KUMARAGURU
COLLEGE OF TECHNOLOGY
Department of Computer Science and Engineering

ISO 9001:2000 Certified

DIGITAL AUDIO ANALYZER

PROJECT WORK DONE AT
PELLUCID SYSTEMS
CHENNAI

P-1245

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF
M.Sc APPLIED SCIENCE (SOFTWARE ENGINEERING)
OF BHARATHIAR UNIVERSITY, COIMBATORE.

SUBMITTED BY

S. RAMKUMAR
REG NO: 0137S0049

Under the guidance of

INTERNAL GUIDE

Mr. M. Manikantan, M.C.A.,
Dept. of Computer Science & Engineering,
Kumaraguru College of Technology,
Coimbatore.

EXTERNAL GUIDE

Mr. Shankar Raman, M.C.A.,
Senior Software Engineer,
Pellucid Systems,
Chennai.

CERTIFICATE

KUMARAGURU
COLLEGE OF TECHNOLOGY

Department of Computer Science and Engineering

ISO 9001:2000 Certified

(Affiliated to Bharathiyar University)
Coimbatore - 641006
(JUNE-2004 TO OCTOBER –2004)

CERTIFICATE

This is to certify that the project entitled

"DIGITAL AUDIO ANALYZER"

Done by

S.RAMKUMAR
REG NO: 0137S0049

Submitted in partial fulfillment of the requirements for the award of the degree of M. Sc (Applied Science) Software Engineering of Bharathiyar University.

Professor and HOD **Internal Guide**

Submitted to University examination held on 30.09.04

Internal Examiner **External Examiner**

COMPANY CERTIFICATE

Pellucid S...
Technologies R...

/09/2004

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr. RAMKUMAR.S (RegNo.0137S0049) undergoing 4th year Sc. (Software Engineering) in Kumaraguru College of Technology, Coimbatore has cessfully completed the individual project titled

"DIGITAL AUDIO ANALYZER"

has done the project from June 2004 to September 2004. During the period of his project k with us, he was found to be hard working and sincere in his assignments. We wish him he best in his future.

e : Chennai
e : 15/09/2004

Signature of Issuing Authority

DECLARATION

DECLARATION

I here by declare that the project entitled **"DIGITAL AUDIO ANALYZER"**, submitted to **Kumaraguru College of Technology**, Coimbatore Affiliated to Bharathiyar University as the project work of **Master of Science in Applied Science Software Engineering,** is a record of original work done by me under the supervision and guidance of **Mr. Shankar Raman, M.C.A.,** Pellucid Systems., Chennai and **Mr. M.Manikantan. M.C.A.,** Lecturer of Kumaraguru College of Technology, Coimbatore and the project work has not found the basis for the award of any Degree/Diploma/Associate ship/Fellowship or similar title to any candidate of any University.

Place: COIMBATORE

Date: 24.09.2004

Signature of the student

S. Ramkumar

Countersigned by

(Internal Guide)

(External Guide)

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I am immensely grateful to **Dr.K.K.Padmanabhan, B.Sc (Engg), M.Tech., Ph.D.,** Principal, Kumaraguru College of Technology for his valuable support to come out with this project.

I really feel delighted in expressing my heartful thanks to **Prof. Dr. S. Thangasamy, BE. (Hon's)., Ph.D.,** Head of the Department of Computer Science and Engineering for his endless encouragement in carrying out this project successfully.

My heartfelt thanks to our Course Coordinator **Mr. K.R. Baskaran, B.E., M.S.,** for his unfailing enthusiasm, encouragement and guidance that paved me to the completion of this project.

I am indent to express my heartiest thanks to **Mr. M.Manikantan M.C.A.,** Department of Computer Science and Engineering, my project guide who rendered his valuable guidance and support to do this project work extremely well.

I am greatly indebted to Senior Software Engineer, **PELLUCID SYSTEMS,** Chennai for getting me into his esteemed Institution. I also thank **Mr. Shankar Raman, M.C.A.,** my guide and he has helped me a lot in my project.

I am also thankful to all the faculty members of the Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore for their valuable guidance, support and encouragement during the course of my project work.

My humble gratitude and thanks to my parents who have supported to complete the project and to my friends for lending me valuable tips, support and cooperation through out my project work.

SYNOPSIS

SYNOPSIS

The project entitled **"Digital Audio Analyzer"** is a real-time project developed for Pellucid Systems, Chennai. Only registered users can use this software. Any new users can signup at the time of product Installation and start using the software.

This project attempts to cover the different facets of sound processing by including different utilities that make use of a certain feature of sound processing technique. This software analyzes sounds by plotting the frequencies, enlarging the visual area of the plotted frequency, saving a portion of a selected audio file, playing a selected portion of a wave file, comparing audio files, simultaneously playing several MP3 format files, converting audio pitches, audio mixing, text to speech conversion etc. The software has been separated into different modules, each one performing a significant work. The modules are:

1) Audio Analysis Module
2) Audio Training Module
3) Synchronized Sounds Module
4) Voice Converter Module
5) Audio Mixer Module
6) Text to Speech Module
7) Utilities Module
8) Help Module

These modules cover the whole gamut of my understanding of audio processing. These modules make heavy use of advanced APIs and DLLs. The Application Programming Interface (API) is an interface to the many routines made available through Dynamic Link Libraries (DLLs). Windows comes with hundreds of DLLs. These are precompiled programming libraries that hold many, often related, programming tools. Each of these DLLs contains numerous functions and procedures. Most of these functions are used by Windows and reused by other applications.

The use of APIs resulted in

1) Significant additional functionality
2) Performance gains
3) Low overhead
4) Reduced file size
5) High power audio impact.

CONTENTS

INTRODUCTION

1.0 INTRODUCTION:

Since this project is entirely dealing with digital sound processing, a deep understanding of the different aspects of sound is absolutely necessary to evaluate the worthiness of this software. With this in mind, I have given a basic explanation of the terminology used in any sound processing software.

Sound:

It is a physical phenomenon that stimulates the sense of hearing. In humans, hearing takes place whenever vibrations of frequencies between about 15 and 20,000 hertz reach the inner ear. The hertz, or Hz, is a unit of frequency equaling one cycle per second. Such vibrations reach the inner ear when they are transmitted through air, and the term sound is something restricted to such airborne vibrational waves. Modern physicists, however, usually extend the term to include similar vibrations in liquid or solid media. Sounds of frequencies higher than about 20,000 Hz are called ultrasonic.

