

Arduino Guitar Tuner

Introducere

This is a project that has as an end goal the creation of a functional guitar tuner using Arduino. Its purpose is to be able to detect the sound that each guitar chord emits, and so identify whether the sound is in tune or not (based on its frequency). Musical instruments have always been a great part of my life, just as programming is. My idea is to combine my two passions into a device that is both useful and challenging to build. Hopefully, my device encourages other “musical programmers” to try out creating gadgets for their instruments; who knows? One day we might discover something revolutionary!

Descriere generală

The tuner works as follows:

The sound generated by the guitar is converted to a frequency, then it is compared with the frequency of the note “in-tune”. When the sound is emitted from a string, a yellow LED corresponding to playing that string lights up. Then, according to the accuracy of the note, one of 3 other LEDs will light up: a GREEN LED in the middle (if the note is in tune) / a RED LED to the left (if the sound's frequency is too low) / a RED LED to the right (if the sound frequency is too high).

Block Diagram:



Hardware Design

Components that were used:



Electric schematic:



The circuit:

The two main parts of the circuit are:

- the acquisition circuit where the sound is captured from the guitar and amplified
- the actuator circuit where the frequency is compared with the reference frequencies and leds give indications on the string being played and which way the string should be tuned

Acquisition circuit

This circuit has essentially 2 parts:

- Electret microphone preamplifier

The basic electret mic circuit consists of the mic itself, a resistor and a capacitor. That circuit is provided by the manufacturer.

- Operational amplifier

The op-amp is 1/2 of an LM358 dual op-amp and it operates off a single 5V supply

The amplification follows this formula:



V_s is the voltage coming from the mic 10 μF capacitor and V_o the voltage in the op amp output. R_f corresponds to the 100k Ω resistor and R_s to the 1k Ω resistor.

So, the gain will be $R_f / R_s = 100 / 1 = 100$.

Finally, the signal enters the Arduino A0 analog input pin.

Actuator circuit

The output is based on LED indications.

There are six yellow LEDs, each one corresponding to a different guitar string and each one connected to a different Arduino digital output (8-9-10-11-12-13).

The other three (one green and two red) are the indicators of the tuning accuracy.

The green LED, connected to the pin 5, lights up when the string is tuned in a very tight interval around the reference frequency and the other two red LEDs light up if the string is too tight (pin 2) or too loose (pin 7).

The string frequencies of standard tuning are:



As all LEDs are powered through 5V digital signals, and they need to have a voltage drop of around 2V, there should be a resistor producing a 3V voltage drop. Furthermore, the current through the LEDs should be around 20mA. So, the formula to calculate the resistor is: $R = V/I = 3 / 0.020 = 150$ ohms. The nearest resistor value we had was 100 ohms.

Software Design

The way the code works:

The code consists of two main parts, an autocorrelation algorithm that will be useful for detecting periodic signals, and a peak detector for detecting the peaks from the output of the autocorrelation function.

Distance between peaks = period of incoming signal.

The code calculates the frequency as:

```
freq_per = sample_freq/period;
```

Here are a few relevant snippets of my code (which can be found in full in the DOWNLOAD section below):

The setup:



Autocorrelation:



Peak detection:



Tuning sample for E low:



