

Application for Conversion of Audio Signal to MP3

Ondrej Krejcar^{1,2}, Lukas Cernohorsky¹

¹ 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, lukas.cernohorsky@vsb.cz

² Department of Information Technologies, Faculty of Informatics and Management, University of Hradec Kralove, Hradec Kralove, Czech Republic, ondrej.krejcar@asjournal.eu

Abstract. The project is to create an application that converts the audio signal from your computer microphone to record format MP3. Application is designed to record and preserve the audio recordings to a file in a format of lossy compression of audio file compression based on MPEG algorithm (Motion Picture Experts Group). While maintaining a relatively high quality it can reduce the size of audio files in CD quality at about one tenth, for speech, however, it gives much worse results. The advantage of this application is the ability to set frequency range (the sampling) and set the width of the stream, which indicates the number of bits, by which a second of record is registered with. The application is written in C # using Microsoft Visual Studio 2008.

Keywords: MP3, MPEG, frequency, bitrate, Joint - Stereo

1. Introduction

Capturing the sound is mostly made by condenser microphones, which operate by using electric voltage change induced by the change of capacity. The conductive membrane is deformed due to noise, which causes a capacity change of microphone. This creates an electrical signal, which is composed of pure sine waves of different frequencies and amplitudes. The computer processes sound digitally because its analog recording is not, by definition of computing techniques, possible. This means that it is necessary to convert sound waves into a sequence of numbers describing the instantaneous value of amplitude, or a derivative of the signal characteristics. Under the so-called Shannon's sampling theorem, any periodic signal can be digitized without loss of information if the sampling frequency is at least twice as big than the highest frequency contained in the signal. For faithful sound reproduction sampling frequency of around 40kHz would therefore be sufficient. For lower quality and memory efficient record sound cards are able to work with fractions of the fundamental frequency, especially with the half (22 050 kHz) and a quarter (11 025 kHz). Quality of digital recording with a sampling frequency determine the accuracy of recording of the analog signal. In digital audio recording it is the most efficient to talk about accuracy as about the number of bits at the output of the A / D converter. Number of levels of the sample is called the resolution. For 8-bit recording we distinguish 256 values, for the 16-bit 65536 values [1].

MP3 is an audio signal notation. Its size depends on the length and breadth of recording bitrate (bitrate). The higher the bitrate the better quality but less compression and larger file. Today, bitrate in the range of 128-192 kbps (kilo bits per second) is most used. From 160 kbps record is almost indistinguishable from the original. Uncompressed formats (such as standard audio CDs) are impractical for computer processing, as each minute of CD-quality recording takes up approximately 10 MB (megabytes). Transfer to MP3 can reduce this size (with a minimal loss of quality) about ten times. One minute the signal is approximately 1 MB (at 128 kbps bitrate) [2] [3]. MP3 encoding algorithm is so efficient with a few basic procedures. Most important, however, still change of the record to model of human hearing sensitivity (Perceptual Model) is.

Encoder removes from the original signal frequencies, which are assessed as unnecessary by the model - normal ear does not notice that something is missing. The mathematical model simulates the sensitivity of human hearing, it has its weaknesses and imperfections defined, so you can determine which part can be removed. Another procedure is called the Joint Stereo. Both channels of stereo record are often the same signal, Joint Stereo reduces this excess information. Some encoders support the "variable bit rate", where the quiet passages of music are recorded in low bitrate, while the dynamic contrast in high bit rate [4].

The application is written in the programming language C # in development environment from Microsoft. This environment can be used to develop console applications and GUI applications along with Windows Forms applications, web applications, sites and services as native code and managed code in Microsoft Windows platforms, Windows Mobile, Windows CE. NET, .NET Compact Framework and Microsoft Silverlight [5].