In general, waves can be propagated transversely or longitudinally. In both cases, only the energy of wave motion is propagated through the medium; no portion of the medium itself actually moves very far. As a simple example, a rope may be tied securely to a post at one end, and the other end pulled almost taut and then shaken once. A wave will travel down the rope to the post, and at that point it will be reflected and returned to the hand. No part of the rope actually moves longitudinally toward the post, but each successive portion of the rope moves transversely. This type of wave motion is called a transverse wave. Similarly, if a rock is thrown into a pool of water, a series of transverse waves moves out from the point of impact. A cork floating near the point of impact will bob up and down, that is, move transversely with respect to the direction of wave motion, but will show little if any outward or longitudinal, motion. As the energy of wave motion is propagated outward from the center of disturbance, the individual air molecules that carry the sound move back and forth, parallel to the direction of wave motion. Thus, a sound wave is a series of alternate compressions and rarefactions of the air. Each individual molecule passes the energy on to neighboring molecules, but after the sound wave has passed, each molecule remains in about the same location.

PHYSICAL CHARACTERISTICS:

Any simple sound, such as a musical note, may be completely described by specifying three perceptual characteristics:

1) Pitch,
2) Loudness (or intensity), and
3) Quality (or timbre).

These characteristics correspond exactly to three physical characteristics: frequency, amplitude, and harmonic constitution, or waveform, respectively. Noise is a complex sound, a mixture of many different frequencies or notes not harmonically related.

Frequency:

Sounds can be produced at a desired frequency by different methods. For example, a sound of 440 Hz can be created by actuating a loudspeaker with an oscillator tuned to this frequency. An air blast can be interrupted by a toothed wheel with 44 teeth, rotating at 10 revolutions/sec; this method is used in operating an ordinary siren. The sound of the speaker and that of the siren at the same frequency are very different in quality, but will correspond closely in pitch, equivalent to the A above middle C on a piano. The next higher A on the piano, the note one octave above, has a frequency of 880 Hz. Similarly, notes one or two octaves below have frequencies of 220 or 110 Hz, respectively. Thus, by definition, an octave is the interval between any two notes the frequencies of which are in a two-to-one ratio.

A fundamental law of harmony states that two notes an octave apart, when sounded together, produce a euphonious combination. A fifth and a major third produce successively less euphonious combinations. Physically, an interval of a fifth consists of two notes, the frequencies of which bear the arithmetical ratio three to two, and a major third, the ratio five to four. Fundamentally, then, the law of harmony states that two or more notes sound euphonious when played together if their frequencies bear the ratio of small, whole numbers; if the frequencies do not bear such ratios, a dissonance is produced. On a fixed-pitch instrument, such as a piano, it is not possible to arrange the notes so that all of these ratios hold exactly, and some compromise is necessary in tuning, called the meantone system, or tempered scale.

Amplitude:

The amplitude of a sound wave is the degree of motion of air molecules within the wave, which corresponds to the extent of rarefaction and compression that accompanies the wave. The greater the amplitude of the wave, the harder the molecules strikes the ear drum and the louder the sound that is perceived. The amplitude of a sound wave can be expressed in terms of absolute units by measuring the actual distance of displacement of the air molecules, or the pressure differential in the compression and rarefaction, or the energy involved. Ordinary speech, for example, produces sound energy at the rate of about one hundred-thousandth of a watt. All of these measurements are extremely difficult to make, however, and the intensity of sounds is generally expressed by comparing them to a standard sound, measured in decibels.

Intensity:

The distance at which a sound can be heard depends on its intensity, which is the average rate of flow of energy per unit area perpendicular to the direction of propagation. In the case of spherical waves spreading from a point source, the intensity varies inversely as the square of the distance, provided that no loss of energy is due to viscosity, heat conduction, or other absorption effects. Thus, in a perfectly homogeneous medium, a sound will be nine times as intense at a distance of 1 unit from its origin as at a distance of 3 units; that is, intensity varies inversely as the square of the distance. In the actual propagation of sound through the atmosphere, changes in the physical properties of the air, such as temperature, pressure, and humidity, produce damping and scattering of the directed sound waves, so that the inverse-square law generally is not applicable in direct measurements of the intensity of sound.

Quality:

If the note a above middle C is played on a violin, a piano, and a tuning fork, all at the same volume, the tones are identical in frequency and amplitude, but very different in quality. Of these three sources, the simplest tone is produced by the tuning fork, the sound in this case consisting almost entirely of vibrations having frequencies of 440 Hz. Because of the acoustical properties of the ear and the resonance properties of the ear's vibrating membrane, however, it is doubtful whether a pure tone reaches the inner hearing mechanism in an unmodified form. The principal component of the note produced by the piano or violin also has a frequency of 440 Hz, but these notes also contain components with frequencies that are exact multiples of 440, called overtones, such as 880, 1320, and 1760. The exact intensity of these other components, which are called harmonics, determines the quality, or timbre, of the note.

Velocity of Sound:

The frequency of a sound wave is a measure of the number of waves passing a given point in 1 second. The distance between two successive crests of the wave is called the wavelength. The product of the wavelength and the frequency must equal the speed of propagation of the wave, and is the same for sounds of all frequencies (if the sound is propagated through the same medium at the same temperature). Thus, the wavelength of A above middle C is about 78.2 cm (about 2.6 ft), and the wavelength of A below middle C is about 156.4 cm (about 5.1 ft).

The speed of propagation of sound in dry air at a temperature of 0° C (32° F) is 331.6 m/sec (1088 ft/sec). If the temperature is increased, the speed of sound increases; thus, at 20° C (68° F), the velocity of sound is 344 m/sec (1129 ft/sec). Changes in pressure at controlled density have virtually no effect on the speed of sound. The velocity of sound in many other gases depends only on their density. If the molecules are heavy, they move less readily, and sound progresses through such a medium more slowly. Thus, sound travels slightly faster in moist air than in dry air, because moist air contains a greater number of lighter molecules. The velocity of sound in most gases depends also on one other factor, the specific heat, which affects the propagation of sound waves. See Temperature.

