necessary we will change the isAcusticMode value, based on the input we want. Then, we call the function FindFrequency(). If we have a match with one of the chords frequency, we can fine tune it to match the exact frequency by providing helpful instruction(we call the function displayNoteAndTune().

Concluzii

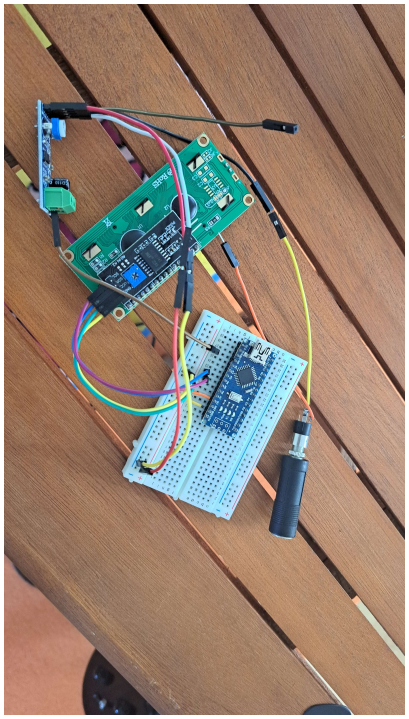
This project was an enjoyable experience, combining electronics and coding to create a guitar tuner. Building the circuit and connecting the components, such as the LM386 amplifier and the jack, was a rewarding hands-on task. The code development for frequency detection and tuning was challenging but provided a great opportunity to learn about signal processing techniques. Fine-tuning the frequency ranges and thresholds for each note required careful experimentation and testing. Overall, the project offered a fun and engaging way to explore both hardware and software aspects of building a guitar tuner.

Download

Acoustic and electric guitar-Source code pavaloiubiancaanastasiacode.txt

Jurnal

- 22.04.2023 after a lot of research, I chose my project theme, arduino guitar tuner
- 23.04.2023 found the components for my arduino project, made research for them
- 29.04.2023 made more research about FFT and how I can implement it in my code
- 03.05.2023 ordered the components online
- 05.05.2023 the components had arrived, began to update the first part of the ocw website for my project
- 07.05.2023 made the hardware part and tried to make the software part(a lot of mistakes needed to be corrected, unreliable code)



- 19.05.2023 made a diy case (out of a plastic box with holes made by me) for my project to protect the components and to make it look unique

- 20.05.2023 adding button to change input modes (acoustic or electric), button software implementation

- 25.05.2023 finished with the ocw website

Bibliografie/Resurse

Projects and sources that helped me make my project

1. https://circuitdigest.com/microcontroller-projects/arduino-uno-guitar-tuner
2. https://www.instructables.com/Arduino-Guitar-Tuner/
3. https://www.instructables.com/Guitar-Tuner-With-an-Arduino/

Export to PDF