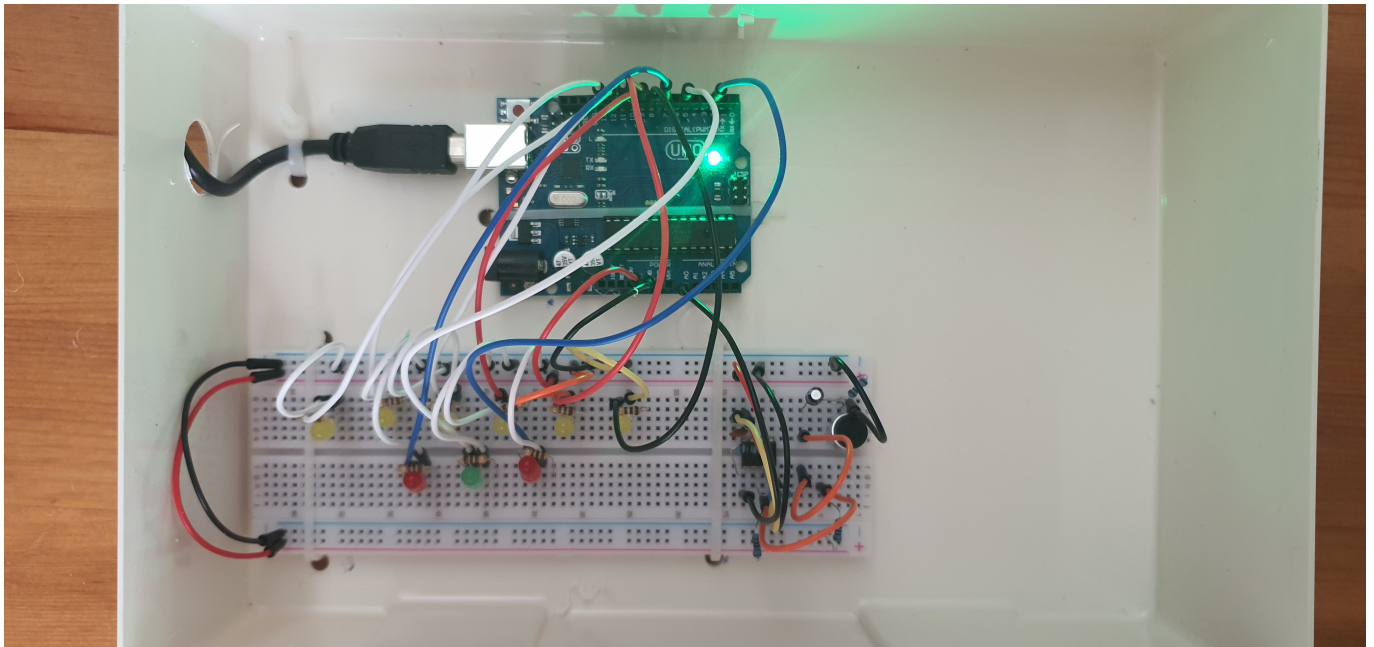
Disclaimer

Part of the code is based on a similar project in this website:

<http://www.akellyirl.com/arduino-guitar-tuner/> (link is found in Bibliography)

Rezultate Obținute

The final result: a functional Arduino guitar tuner:



Concluzii

This has been a pleasant and challenging experience that has taught me a lot about what can be done with creativity and a couple of wires (and the internet :D).

