

## Realization of Microphone Volume Intensity Indication

Ondrej Krejcar<sup>1,2</sup> and Dalibor Janckulik<sup>1</sup>

<sup>1</sup> Department of Measurement and Control, Faculty of Electrical Engineering and Computer Science, VSB Technical University of Ostrava, Ostrava-Poruba, Czech Republic, [ondrej.krejcar@remoteworld.net](mailto:ondrej.krejcar@remoteworld.net), [dalibor.janckulik@hotmail.com](mailto:dalibor.janckulik@hotmail.com)

<sup>2</sup> Department of Information Technologies, Faculty of Informatics and Management, University of Hradec Kralove, Hradec Kralove, Czech Republic, [ondrej.krejcar@asjournal.eu](mailto:ondrej.krejcar@asjournal.eu)

**Abstract.** In many desktop applications, an audio signal is captured from the microphone. Unfortunately, most applications lack quality visual indication of arousal microphone volume, which would clearly provide to users (surveillance) information on current values and, if so achieved within a short past. The usability of applications could be in using it as an acoustic monitoring or as a visual area, or volume level for the detection of a communication program. Usability is, of course, also in programs for recording and monitoring audio. The component is designed in .NET Framework.

**Keywords:** sound level meter, VU meter, sound, sound card, microphone

### 1. Introduction

To zoom the project physical foundation of a problem can be applied. In words of basic physics definition of sound, the sound wave is a periodic compression and expansion of flexible space in which it spreads, for example, air, water. The sound is perceived by a hearing body volume. Loudness is a subjective variable and depends on the sensitivity of hearing. For an objective evaluation of sound there was introduced so-called sound intensity  $I$  which is defined as the proportion of sound power  $P$  waves and area  $S$  through which the wave passes. According to the sensitivity of a human ear there are established two boundaries of sound intensity:

- threshold of hearing - it is the smallest registerable audio enjoyment -  $20\mu\text{Pa}$  (sound).
- threshold of pain - it's a border where the man begins to feel the pain and the sound can cause permanent damage -  $130\text{Pa}$  (sound).

To better distinguish the volume levels of sound there are used logarithmic coordinates in units of dB, because the human ear can detect the order of just 1 db [1].

To record or work up an audio microphone, which converts sound into an electrical signal, is used. Since each microphone is not able to process signals from a full spectrum, the different types of construction working at the other physical laws, according to the required properties. The most widely used electrostatic microphones are microphones that have a high output impedance, balanced frequency response, high sensitivity, low distortion and high stability of their properties. With the audio card they are connected to the computer. Because of the analogue input signal A / D converters are used in sound card [2].

The transfer takes place by the means of sampling. This means that at each time interval the current status of the signal is detected and recorded (sample). It is obvious that the shorter this interval is, the higher the sampling rate is, the more samples will be acquired and the resulting record will be better [3].

As a similar interpretation of this project let's include the so-called VU meter or PEAK meter, thus measuring devices showing the actual value (intensity) of audio signal. In former times it was formed by an analogue magneto device with calibrated scale, currently LED-vertical or horizontal scale (bars) are used more [4].

Because of the high intensity sounds audible 0-130 db, it should be restricted to certain range within which the excitation light is working. The best according to the human speech and in the range of 20-80 db. The sound is always tied to a particular frequency, the human ear's frequency range is 20Hz - 20kHz. At the same time the human ear is highly nonlinear in the frequency spectrum, it introduces non-linearity in the intensity of the volume [5]. It is therefore necessary for the project to select one frequency to avoid non-linearity characteristics. The ideal frequency is in the area around the human speech, which is around 2kHz.

Custom component is made from colour graphics bar that indicates the current excitation of the microphone on the sound card. The sensitivity of the bar is set according to the current level of amplification on the sound card. At the same time the last reached level of the bar is in time delays in order to achieve clearer level subtraction.

## 2. Problem Definition

Access to the problem of visual display component of volume excitation using .NET framework is solved by several other projects.

