



**POLITECNICO**  
MILANO 1863

---

# Digital Guitar Tuner

[ Coding Project ]

---

**Student** Matheus Fim  
**ID** 876069

**Student** Lorenzo Binosi  
**ID** 876022

**Course** Advanced Operating Systems  
**Academic Year** 2016-2017

**Advisor** Federico Terraneo  
**Professor** William Fornaciari

June 5, 2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Problem statement . . . . .	3
1.2	Summary of the work . . . . .	3
<b>2</b>	<b>Design and implementation</b>	<b>4</b>
<b>3</b>	<b>Experimental evaluation</b>	<b>4</b>
3.1	Experimental setup . . . . .	4
3.2	Results . . . . .	5
<b>4</b>	<b>Conclusions and Future Works</b>	<b>6</b>

# 1 Introduction

The goal of this project was to deliver the code of a guitar tuner precise and fast enough to be satisfactorily used by all kinds of musicians. The main motivation behind the development of the code was enhancing the comprehension of real time operating systems and acquiring programming abilities in this area, such as handling threads, protocols of communication and other algorithms commonly used in operating systems.

## 1.1 Problem statement

Some minimum requirements for the guitar tuner were accorded by the group in order to guarantee the quality of the project. A satisfactory tuner should deliver consistent measures with a precision of at least 3 hertz. Also, it should be fast enough to give real time feedback of the tuning, more specifically it can not leave the user with the feeling of being slowed down by the hardware.

For the execution, it was used a STM32F407 Discovery board and a single-chip USB-TO-UART Bridge to provide the real time communication. As a starting point it was handled the driver for the microphone embedded in the board. And, from there on, the group had to work towards being able to translate the captured audio into reliable sound frequencies. Those frequencies would then be used to possibillitate the tuning algorithm and a simple user interface.

## 1.2 Summary of the work

The project had five milestones: Firstly, the driver of the microphone had to be updated to the current version of Miosix; Secondly, the Fast Fourier Transform (FFT) algorithm had to be implemented using Miosix compatible libraries; Thirdly, the initial achieved resolution was of 10.7 hertz, which is far from good for a tuner, so a higher decimation had to be implemented in the microphone; Then, the treatment of wrong measurements, such as harmonics and frequencies out of the curve had to be carried out; and to finalize, a simple user interface to choose the desired tuning and actually execute the work was developed.

As it is expected, the main difficulties the group faced were related to the second and third milestones. In the second, the fft function makes use of specific mathematical functions that needed to be optimized to support the real time requirement of the project while still using the maximum number of samples possible. And in the third, the increase of decimation involved key concepts of both the way the hardware is handled and digital sound.

## 2 Design and implementation

For the design of the project it was kept in mind the “low coupling and high cohesion” principle of software design. That being said, it is possible to notice specific files that deals with each part of the tuner. In this way, it is simple to reuse the code in the future and also guarantee an easy maintenance.

The interaction of the user goes from starting the tuner, choosing the desired tuning, and one string at a time, see if he should bring the pitch up or down. When satisfied, the user can press a button in either the keyboard or the board itself and move forward to the next string. Some cautions are taken, for example, if the frequency received is too far from the one it should be measuring, that probably means the wrong string is being played, and a warning is issued.

Bearing in mind the desired user experience, the flow of the program consists of the microphone's driver recording the audio in samplings, and as soon as a buffer is full it is created a new thread that is passed to the FFT function. The function, by its turn, translates the vector of 16-bit PCM audio information into a frequency value. And for the following part, the frequency is compared with the desired one and the feedback is displayed to the user.

In order to discuss what and how was implemented it is of easier comprehension if each functionality is addressed individually. Therefore, starting from the FFT function, it was researched and imported several ARM library functions related to the mathematical operations needed, those functions proved to be optimized, as they were not only orders of magnitude faster than the others available in the literature as they also achieved the goal of execute all the calculations in real time. The FFT function can handle successfully 8192 samples, that is, the 4096 that are provided by the microphone driver plus the imaginary part of each. This initial set up of FFT function and the original microphone driver resulted in a frequency resolution of 10.76 hertz according to the formula !!!!!!!!!!!!! .