2. New Solution

Therefore, to be able to test our created MP3 - if they are working, the application was extended to mp3 player. So the overall concept was created as follows: After starting the application we can set the parameters of our Mp3. Firstly, you choose the device that will retrieve the audio signal. We use the default Windows recording device. The next step is to select the sampling frequency and the width of the mp3 stream (bitrate). You can also select or create a new MP3 file in which to save the replay.

Application captures sound using the MDX (Managed DirectX 1.1) from the sound card in raw PCM format. PCM format is simple and sound sequence of sample values. The samples CAN BE 8bit (0 .. 255) or 16 bit (-32768 .. 32767) each. Samples can be 8-bit (0 .. 255) or 16-bit (-32768 .. 32767). Stereo sound is a sequence of pairs of samples (left, right, left, right ...). Stereo sound is a sequence of pairs of samples (left, right, left, right ...). PCM is a Proper Format for streaming raw data. PCM is the correct format for transmission of raw data. Raw PCM is a type of basic production Mp3SoundCapture components. If you direct this stream format it and WAV file, You Will Not Be Able to Play WAV file, and it must additionally CONTAIN header.WAV and the RIFF (RIFF) format Requires the raw PCM data: to be prefixed with a RIFF header, apart from Which format ook information Contains information about the total length of the PCM data in the file. The third type of component output is Mp3SoundCapture MP3.Typ Mp3SoundCapture output components can be MP3. A Bit rate (kbit / s) and the parameter which decides MP3 sound quality. Baud rate (kbit / s) is the parameter that determines the quality of MP3 sound. Apart from that, the sound quality depends on the format of PCM data being compressed. Not every combination of bit rate and sampling frequency of parameters is allowed.

In our application, you must add a reference: Itrib.Sound.Mp3.dll lame_enc.dll. Secondly mentioned is the king of the coders which create the MP3 format, it is called LAME. Fraunhofer original overtakes not only in quality but also in configurability - a music lover can perfectly suit it to his needs. Variable bitrate (bitrate), Joint Stereo and sophisticated profiles take care of corresponding ratio between quality and compression rate. Against fraunhofer coder behind LAME a whole community of developers stands, which continuously improves it and gives almost regularly every few weeks a new version that is based on optimized psychoacoustic model. Really good version of LAME is 3.91. It allows encoding using profiles (Presets), which use the encoder's skills and are a relief mainly for variable bitrate. Through it, user can e.g. by "alt-preset standard" setting create MP3 file with high quality, but this file is not much bigger than the same recording encoded at 128 kbps. This method saves the encoder bitrate to quiet places and adds it in complex passages. But it leaves the LAME MP3 encoder, this means that at low bit rates like 64 kbps it mercilessly cuts off high frequencies and distorts the sound. But it collects points in faithfulness to the original at 128 Kbps. There, the listener must be all ears to differentiate between the original and the output of LAME [12], [13].

3. Solution Implementation

As we already mentioned, the application was written in C # using Microsoft Visual Studio 2008, connected to .NET Framework 3.5. When the project was started Windows Forms Application was used. We used present form as a skeleton for the interface between the user and program.

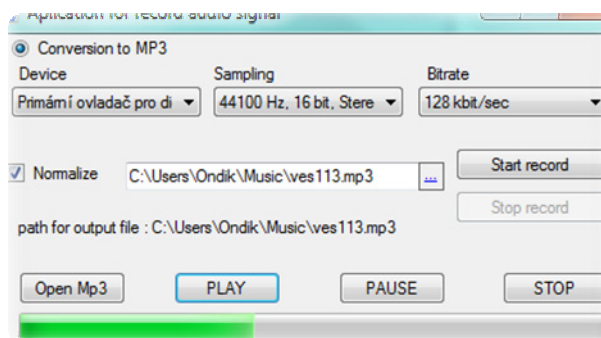


Fig. 1: Sample test programs.