Sound generally moves much faster in liquids and solids than in gases. In both liquids and solids, density has the same effect as in gases; that is, velocity varies inversely as the square root of the density. The velocity also varies directly as the square root of the elasticity. The speed of sound in water, for example, is slightly less than 1525 m/sec (5000 ft/sec) at ordinary temperatures but increases greatly with an increase in temperature. The speed of sound in copper is about 3353 m/sec (about 11,000 ft/sec) at ordinary temperatures and decreases as the temperature is increased (due to decreasing elasticity); in steel, which is more elastic, sound moves at a speed of about 4877 m/sec (about 16,000 ft/sec). Sound is propagated very efficiently in steel.

Refraction, Reflection, and Interference:

Sound moves forward in a straight line when traveling through a medium having uniform density. Like light, however, sound is subject to refraction, which bends sound waves from their original path. In Polar Regions, for example, where air close to the ground is colder than air that is somewhat higher, a rising sound wave entering the warmer region, in which sound moves with greater speed, is bent downward by refraction. The excellent reception of sound downwind and the poor reception upwind are also due to refraction. The velocity of wind is generally greater at an altitude of many meters than near the ground; a rising sound wave moving downwind is bent back toward the ground, whereas a similar sound wave moving upwind is bent upward over the head of the hearer.

Sound is also governed by reflection, obeying the fundamental law that the angle of incidence equals the angle of reflection. An echo is the result of reflection of sound. Sonar depends on the reflection of sounds propagated in water. A megaphone is a funnel-like tube that forms a beam of sound waves by reflecting some of the diverging rays from the sides of the tube. A similar tube can gather sound waves if the large end is pointed at the source of the sound; an ear trumpet is such a device.

Sound is also subject to diffraction and interference. If sound from a single source reaches a listener by two different paths-one direct and the other reflected-the two sounds may reinforce one another; but if they are out of phase they may interfere, so that the resultant sound is actually less intense than the direct sound without reflection. Interference paths are different for sounds of different frequencies, so that interference produces distortion in complex sounds. Two sounds of different frequencies may combine to produce a third sound, the frequency of which is equal to the sum or difference of the original two frequencies.

Sensations of the Tone:

If the ear of an average young person is tested by an audiometer, it will be found to be sensitive to all sounds from 15 to 20 Hz to 15,000 or 20,000 Hz. The hearing of older persons is less acute, particularly to the higher frequencies. The ear is most sensitive in the range from A above middle C up to A four octaves higher; in this range a sound can be perceived hundreds of times fainter than a sound an octave higher or two octaves lower. The degree to which a sensitive ear can distinguish between two pure notes of slightly different loudness or slightly different frequency varies in different ranges of loudness and frequency of the tones. A difference in loudness of about 20 percent (1 decibel, dB), and a difference in frequency of 1/3 percent (about 1/20 of a note) can be distinguished in sounds of moderate intensity at the frequencies to which the ear is most sensitive (about 1000 to 2000 Hz). In this same range, the difference between the softest sound that can be heard and the loudest sound that can be distinguished as sound (louder sounds are "felt," or perceived, as painful stimuli) is about 120 dB (about 1 trillion times as loud).

All of these sensitivity tests refer to pure tones, such as those produced by an electronic oscillator. Even for such pure tones the ear is imperfect. Notes of identical frequency but differing greatly in intensity may seem to differ slightly in pitch. More important is the difference in apparent relative intensities with different frequencies. At high intensities the ear is approximately equally sensitive to most frequencies, but at low intensities the ear is much more sensitive to the middle high frequencies than to the lowest and highest. Thus, sound-reproducing equipment that is functioning perfectly will seem to fail to reproduce the lowest and highest notes if the volume is decreased.

Three Important Types of Ordinary Sound

In speech, music, and noise, pure tones are seldom heard. A musical note contains, in addition to a fundamental frequency, higher tones that are harmonics of the fundamental frequency. Speech contains a complex mixture of sounds, some (but not all) of which are in harmonic relation to one another. Noise consists of a mixture of many different frequencies within a certain range; it is thus comparable to white light, which consists of a mixture of light of all different colors. Different noises are distinguished by different distributions of energy in the various frequency ranges.

When a musical tone containing some harmonics of a fundamental tone, but missing other harmonics or the fundamental itself, is transmitted to the ear, the ear forms various beats in the form of sum and difference frequencies, thus producing the missing harmonics or the fundamental not present in the original sound. These notes are also harmonics of the original fundamental note. This incorrect response of the ear may be valuable. Sound-reproducing equipment without a large speaker, for example, cannot generally produce sounds of pitch lower than two octaves below middle C; nonetheless, a human ear listening to such equipment can re-supply the fundamental note by resolving beat frequencies from its harmonics. Another imperfection of the ear in the presence of ordinary sounds is the inability to hear high-frequency notes when low-frequency sound of considerable intensity is present. This phenomenon is called masking.

In general, speech is understandable and musical themes can be satisfactorily understood if only the frequencies between 250 and 3000 Hz, the frequency range of ordinary telephones, are reproduced, although a few speech sounds, such as ***th***, require frequencies as high as 6000 Hz. For naturalness, however, the range of about 100 to 10,000 Hz must be reproduced. Sounds produced by a few musical instruments can be reproduced naturally only at somewhat lower frequencies, and a few noises can be reproduced at somewhat higher frequencies.

PHONETICS

		BI-LABIAL (both lips)	LABIODENTAL (lips and teeth)	DENTAL (teeth)	ALVEOLAR ("gum ridge")	PALATOALVEOLAR (ridge and palate)	PALATAL (hard palate)	VELAR (soft palate)	GLOTTAL (throat)
PLOSIVE (air stops)	tense	p sip p^h pen			t step t^h tip			k sick k^h kiss	
	lax	b bit			d dip			g give	
NASAL (air resonates)		m map			n nip			[ŋ] king	
LATERAL (tongue narrows)					l lip				
FLAP (tongue flaps)					[ɾ] utter				
FRICATIVE (turbulence)	tense		f fat	[θ] thin	s sip	[ʃ] mission			h hit
	lax		v vat	[ð] this	z zip	[ʒ] vision			
SEMI-VOWELS		w wit			[ɹ] red		j yet		
OFF-GLIDES		ᵘ(raised) out					ⁱ(raised) bite		

In order to vocalize a consonant, a speaker must constrict his or her air flow at some point in the delivery, using either the tongue, mouth, or throat. This table, showing American English consonants, indicates the part of the mouth or throat in which each sound is formed.