The project "Analog and LED Meter" focuses on the visual components of the VU meters, peek LED meters, and imaging through analog display visualization. The project is written in C # language and the output is implemented as a component in a dynamic library .DLL, which you can work with and apply the various bars and meters in other projects. However, the project deals only with graphic design but it is not focused on linking access to the signal source, a sound card. [6]

Another project which solves the problem is "Multimedia PeekMeter Control." According to the documentation this project includes more advanced approach to the problem, because it solves the sound frequency spectrum. The project is divided into blocks:

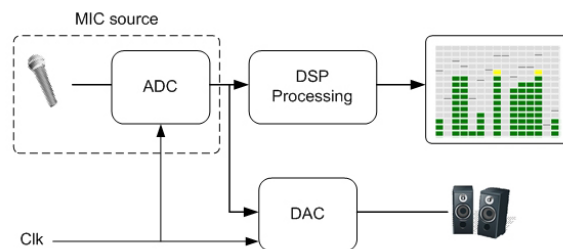


Fig. 1: Project architecture.

- ADC - A / D converter, which is used to convert the analogue signal from a microphone or from external sources to digital. There are several ways of processing the samples, mostly data PCM is used (Pulse Code Modulation) and using a sampling frequency of the CLK samples are accused.
- Processing DSP - this block gets digital samples from a transmitter and approximates them. Important role is played in this block by FFT (Fast Fourier Transform), which is used to analyze a digital signal. It differentiates between the different frequencies of phase and amplitude. Important it is to retrieve a peak amplitude and these then implement on the individual frequencies and visualize them.

DAC - D / A converter designed to convert digital signals back to analogue and subsequent reproduction.

The resulting application is able to retrieve data from various external sources of signals or from audio files. Then it prepares their spectral analysis using FFT and selects the peak values of the digitized signal. The resulting display is made by PeekMeter component that displays the different frequencies and their levels [7]. This project, however, does not meet our requirements due to the unnecessary complexity of the project, but its components may serve to further development of our project.

Project "LED Style Volume Meter Using DirectX" or volume indicator through LED indicators using DirectX. As it is evident from the title that is a component that uses the DirectX libraries. DirectX is a set of libraries published by Microsoft. It provides access to hardware by sophisticated manner. The project uses the library Direct3D for the visual part and DirectSound allowing access to hardware, then loading the

samples. In the documents there are written down various qualities that might be edited and modified according to user needs. Finally, the principal methods and characteristics of the project are brought in. [8]

Project "VolumeMeter (Managed DirectX)" are components for the excitation volume indication. It is made using DirectX and DirectSound libraries in C Sharp. It includes a visual indicator component stereo bar and component for listing the current streaming data in real terms. Application using these components is a part of the application. The whole project provides a wide range of parameters of individual components. This project is functional, but individual components cannot be used, because of unknown reasons, in the new application. The project will be subjected to detailed examination of its structure [9].

None of the approaches solves our problem completely. The closest solution includes project VolumeMeter (Managed DirectX), but the components do not work, only the application that uses them. Error, which generates, new applications using the component, is most likely related to the operating system.

Access to the problem of real-time indicator showing the microphone volume excitation is well described in the "Multimedia PeekMeter Control" in terms of speed of FFT and frequency resolution of the frequencies and their volume excitation indication. The optimum solution is to use dynamic libraries DirectX, especially DirectSound for a relatively easy work in C Sharp and with access to the hardware.

### 3. New Solution

Generally, for the work with sound, especially for recording of the use of so-called buffer (stack) or caches. The stack is basically a memory designed for the current portion of the signal that enters the computer which is used to retrieve the correct data in real time. We can talk about recording of the signal in real time. Analog audio signal entering the sound card first processes interface, which cannot affect the course because there would be a loss of input data. The computer is not processing only the audio signal, but it takes care of many other processes certainly with greater priority. Therefore, to avoid delays and various time delays, a distorted of the signal, the input data should be continuously stored in a buffer. When the computer performs more important instructions, it processes data from the reservoir or buffer.

The solution of our project is to use the Microsoft DirectX and DirectSound libraries, these libraries provide extensive work with the hardware, especially with sound. DirectSound library specializes in just the audio portion. It contains a lot of classes focusing on the hardware input or output, storage, exceptions for missed approaches and different classes for sound effects.