For a description of the interface elements we use italics for the names of the components in the program and ["text"] in square brackets as names for users. To create the interface of the toolbox 3 times ComboBox was inserted, see the description further. DevicesCmb ["Select Device"] to select the device that will "draw" the audio signal. FormatsCmb ["Sampling frequency"] to set the sampling frequency. BitRateCmb ["width stream MP3"] to set the width of the stream (bitrate). Another component is the checkbox. After naming this (normalizeChk) ["Normalization"], the function to maintain constant volume is accessible.

The application was created by the MP3 player whose components can be described subsequently. Button1 ["Open MP3"] opens a MP3 we want to play. Button3 ["PLAY"] starts the selected MP3. Button4 [PAUSE], gives us the chance to pause the song at any time during the recording and using Button3 or Button4 we can start it again from this point. Then there is Button2 [STOP], by this one song is completely stopped. The last component is called the ProgressBar called ProgressBar1. This component takes care of displaying the length of MP3 playback. The main component for the work with processing the record and storing into MP3 format is taken care of by Mp3Sound Capture component.

Here we see a demonstration of working with Capture Mp3Sound when after the start of button StartBtn ["Start Recording"], we select the device with which we will work, there a normalization takes place, whether the CheckBox is marked. The output is set as MP3 with selected bitrate and is written into the file *.mp3. After pressing ["Open MP3"] button we open dialogue, and if we succeed, we open directly the file with the music. We filter only files of type *.Mp3. Using variable zrovnaNecoHraje we retain information about whether a song is being played, so that we could possibly start a new song immediately. If it is something already loaded in mojeHudba, we must first release an old song from memory. We stop the music - mojeHudba.Stop ... mojeHudba.Dispose will eventually release the loaded song from memory. mojeHudba = null initiates metamorphosis. After loading a new file, we set the overall length of ProgressBar and reset it right away. If something has already played before loading, we immediately run it.

4. Testing of Developed Application

The aim of this chapter was to test the solution of our application and describe the findings. To test remove of the audio signal internal microphone of laptop HP EliteBook 4515s was used, the driver IDT High Definition Audio CODEC. Sound recording was reading of the spoken word, always the same article of approximately 30 seconds. Subjective quality of the resulting record was particularly affected by the switching off the standardization and that in a negative sense. There was a significant decrease in volume. Normalization method is very suitable for the unification of the constant volume of records.

Another parameter that we changed was the frequency range. The quality of recording by this step did not change, you must note, however, we do not use high quality equipment to record sound, so we can not expect major differences in this way in each song. Sample Rate (sampling frequency) – gives you chance to change the sampling frequency of audio data to match the quality of data formed by the encoder. If we want to maintain the quality of the upcoming CD, it is necessary to maintain the 44.1 kHz sampling. By reducing the sampling rate a smaller file is obtained, but also lower quality output. Some encoders for low bit rate automatically assign a lower sampling frequency.

Tab. 1: Sample test summary for various recording duration.

record's duration (s)	normalize	sampling (kHz)	bitrate (kbit)	file's size (kB)
32	Yes	44.1	128	501
31	No	44.1	128	500
34	Yes	22.05	128	540
30	Yes	22.05	32	124

The bitrate parameter has already been in quite a large difference, in the size of the resulting record file. It was a record of relatively equal length at Sample Rate = 20.05kHz. The record for the bitrate of 128kbit was almost 3 times greater than for the bitrate of 32kbit. Bitrate is a fundamental parameter affecting the final sound quality of MP3 files. Generally, the higher the data rate is, the higher the resulting file quality is, but also larger. Generally accepted value considered to be one with a quality approaching CD quality is the data rate of 128 kbps. This value corresponds to the compression ratio of 1:12 and the most of files, we are encountered with e.g. at Internet, is compressed this way. Changing this value depends on the individual needs of everyone who deals with creating and listening. Mp3. We use higher values (160, 192 kbps) if the distortion is at 128 kbps, and listeners are still discomforted. Typical examples are recordings of classical music. At 256 kbps, noone knows the difference from the original recording, but the compression ratio is not so convenient. If it is a lower quality recording, or less demanding (spoken word), we can use a lower bitrate and save disk space. The encoders support data flow in different ranges (8-256, respectively up to 320 kbps). The total value of the bitrate is equal to the sum of bitrates of all channels [14], [15].