Phonetics is a branch of linguistics concerned with the production, physical nature, and perception of speech sounds. The main fields of study are experimental phonetics, articulatory phonetics, phonemics, acoustical phonetics, and auditory phonetics. Auditory phonetics is the field involved in determining how speech sounds are perceived by the human ear.

EXPERIMENTAL PHONETICS

This is the physical science that collects measurable data about the articulatory, acoustic, and auditory properties of vocal sounds, using instruments such as the kymograph, which traces curves of pressure, and the X-ray. The amount of detail in the measurement of vocal sounds is limited only by the precision of the instrument. Differences are found in every vocal sound.

ARTICULATORY PHONETICS

This describes speech sounds genetically—that is, with respect to the ways by which the vocal organs modify the air stream in the mouth, nose, and throat in order to produce a sound. All the vocal activities involved in a sound need not be described, but only a selection of them, such as the place and manner of articulation. Phonetic symbols and their articulatory definitions are abbreviated descriptions of these selected activities. The symbols most commonly used are those adopted by the International Phonetic Association (IPA) and are written in brackets.

The manner of articulation is determined by the way in which the speaker affects the air stream with the movable organs. This action may consist of stopping the air completely (plosive); leaving the nasal passage open during the stopping (nasal); making contact with the tongue but leaving space on either side of it (lateral); making merely a momentary light contact (flap); leaving just enough space to allow a continuing stream of air to produce friction as it passes through (fricative); or permitting the air stream to pass over the center of the tongue without oral friction (vocal). The speaker produces vowels of different quality by varying the position of his or her tongue on its vertical axis (high, mid, low) and on its horizontal axis (front, central, back). For example, a speaker moves the tongue from low to high in pronouncing the first two vowels of Aïda, and from back to front in pronouncing successively the vowel sounds in *who* and *he.* The tongue positions for the vowels *u, i,* and *a* are the cardinal points on the so-called vowel triangle $^{u}{}_{a}{}^{i}$. The vowel has the most neutral position. The quality of a vowel also depends on whether the speakers keeps the lips rounded or unrounded, keeps the jaws close together or open, or holds the tip of the tongue flat or curled up (retroflex). At the same time the speaker may move the tongue gradually upward and to the front or upward and to the back, making diphthongal off-glides.

Other modifications may also affect the quality of the sounds. For example, nasals rather than vowels may be made the prominent part of the syllable, and certain typical vowel formations, called semivowels, may be nonsyllabic. The quality of certain sounds is also affected by whether the speaker keeps the speech organs tense or lax. The vocal cords are vibrated to produce sounds that are voiced. Vowels are voiced, and in English, lax consonants are more or less voiced. When the speaker gives a strong puff of air after the contact, this is called aspiration. If the hand is placed before the lips, aspiration may be observed in the p^h sound produced at the beginning of the word *pie.* The accompanying charts of the International Phonetic Alphabet, using standard transcriptions in brackets, presents a schematic description of these activities in English, although not all the modifications are included. An accurate phonetic transcription of all would describe even regional accents.

PHONEMICS

This is a study of the sounds of speech in their primary function, which is to make vocal signs that refer to different things sound different. The phonemes of a particular language are those minimal distinct units of sound that can distinguish meaning in that language. In English, the *p* sound is a phoneme because it is the smallest unit of sound that can make a difference of meaning if, for example, it replaces the initial sound of *bill, till,* or *dill,* making the word *pill.* The vowel sound of *pill* is also a phoneme because its distinctness in sound makes *pill,* which means one thing, sound different from *pal,* which means another. Two different sounds, reflecting distinct articulatory activities, may represent two phonemes in one language but only a single phoneme in another. Thus phonetic *r* and *l* are distinct phonemes in English; whereas these sounds represent a single phoneme in Japanese, just as p^h and *p* in *pie* and *spy,* respectively, represent a single phoneme in English although these sounds are phonetically distinct. Phonemes are not letters; they refer to the sound of a spoken utterance.

For example, flocks and phlox have exactly the same five phonemes. Similarly, bill and Bill are identical phonemically, regardless of the difference in meaning. Each language has its own inventory of phonetic differences that it treats as phonemic—that is, as necessary to distinguish meaning. For practical purposes, the total number of phonemes for a language is the least number of different symbols adequate to make an unambiguous graphic representation of its speech that any native could read if given a sound value for each symbol, and that any foreigner could pronounce correctly if given additional rules covering non-distinctive phonetic variations that the native makes automatically. For convenience, each phoneme of language may be given a symbol.

ACOUSTICAL PHONETICS

This is the study of speech waves as the output of a resonator—that is, the vocal tract coupled to other sources. Sound waves are closer than articulations to the essence of communication, because the same auditory impression can be produced by a normal articulation and by an entirely different sound apparatus, like that of parrots. A spectrograph may be used to record significant characteristics of speech waves and to determine the effect of articulatory activities. Parts of this record of speech waves can be cut out experimentally and the rest played back as sound in order to determine which features suffice to identify the sounds of a language.

Theoretically, the spelling of phonemes, the simplest sound elements used to distinguish one word from another, should indicate precisely the sound characteristics of the language. For example, in English, at contains two phonemes, mat three, and mast four. Very frequently, however, the spelling of English words does not conform to the number of phonemes.

Enough, for example, which has four phonemes (enuf), is spelled with six letters, as is breath, which also has four phonemes (breu) and six letters. The main vowel phonemes in English include those represented by the italicized letters in the following words: bit, beat, bet, bate, bat, but, botany, bought, boat, boot, book, and burr. These phonemes are distinguished from one another by the position of articulation in the mouth. Four vowel sounds, or complex nuclei, of English are diphthongs formed by gliding from a low position of articulation to a higher one. These diphthongs are the i of bite (a glide from o of botany to ea of beat), the ou of bout (from o of botany to oo of boot), the oy of boy (from ou of bought to ea of beat), and the u of butte (from ea of beat to oo of boot). The exact starting point and ending point of the glide varies within the English-speaking world. difference achieved by pronouncing one syllable more forcefully than another, for example, the difference between rec‰ ord (noun) and re cord‰ (verb). pitch is, for example, the difference between the pronunciation of John and John?