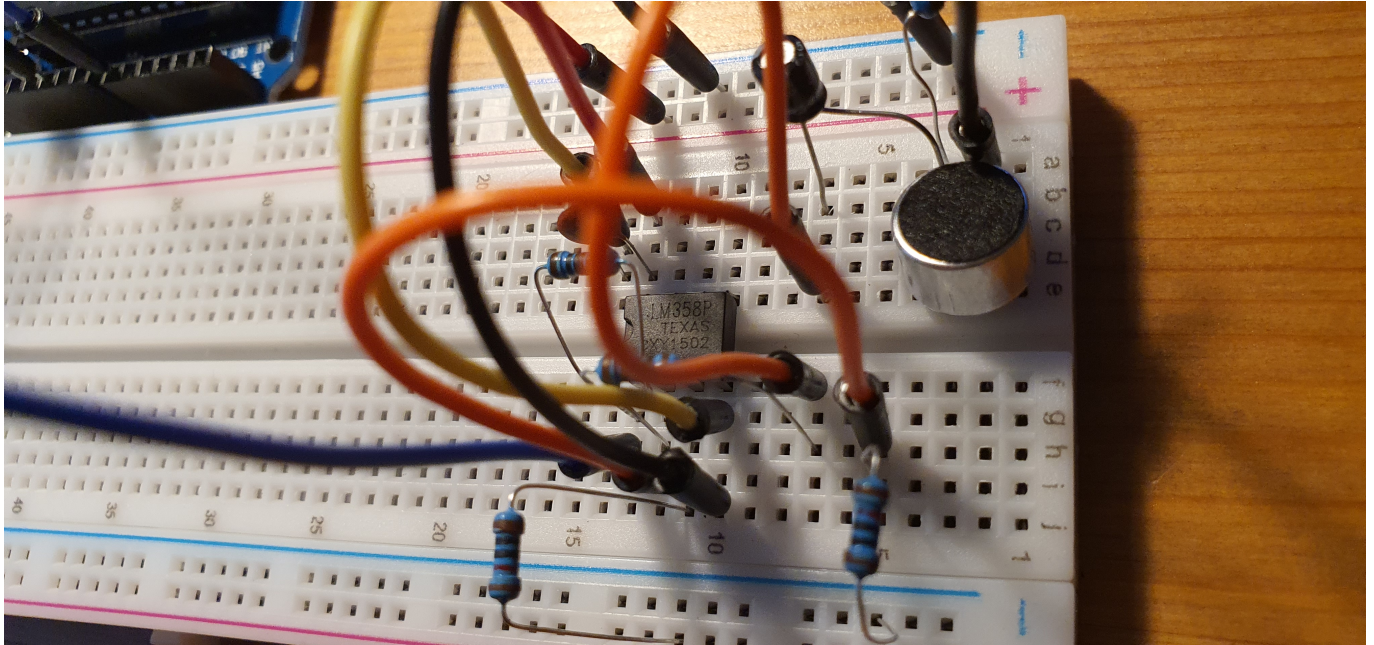
This has been a great opportunity for me to work with 2 things I like a lot (music and coding) and it has definitely inspired me to try out other projects like this in the future.

Download

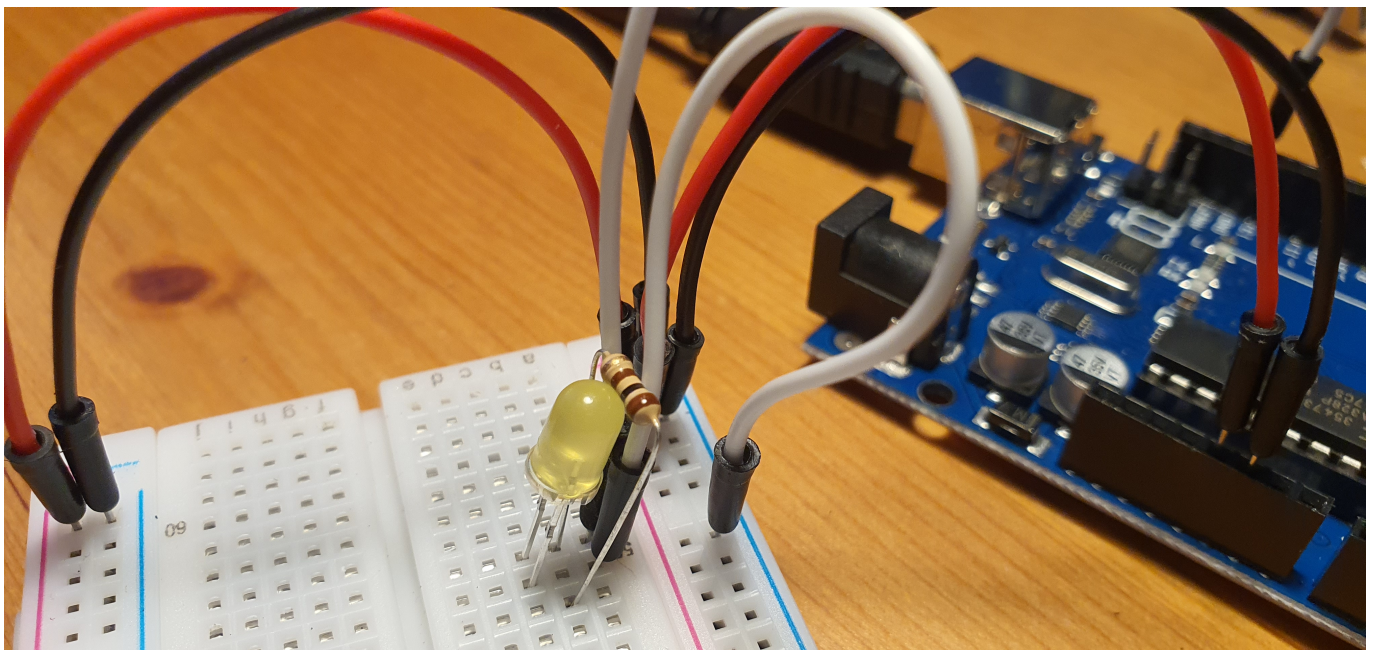
alberta-milicu_1221b_arduino-guitar_tuner.zip