Tab. 2: Summary of compression rates for various sources.

qualities	band' s width (kHz)	sample rate (Hz)	modulation	bitrate (kbit)	compression rate / CD)
telephone	2.5	11.025	mono	8	96 : 1
FM radio	11	22.025	stereo	56 - 64	26 -24 : 1
CD	16	44.100	stereo	112 - 128	14 -12 : 1

MP3 player was tested not only for playing records produced, but also to the already existing MP3 file. Specifically, it was a song and then 128kbit subjective comparison with your MP3 player Winamp 5581. Practically, we can say that the players brought an equivalent performance in the recitation of same songs. So we hereby verify the quality of library Microsoft.DirectX.AudioVideoPlayback we used in the application. This library is intended for basic playback of audio and video formats and also adds the basic control elements [16]. In this section we even compare the MP3 compression methods. Psychoacoustic model: simulating the human ear and mathematically defines its ability to perception (eg, sounds above 16 kHz are for humans with advancing age only slightly or not at all audible). Furthermore, the ear of two consecutive notes with a similar timbre in detail only the louder one is heard. Psychoacoustic model knows these human weaknesses, it defines them mathematically and summarizes the results in tables. When comparing encoders compare output signal with table and cut the matching frequencies. The technical challenge for coders is to better adapt psychoacoustic model to human hearing. Joint Stereo: it uses the fact that on both channels often is the same signal.

Compressed file: The right channel contains the sum of the two channels, left one is the difference. The decoder reproduces later the signal correctly decoded. The combination of bits: When playing an audio signal by CD player, digital information of a song is recorded in so-called samples. CD player with a sampling frequency (sampling rate) 44.1 kHz samples CDs 44,100 times per second. It brings together 1152 samples per frame. Each sample consists of a number of bits in the case of a 16-bit encryption, so the 16 bits. If the bit pattern starts at 000,000, the six zeros can be omitted. With the header information decoder knows during the playback that from 10-bit signal the 16-bit signal must be made by adding the missing bits. Space saving - cropping of high tones: At high frequencies, which the human ear can barely perceive, coders cut all information containing the original WAV.

5. Conclusions

In this project, we first tried working with the programming language C # in development environment Microsoft Visual Studio 2008. The first idea was to only create an application that creates the audio to MP3, but regarding the need to test the application, the application was expanded to MP3 player. Of course, it is possible to use external programs, but it would be inconsistent with our initial specification (speed, simplicity ...). Essential in this project is to use the lame encoder. Its main advantage (free use), makes it a powerful tool, because it leads to its constant improvement by the numerous amount of developers. Applications have been tested. We present the results in chapter IV. The possibility of usage is mainly for private use in recording the audio signal processing and its preservation. Thanks to compression it does not take a large amount of space on our HDD.

6. 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 Ondrej Vesely in developing of test application and Dalibor Janckulik in consultation support.

7. References

- [1] Ponce, B.; , "The impact of MP3 and the future of digital entertainment products ," Communications Magazine, IEEE , vol.37, no.9,pp.68-70,Sep1999 doi: 10.1109/35.790866
- [2] 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
- [3] 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
- [4] 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.
- [5] 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)
- [6] O. Krejcar, D. Janckulik, 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
- [7] 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
- [8] 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.
- [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] LIU, Qingzhong; SUNG, Andrew; QIAO, Mengyu. Detection of Double MP3 Compression. Cognitive Computation. 2010-05-22, 1866-9956, s. 1-6. from WWW: <<http://dx.doi.org/10.1007/s12559-010-9045-4>>.