1.1 PROJECT OVERVIEW

The Digital Audio Analyzer Project has been created with the aim of combining several unique audio features like Audio Analysis, Audio Comparison, Synchronized Sound Player, Mixer Control, Text to Speech, Audio Converter, Audio Reminder, and Audio Calculator in a single package, to correct the defects prevalent in the existing system, to create a user friendly interface, and to create a software audio package that is both affordable in cost and also simple to maintain.

There is provision in the program to plot a wave file by sampling the frequencies of a sound file and plotting the samples in a graph for the purpose of Sound Comparison. A wave file can be played back wholly or by playing back only the selected portion. Converting an audio input of a given frequency and pitch, to that of a desired frequency for sound conversion purposes. A recorded male voice can be made to sound like a female voice, and vice versa. A unique feature to play multiple sound files simultaneously by invoking the MCI (Media Control Interface) commands is present in this project. Optimum usage of the Sound APIs has been made in the Audio Mixer to dynamically adjust the various parameters of the sound card. The user can choose to play music files at a preset time. This feature can be used as an audio reminder cum Calendar. The program also contains another feature that converts text to speech. It is done by using the TTS engine. And finally, this software also contains an audio calculator which also utilizes the TTS engine. These features make this software a unique and highly efficient one in the sound processing field.

1.2 ORGANIZATION PROFILE

Pellucid Systems, Chennai is a reputed company dealing in audio products for the past 7 years. Started in the year 1997 with strength of 15 workers, the company has grown manifold over the years. Now with worker strength of over 120 in the Chennai Head office alone, the company has established many branches all over South India

The company primarily deals in audio electronic items: both hardware and software. It has a vast client base including Digital Sound Studios, Entertainment industry, Television, Advertising, homes, schools, etc. The company has fulfilled the requirement of its client in a professional manner over the years and has earned a very good name. The company has extensive and continually updated expertise in real-time systems, on-line systems, embedded systems, image processing, audio processing, data communications, networking, e-commerce technologies etc.

VISION

The company's vision is to operate globally and bring the benefits of Information Technology to improve the productivity of their customers and quality of their products and services.

MISSION

Pellucid systems will constantly endeavor to delight its customers through excellence in service delivery and achieve worldwide recognition. People are pellucid systems strength, through staff empowerment, customer focus and Quality Management Systems, Pellucid will continuously improve its products and services to meet international standards of quality, cost and time.

SYSTEM STUDY AND ANALYSIS

2.0 SYSTEM STUDY AND ANALYSIS

System study is the process of gathering and interpreting the facts, diagnosing the problems, and using the information to recommend improvements to the system. Basically system study is done to understand the system and its problems thoroughly. The investigation into the system's operations brings out the current methods that the system is following and evaluates the effectiveness by using this information. Finally, systems analysis tells 'what the system should do' to overcome the existing problems.

2.1 SOFTWARE REQUIREMENT SPECIFICATION

The company's requirement specification points out the following features implemented in the new software:

1. The software should be user friendly with a pleasant interface.

2. It should be capable of being operated with the least amount of training.

3. It should be competent to produce professional output.

4. It should contain an authentication and verification module.

5. The software should be capable of opening audio files from the hard disk, floppy drive, and the CD Rom drive.

6. There should be live audio recording facility to record audio matter through microphone and facility to save such recordings.

7. It should contain modules that display the audio files frequencies in a graph.

8. The software should contain facility to magnify a particular portion of an audio file for analysis.

9. There should be a feature for saving only a portion of the audio files.

10. There should be an audio comparison feature implemented.

11. There should be a training module for recording the phonemes, words, and sentences at varying speeds.

12. The required software should contain a module that converts one frequency to another.

13. The software must be capable of playing different MP3 format files simultaneously.

14. The software should contain features that adjust the soundcard capabilities.

15. It should have a Text to Speech facility.

16. There should be facility to play a particular audio file at a preset time.

17. A help module must also be included.

These are some of the specific requirement of the company. The proposed software should include all the above specification made by the company.

2.2 EXISTING SYSTEM

The existing system is riddled with innumerable defects. I will point out the critical ones.

- First and foremost, the software designed and used by the company was created at the time of the company's inception. It now cannot cope with the present day technological requirement. Any idea to upgrade the existing software is unfeasible because of the obsoleteness of the designing platform and the machines.
- Secondly, the resources used by the existing software are power-intensive and they consume huge processor timings.
- Thirdly, only a technically skilled person can use the software. A lay user cannot hope to comprehend and use the software. The company had to spend a lot of time and money to train a worker to use the existing system. The user interface is very complicated in the existing. The first time user has to spend a considerable time and effort to find out about the working of this software.

- Fourthly, the hardware required by the existing system is priced very high and it has also become obsolete. Also, the existing system does not analyze wave sounds properly.
- In the existing system there is no feature for visually plotting and enlarging the visually plotted wave's frequencies.
- There is no facility to copy and paste a particular section of an audio file & play it.
- The existing software lacks in saving only a particular portion of an audio file.
- The existing system cannot play several music files simultaneously.
- The existing system cannot read the text matter typed by the user.
- The mixer control present in the existing does not contain many features required in the audio field.
- The audio comparison feature in the existing system does not produce expected results.
- There is no audio training module in the existing system for analysis of sounds.
- There is no facility to record live audio.
- These and several other unmentioned defects make the existing software an unsuitable one in the present day context.

2.3 PROPOSED SYSTEM

The purpose of this project is to overcome all the defects present in the existing system. These problems and defects have been addressed to in a professional manner. Besides, the proposed system also contains several other new and unique features.

The main advantages of the proposed system are enumerated hereunder:

1) All the features required by the company are bundled in a single package. The IDE obviates the need for switching over from one program to another to use a particular function.

2) The next major advantage is the proven capability of this software for audio comparison.

3) The presence of an audio training module for recording sounds is a feature not implemented in the old system.

4) The presence of a mixer control is an important advantage over the existing system. The user need not go outside this software to adjust sound settings. Once settings are made in this control, they will be in effect till the user changes them next time. The settings persist even after the user closes this application.

5) Text to voice converter is a boon for those people who are visually impaired. This feature can also be used to teach young children to spell and pronounce words correctly.

6) Another special advantage of the proposed system is to play multiple songs simultaneously. This feature can be well used by a disk jockey. The multiple song player can also be used in creating different sound effects.