Moving on to the Microphone driver, it was made the update of the code to make it compatible with latest Miosix kernel. The changes consisted of fixing the callback that due to a difference in the way exceptions were being handle was not working and also renaming functions and variables. Concerning the frequency resolution, it was increased the decimation factor in the PDM filter from 16 to 64 bits, by dropping 3 out of every 4 samples measured. In this way it was achieved a frequency resolution of 2,69 hertz.

To finalize the project, the previously described user interface was implemented concurrently with an algorithm that ignores frequencies that are above a specified delta between the latest measurements, as a form of ignoring background noise and misread harmonics.

## 3 Experimental evaluation

The experiments were divided into two categories. In the first category, a more technical approach was used and consisted of various frequencies being played by computer speakers and compared to the results given by the tuner, this provided solid information on the reliability and consistency of the measurements. In the second one, aiming real world situations, real guitars were tuned both with the developed tuner and other tuners sold in the market, providing information on non-functional aspects of the system, as if the response time is adequate.

### 3.1 Experimental setup

In the first phase, were played all the frequencies that matches the musical notes contained in any of the available tunings, also the frequencies 5 hertz above and below and at least two harmonics of each note. The frequencies corresponding to each note can be seen in tables 1 and 2.

Table 1: E Standard

Note	Frequency	Harmonics
E	329.6	659.2 and 988.8
B	246.9	493.8 and 740.7
G	196.0	392.0 and 588.0
D	146.8	293.6 and 440.4
A	110.0	220.0 and 330.0
E	82.4	164.8 and 247.2

Table 2: Eb Standard

Note	Frequency	Harmonics
Eb	311.1	622.2 and 933.3
Bb	233.1	466.2 and 699.3
Gb	185.0	370.0 and 555.0
Db	138.6	277.2 and 415.8
Ab	103.8	207.6 and 311.4
Eb	77.8	155.6 and 233.4

The experiment was repeated a total of three times, first in a quite environment, after in a normal one and at last in environment with loud background noise.

For the second phase, a guitar was tuned three times in each of the environmental conditions described above, totalizing nine repetitions.

## 3.2 Results

**Please make sure you explicitly cite the tables. Table 3 shows a complex table. Please also make sure you provide each table with an exhaustive caption. Captions for tables must be placed before not after the tables.** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula

Table 3: Summary of the test scenarios.  $M_1 \xrightarrow{\alpha} M_2$  means that an application mapping is changed from  $M_1$  to  $M_2$  after application  $\alpha$  has terminated.

Name of scenario	Description of the workload			Cores allocation	
	Application	Threads	$\frac{Threads}{Cores}$	HMP	HMP w/policy
LITTLE 1	ferret <sup>†</sup>	1	1.00	0 – 3	0
	vips	3		0 – 3	$1 - 3 \xrightarrow{\dagger} 0 - 3$
LITTLE 2	freqmine <sup>†</sup>	2	1.25	0 – 3	0 – 1
	blackscholes	3		0 – 3	$0 - 3 \xrightarrow{\dagger} 0 - 3$
LITTLE 3	bodytrack <sup>†</sup>	3	1.25	0 – 3	0 – 1
	facesim	2		0 – 3	0 – 3
LITTLE 4	facesim	3	1.50	0 – 3	$0 - 1 \xrightarrow{\dagger} 0 - 3$
	blackscholes <sup>†</sup>	3		0 – 3	2 – 3
big 1	vips	3	1.00	4 – 7	$4 - 5 \xrightarrow{\dagger} 4 - 7$
	ferret <sup>†</sup>	1		4 – 7	6 – 7
big 2	freqmine <sup>†</sup>	2	1.25	4 – 7	4 – 5
	blackscholes	3		4 – 7	$6 - 7 \xrightarrow{\dagger} 4 - 7$
big 3	facesim	2	1.25	4 – 7	4 – 5
	bodytrack	3		4 – 7	4 – 7
big 4	facesim	3	1.50	4 – 7	$4 - 5 \xrightarrow{\dagger} 4 - 7$
	blackscholes <sup>†</sup>	3		4 – 7	4 – 7

feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

## 4 Conclusions and Future Works

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## References

- [1] John C Mathes and Dwight W Stevenson. Designing technical reports: Writing for audiences in organizations. 1976.
- [2] Carolyn R Miller. A humanistic rationale for technical writing. *College English*, 40(6):610–617, 1979.