Basic work with DirectSound library: (1) Creating a hardware device objects; (2) Setting the level of cooperation; (3) Creating a buffer object; (4) Recording audio data to the stack; (5) Playing back buffer

By creating a class "device" we model, how an audio input / output will act with the audio signal. For example, setting the parameters of class device by signal sampling, setting the level of cooperation and "set a Cooperative level". Each application using DirectSound must have set the level of cooperation with other applications for the access to hardware or what priority the application has for the access to the hardware compared to other applications using the same source. Creating a container class constructors we enable the work to the buffer. Through this reservoir, "buffer", all the audio data passes. Buffer is nothing more than the allocated space in memory of the computer and the memory serves as a buffer. There are two types of reservoirs, primary and secondary. The primary reservoir is provided by the DirectX and the secondary reservoir sound is adjusted modeled according to our needs or effects. The easiest way to work with containers is to use a constructor that takes a filename as an argument. It then fills in a new buffer. To play a sound we approach, as well as for the record, through the buffer. Basically, we spool data from the buffer [10]. Creating components. Component is basically a dll library, that means a dynamic library. We can work with it as with any other library. Classes, objects, variables and properties are contained in. The great advantage of component (library) is that it is applicable in any other project. It good enough to add a reference and copy it to the project. Then we can use the component, run its methods and, where appropriate, before or during runtime of the programme change its parameters (properties). By creating component we create a portable and multiple times in different projects usable application. In our case, it makes sense to create especially a component, because excitation volume indication is not stand-alone application.

Component is formed as an application, however, the variable parameters of object we create objects using the gets and sets.

## 4. Solution Implementation

The principle consists in drawing LEDs and their subsequent gradual redraw by a different color. The delay effect of the last value is solved by setting a timer, and the tick interval.

The indicator bar is designed as a component. However, part of the access to the hardware is attached as an additional component, and must work together if the visualised signal is from the new card. Connection is done through the transmitting of the current volume level into the display bar.

The most important parts of the indicator bar is an implementation of LEDs colors, activated and switched off. Defining the parameters of each LED using methods, setting the value and returning value, or gets and sets.

## 5. Testing of Developed Application

When creating component, testing, and debugging was carried out in test application already during production. For final testing, and demonstration of components application was developed, where their main parameters and settings are used. Furthermore, the example of their possible use. The main test parameter is a determinism (to the same input signal corresponds with the same output parameters). However, using input parameters from the microphone we can never absolutely guarantee the same initial conditions.

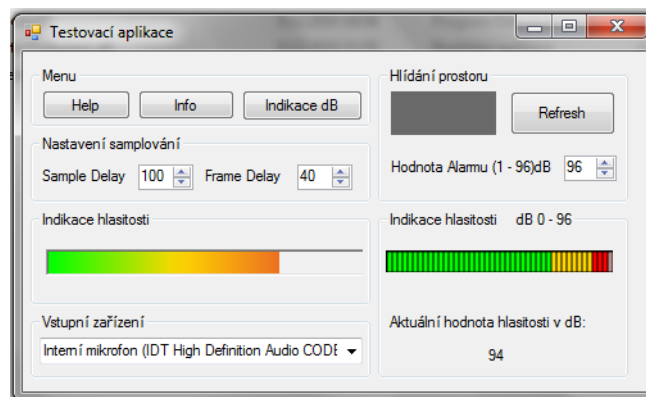


Fig. 2: Sample test programs.

Testing was conducted on the volume of the speech. The left indicator shows volume level and creates an artificial zero noise level. Indications in decibels on the right side indicates all data entering into the microphone, thus indicates a considerable noise. Influence on zero level can be achieved only by increasing the size of the input attenuation or volume. Moreover, the values are converted to a logarithmic scale. An interesting finding is that indicator responded only to certain frequency spectrum. The reason is the nonlinearity of the microphone in the audio frequency spectrum. Increased frequencies are just around 1kHz, which corresponds to the frequency of human voice. Other frequencies are damped, therefore, the indicator cannot respond equally to all levels of sound frequency as well.

## 6. Conclusions

Creating custom Indication component of LED excitation of volume is not a completely stand-alone component on the volume excitation. Therefore, it is always necessary to use another library for input data from the PC sound input. However, non-using components for access to the hardware, component with an even broader application in other applications was created. Since this is only a display bar, the component can be used to indicate other variables than just the inputs from the sound card. By the Indications component of arousal we managed to encompass access to the sound card using DirectSound and DirectX libraries. When seeking for information on the practical component of the LED display, we drew from similar projects from the pages of codeproject. In the second part of the project, we drew from the Microsoft

MSDN library database DirectX and DirectSound, see reference [11], and sites dealing with this topic, see reference [10]. Utilization of this project will certainly be found in its use to create applications using the indicator bar, or for any work involving the use of audio input sound cards. The components were developed in Microsoft Visual Studio 2008 and Visual C#. NET Framework 3.5.