7) There is panning effect present in the multiple sound player. The user can control the volume of all the simultaneous players individually by using a single slider.

8) Another advantage of this software is the ability to set audio alarm.

9) A user friendly help module clears any doubt arising in the mind of a user.

The aforementioned points answers the question about the advantages of the implementing the proposed system. The proposed system has been divided into the following modules and a brief explanation of the actions performed in each module is given below.

Authentication Module:

When the program is run, the Administrator is provided with a Dialog Box prompting him to enter the Password for Authentication. The Password is Stored initially in the Windows Registry at the time of installation. The password is encrypted. The Password has to be retrieved from the Registry at the time of invocation by querying the Registry and it is decrypted.

In the Dialog Box the user types the password and this password is checked with the decrypted password from the Registry. If the Password matches, the user can log into the Application. Else, the same screen will be displayed again for reentry or for cancellation.

Audio Analysis Module:

In this module a wave sound to be analyzed is opened first. The audio file should be of 'WAVE' type. This module checks whether the file is wave file and then checks whether it is a Stereo or Mono file. Once this is ascertained, the functions in the module plot the frequency. For a stereo wave file, both the left and right channels are plotted. If the file is a mono one, only a single channel is plotted. This module shows all the details regarding the file's rate of sampling, number of channels, 8 bit or 16 bit, file size, number of milliseconds needed to play the file completely. This module also contains provisions for playing, pausing, and stopping the audio file. There is also facility for continuous playing of the audio file.

Once the wave file is plotted, the user can select a particular portion for analysis. The user has to simple click and drag the mouse in the wave plotted area to select the starting and end positions. The selected area is marked in a different color for easy identification. Details of the selected portion such as samples, time, starting position, end position, and total length are displayed in label controls.

The selected portion of the wave file can be zoomed a number of times for close observation. The selected portion as well as the zoomed portion can be copied and pasted into a new form. The selected portion can also be played back. When the wave file is opened, the frequency rates are also displayed in a list control. The values displayed in the list control are used for analysis and comparison of different wave files.

Audio Training Module:

This module presents the user to train different word sounds. In this module the user is expected to provide the different phonemes at varying speeds and varying intensities. A large collection of these will help in determining the comparison between two audio files easier.

Synchronized Sounds Module:

In this module the user is presented with an interface that contains different selection options to choose multiple MP3 format files. Once the songs are selected, any or all of the songs can be played simultaneously. This is very useful for DJs, and sound studios for mixing different sounds.

Text to Speech Module:

This module contains a speech engine that converts the written text in to spoken words. The True Voice Speech Engine is a collection of DLLs that formulate the written text in to speech by analyzing different syllables and their pronunciation.

Audio Mixer Module:

The audio mixer module contains controls that interact with the sound hardware of the computer by way of low level APIs. Using this module a user can set various parameters like Bass, Treble, Master Volume, CD Volume, Wave Volume, Line in, etc. The parameters set in this module persist even after closing the application.

Utilities Module:

The utilities are the Audio reminder that plays a particular song at the particular time and also plays the role of Scheduler.

Help Module:

This module contains details regarding proper usage of this software.

PROGRAMMING ENVIRONMENT

3.0 PROGRAMMING ENVIRONMENT

3.1 HARDWARE CONFIGURATION

Minimum hardware configuration required for implementing this software:

- A Processor of the following type and speed:

  Pentium processor with clock speed of 266 MHz and above.

  Celeron processor with clock speed of 400 MHz and above.

  AMD Athlon processor with clock speed of 400 MHz and above

- An Intel / Mercury / Asus Mother Board.
- SDRAM or RDRAM or DDRAM with a minimum capacity of 64 MB.
- A very good quality Sound Card
- A Hard disk drive with a minimum capacity of 10 GB.
- A CD ROM Drive with a minimum speed of 24X.
- A Good Quality 15 inch Monitor like LG / Samsung / Samtron etc.
- A Multimedia Keyboard.
- A Very good quality Microphone.
- A good quality Mouse.
- A Floppy drive of 1.44 MB size.
- External Speakers.

3.2 DESCRIPTION OF SOFTWARE AND TOOLS USED

INTRODUCTION TO VISUAL BASIC:

Visual Basic has been chosen as it provides a GUI based environment for creating user-friendly forms. Visual Basic is an ideal programming language for developing sophisticated applications in Windows platform. The 'Visual' part refers to the graphical user interface (GUI). Rather than writing numerous lines of code to describe the appearance and location of interface elements, we simply use pre-built objects into place on the screen. The 'Basic' part refers to the BASIC language. Visual Basic has evolved from the original BASIC language and now contains several hundred statements, functions and keywords, many of which relate directly to the Windows GUI.

Beginners can create useful applications by learning a few keywords yet the power of the language allows professionals to accomplish anything that can be accomplished using any other windows programming language. It makes use of Graphical User Interface (GUI) for creating robust and powerful applications. The GUI enables the users to interact with an application. This feature makes it easier to comprehend things in a quicker and easier way. In a GUI environment, coding is similar to linear programming methods and it is highly interactive and user-friendly. One of the interesting features of Visual Basic is the Integrated Development Environment (IDE). Another important feature of Visual basic is that it has easy methods to allow users to control and access databases. The front end can also be connected to the databases via ODBC, JDBC, etc.

<u>Components of Visual Environment</u>

The components of the VB environment are:

1. VB window
2. Project Window
3. Menu Bar

4. Toolbox
5. Properties Window
6. Form Window
7. Data Report Designer
8. Data Environment Designer

VB Window

It contains title bar, menu bar and tool bar.

Menu Bar

It displays an option that enables to develop VB application

Toolbox

Provides control tools that enable to add features such as buttons, text boxes and dialog boxes to the application. Some of the controls include the following:

Label displays a text that the user cannot modify or interact with.

Frame control serves as a visual and functional container for controls.

Check box displays a True/False or Yes/No option

Textbox is a control used to display message and enter text.

Combo box contains Textbox and a List box. This allows the user to select an item from the Dropdown List box, or to type in a selection in the Text box.

Command Button carries out the specified action when the user chooses it.

Project Window

List the files that make up the application

Form Window

The user interface consists of several different components put together as required. These components cannot just be placed in screen. They have to be placed in some sort of a container. When a user interface is being created, the creation of the window is the first step. It is after the window is created that the various controls may be positioned in it. A window is also referred to as a FORM.