Journal

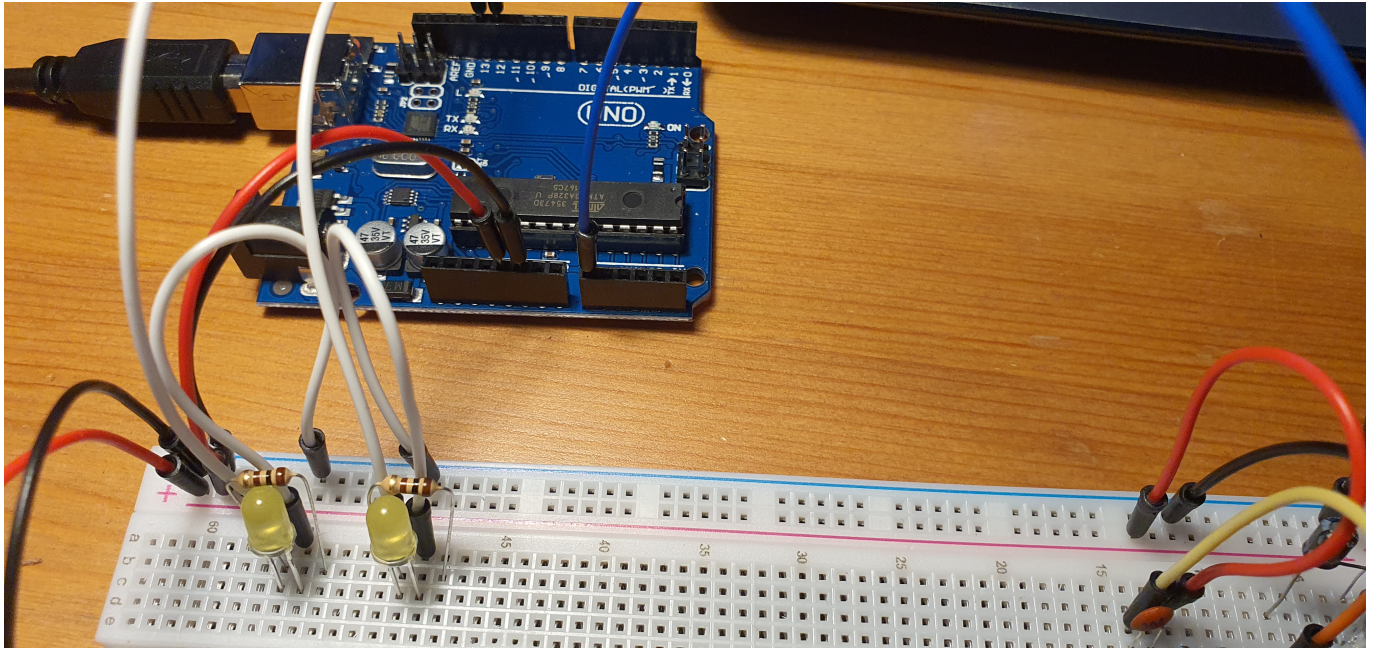
The set-up of the amp:



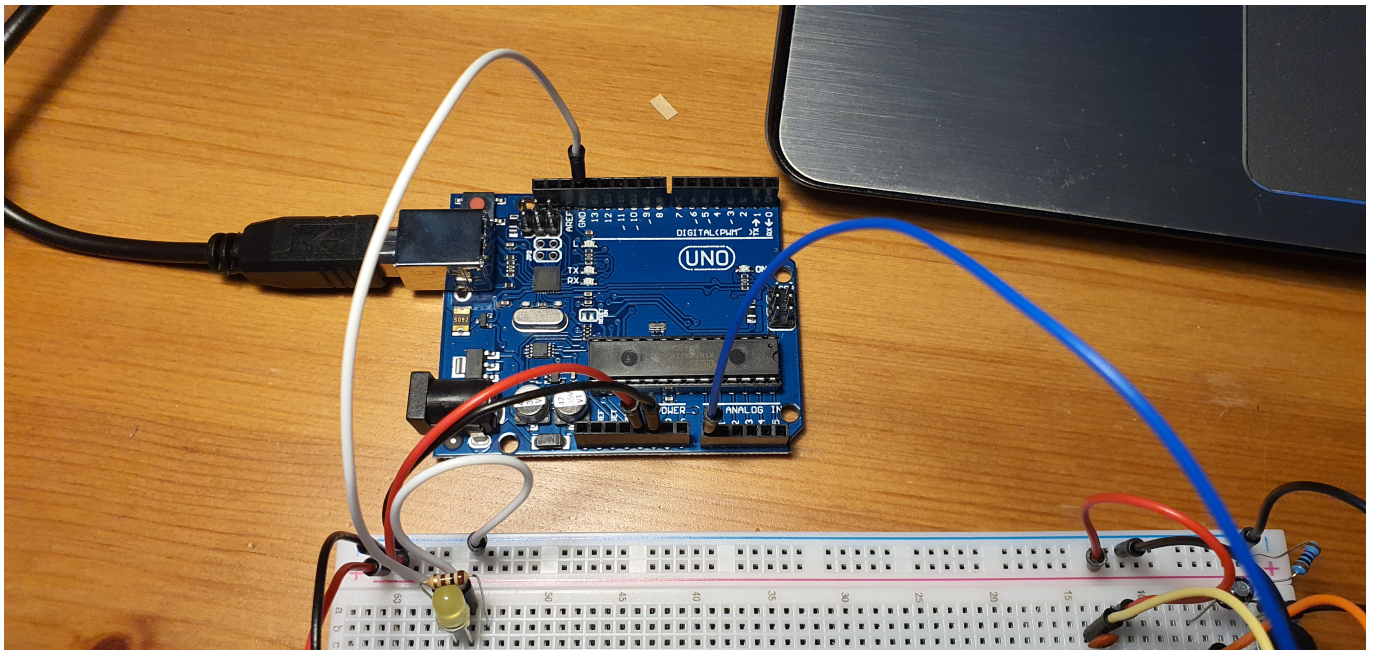
Setting up the LEDs (part 1):



Setting up the LEDs (part 2):



Connecting to Arduino:



Bibliografie/Resurse

Bibliography:

<http://www.akellyirl.com/arduino-guitar-tuner/>

<https://circuitdigest.com/microcontroller-projects/arduino-uno-guitar-tuner>

https://en.wikipedia.org/wiki/Guitar_tunings

<https://arduino guitartuner.wordpress.com/>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2022/agmocanu/guitar_tuner



Last update: **2022/05/29 11:54**