## 7. Acknowledgements

This work was supported in part by (1) „Centre for Applied Cybernetics“, Ministry of Education of the Czech Republic under project 1M0567, (2) „SMEW – Smart Environments at Workplaces“, Grant Agency of the Czech Republic, GACR P403/10/1310, (3) „SCADA system for control and monitoring of processes in Real Time“, Technology Agency of the Czech Republic, TACR, TA01010632 and (4) "User Adaptive Systems", VSB - Technical University of Ostrava under project SP/2011/22. We also acknowledge a support of Vitezslav Hanak in developing of test application and Lukas Cernohorsky in consultation support..

## 8. References

- [1] H. Schmid, Audio Program Level, the Meter, and the Peak-Program Meter, The VU meter, In Journal IEEE Transactions on Broadcasting, Volume: BC-23, Issue: 1, pp. 22-26, ISSN: 0018-9316, March 1977, doi: 10.1109/TBC.1977.266233
- [2] O. Krejcar, Problem Solving of Low Data Throughput on Mobile Devices by Artefacts Prebuffering. *EURASIP Journal on Wireless Communications and Networking*, 2009, Article ID 802523, 8 pages. Hindawi publishing corp., New York, USA, DOI 10.1155/2009/802523
- [3] O. Krejcar, R. Frischer, Non Destructive Defects Detection by Performance Spectral Density Analysis, *Journal Sensors*, MDPI Basel, Vol. 11, No. 3., pp. 2334-2346. (2011)
- [4] O. Krejcar, R. Frischer, Detection of Internal Defects of Material on the Basis of Performance Spectral Density Analysis, *Journal of Vibroengineering*, 2010 – Vol. 12, No. 4 - P. 541-551.
- [5] O. Krejcar, D. Janekulik, L. Motalova, Complex Biomedical System with Biotelemetric Monitoring of Life Functions. In Proceedings of the IEEE Eurocon 2009, May 18-23, 2009, St. Petersburg, Russia. pp. 138-141. DOI 10.1109/EURCON.2009.5167618
- [6] P. Tucnik, Optimization of Automated Trading System's Interaction with Market Environment, 9th International Conference on Business Informatics Research, Univ. Rostock, Rostock, Germany, Lecture Notes in Business Information Processing, 2010, Vol. 64, pp. 55-61
- [7] P. Mikulecky, Remarks on Ubiquitous Intelligent Supportive Spaces, 15th American Conference on Applied Mathematics/International Conference on Computational and Information Science, Univ Houston, Houston, TX, 2009, pp. 523-528, ISBN: 978-960-474-071-0
- [8] M. Augustynek, M. Penhaker, D. Korpas, Controlling Peacemakers by Accelerometers. In 2010 The 2nd International Conference on Telecom Technology and Applications, ICTTA 2010. March 19-21, 2010, Bali Island, Indonesia, Volume2, 2010, p. 161–163. ISBN 978-0-7695-3982-9, DOI: 10.1109/ICCEA.2010.288
- [9] J. Pindor, M. Penhaker, M. Augustynek, D. Korpas, Detection of ECG Significant Waves for Biventricular Pacing Treatment. In 2010 The 2nd International Conference on Telecom Technology and Applications, ICTTA 2010. March 19-21, 2010, Bali Island, Indonesia, Volume 2, 2010, p. 164–167. DOI: 10.1109/ICCEA.2010.186
- [10] Z. Labza, M. Penhaker, M. Augustynek, D. Korpas, Verification of Set Up Dual-Chamber Pacemaker Electrical Parameters. In 2010 The 2nd International Conference on Telecom Technology and Applications, ICTTA 2010. March 19-21, 2010, Bali Island, Indonesia, Volume 2, 2010, p. 168–172. DOI: 10.1109/ICCEA.2010.187
- [11] J. Krawiec, M. Penhaker, O. Krejcar, V. Novak, R. Bridzik, Web System for Electrophysiological Data Management In Proceedings of 2010 Second International Conference on Computer Engineering and Application, ICCEA 2010, 19. – 21. March 2010, Bali Island, Indonesia, Volume 1, 2010 p. 404-407.
- [12] W. L. Hetrich, The ACCU-PEAK TM Level Indicator, The measurement of audio volume, In Journal IEEE Transactions on Broadcasting, Volume: BC-22, Issue: 3, pp. 101-105, Sept. 1976, doi: 10.1109/TBC.1976.266212
- [13] P. Brida, J. Machaj, J. Duha, A Novel Optimizing Algorithm for DV based Positioning Methods in ad hoc Networks, *Electronics and Electrical Engineering*. 2010. – No. 1(97). – P. 33–38.