Project Window

It describes the characteristics of the components. It can also enable to change the components also. We can also modify the components properties at the run time.

Data Report Designer

This is one of the most exciting features of Visual Basic. Instead of needing a third-party utility to build simple reports, we can do it within Visual Basic. The resulting report object can then be called programmatically and used within our program. It supports page and report headers, detail lines and many other common features, including variety of graphics and font features.

Data Environment Designer

A concept very new to this version of Visual Basic is that of the Data Environment. Instead of being familiar about the several places where the database file is referenced, we need to make only a single change that is propagated throughout our application. A Data Environment is very similar to a data control. It is always available, but instead of just linking to a single query or table, we can link it to all our databases, tables and queries with a single object.

FEATURES OF VISUAL BASIC:

Object Linking and Embedding

OLE is a means of communication, which gives applications the power to directly use and manipulate other Windows applications. It is available in Visual Basic. This permits the user to link and embed objects, and also edit the object within the container application.

ActiveX Controls

ActiveX controls can be developed using wizards or can be designed using the interface of the control by writing code. The different types of ActiveX controls offered by VB are,

- User-Drawn Controls
- Enhancements to existing Visual Basic Controls
- New controls built with constituent controls

Internet tools in Visual Basic

The Internet is one of the fastest growing and most exciting new areas of computer programming. It is possible to develop Web-browsing applications in Visual Basic using the Internet controls like the Web Browser control, WinSock control etc.,

Using DHTML in Visual Basic

DHTML is based on the Document Object Model, which is a hierarchy of web page elements. Elements are related to DHTML in the same manner in which controls are related to Visual Basic applications.

- Open multiple projects in the same instance of VB.
- ActiveX documents boost the Visual Basic application to the internet and intranet browser windows.

- Ability to do single, multiple, or Microsoft explorer style document interface application.
- The new model allows us to programmatically extend the development environment and control project, events, code visual elements.
- The application wizard is new and the setup wizard has been enhanced to enable creating a dependency file for a standard projects.
- Most control now support drag and drop components specifically for employment of the web.

SOFTWARE REQUIREMENTS

Platform	- Windows 9x / 2000 / XP
Language	- Visual Basic 6 (Enterprise edition)
Tools	- True Voice Text to Speech (TTS) Engine
	- Advanced APIs and DLLs.
Front Design	- Adobe Photoshop

Data Dictionary

Sound Card:

It is a printed circuit board, or card, that can translate digital information into sound and back; also called a sound board or sound adapter. Sound cards plug into a slot on the motherboard (the main circuit board of a computer) and are usually connected to a pair of speakers. To play sounds, the sound card receives digital information from a stored file and turns it into an electrical signal it sends to the speakers, which produce the sound. If the sound card is attached to a microphone, the sound card can take the incoming sound and convert it into digital information by sampling, or taking tiny sections of, the sound many times each second (the most sophisticated sound cards can take almost 200,000 samples per second, but most take around 50,000 to 100,000 samples per second

Frequency:

It is a term used in the physical sciences to denote the number of times that any regularly recurring phenomenon occurs in one second. Frequency is important in many fields of science, such as mechanics, and the study of sound waves. Frequencies of oscillating objects can cover a wide range of values. The tremors of earthquakes may have a frequency of less than 1, while the rapid electromagnetic oscillations of gamma rays may have frequencies of 1020 or more. In almost all forms of mechanical vibration, a relationship exists between frequency and the physical dimensions of the vibrating object. The frequency or speed of vibration of a string of a musical instrument is partly determined by the length of the string. In each instance, the shorter the object, the higher the frequency of vibration.

In wave motion of all kinds, the frequency of the wave is usually given in terms of the number of wave crests that pass a given point in a second. The velocity of the wave and its frequency and wavelength are interrelated. The wavelength (the distance between successive wave crests) is inversely proportional to frequency and directly proportional to velocity. In mathematical terms, this relationship is expressed by the equation V = ë f, where V is velocity, f is frequency, and ë is wavelength. From this equation any one of the three quantities can be found if the other two are known.

Frequency is expressed in hertz (Hz); a frequency of 1 Hz means that there is 1 cycle or oscillation per second. The unit is named in honor of the German physicist Heinrich Rudolf Hertz, who first demonstrated the nature of electromagnetic wave propagation. Kilohertz (kHz), or thousands of cycles per second, megahertz (MHz), or millions of cycles per second, and gigahertz (GHz), or billions of cycles per second, are employed in describing certain high-frequency phenomena, such as radio waves.

Electromagnetic waves of extremely high frequencies, such as light and X rays, are usually described in terms of their wavelength measure, which is often expressed in angstrom units (Å; hundred-millionths of a cm).

Sound intensities:

These are measured in decibels (dB). For example, the intensity at the threshold of hearing is 0 dB, the intensity of whispering is typically about 10 dB, and the intensity of rustling leaves reaches almost 20 dB. Sound intensities are arranged on a logarithmic scale, which means that an increase of 10 dB corresponds to an increase in intensity by a factor of 10. Thus, rustling leaves are about 10 times louder than whispering.

Sound Quality:

It is the characteristic of sound that allows the ear to distinguish between tones created by different instruments, even when the sound waves are identical in amplitude and frequency. Overtones are additional components in the wave that vibrate in simple multiples of the base frequency, causing the differences in quality, or timbre. The ear perceives distinctly different qualities in the same note when it is produced by a tuning fork, a violin, and a piano.

Electronic Music:

Music that requires knowledge or use of electronic devices to produce or manipulate sound during its composition and performance. The sound may be produced entirely through electrical means, as by an electronic sound synthesizer, a complex system of generators that can originate and control sound. The essential feature is that electronic devices are necessary in composing. Electronic music is thus distinguishable from music composed in the traditional manner but played on an electronic instrument such as the electronic organ or electric guitar, and from music played through an electronic medium such as a phonograph or radio.

SYSTEM DESIGN AND DEVELOPMENT

4.0 SYSTEM DESIGN

FEASIBILITY ANALYSIS

Feasibility analysis helps the system to truly meet the user expectations, feasibility analysis involves

(i) Technical feasibility

(ii) Behavioral feasibility

Technical Feasibility

This centers on the existing computer system and to what extent it can support the proposed system. The existing computer system has enough capacity to support the proposed system.

Behavioral feasibility

This is an estimate of how strong a reaction the user staff is likely to have toward the development of a computerized system.

4.1 INPUT DESIGN

Input is one of the expensive phases of the operation of a computerized system. The input data are the lifeblood of a system and has to be analyzed and designed with utmost care and consideration. Input system design features can ensure the reliability of the system and generated correct reports from the accurate data. It also determines whether the user can interact efficiently with the new system. Different types of problems occurring in the system can usually be traced back to faulty input design method. The input data is validated, edited, organized in an acceptable from by the system before being processed.

Error message are displayed in the screen when any invalid data was given as input in the system. While entering data, the user needs to know the following:

Type of audio file to be processed

The capability of the sound card to produce the required effect.

The sequence of operations required to perform a particular task.

A thorough understanding of how the software functions.

4.2 OUTPUT DESIGN

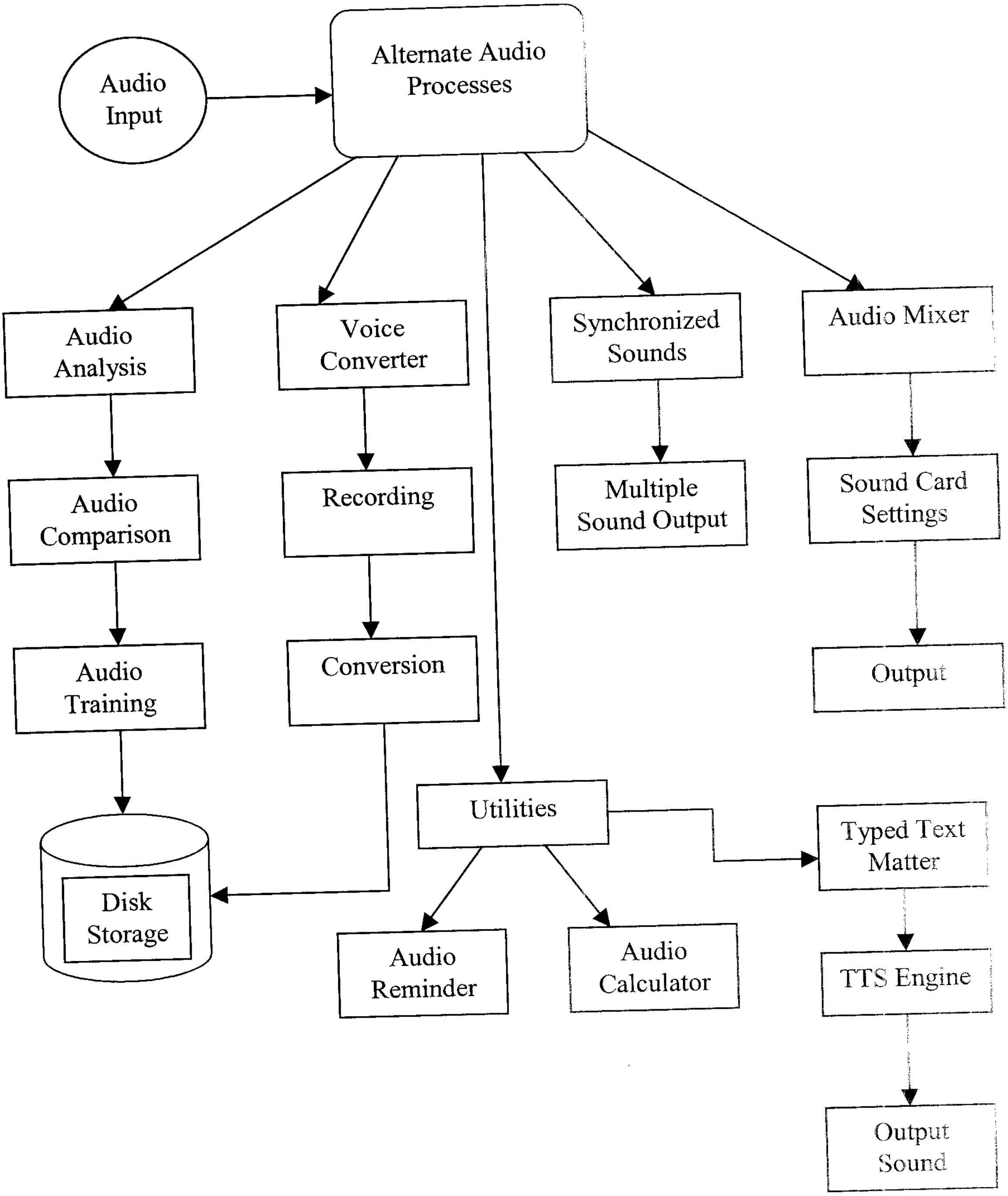
Computer output is most important and is a direct source of information to the user. Efficient and intelligible output design should improve the system's relationship with the user and help in decision making. The nature of processing and procedures related to the system were classified and verification whether they give the expected results as output.

Outputs from the computer system are required primarily to communicate the result of processing and provide permanent copy of these results for later consultation. While designing the output, the type of the output, concerning format, frequency, responses etc have been taken into consideration. The audio output plays a major role in the system.

4.3 CODE DESIGN

The purpose of the code is to facilitate the identification and retrieval of items of information. A code is an ordered collection of symbols designed to provide unique identification of an entity or attributes. In the system design phase, code design has an important role. The coding system is used to reduce the input, control errors and speed up the entire process. So coding system are methods in which conditions, words, idea or relationship are expressed by a code. The code designed offers uniqueness, expandability, conciseness, uniformity, simplicity, versatility, meaningfulness and operability.

4.4 DATA FLOW DIAGRAM



SYSTEM TESTING & IMPLEMENTATION

5.0 SYSTEM TESTING AND IMPLEMENTATION

Testing is the process of exercising of evaluating a system or system components by manual or automated means to verify that it satisfies specified requirements. Testing is a process of executing a program with the intent of finding errors. A good test case is one that has a high probability of finding an error. A successful test case is one that detects an as-yet-undiscovered error.

Software testing can be looked upon as one among the many processes an organization performs that provides the last opportunity to correct any errors in the developed system. Software testing includes selecting tests and test data that have more probability of finding errors.

The first step in system testing is to develop a plan that tests all aspects of the system. Completeness, correctness, reliability and maintainability of the software are to be tested for best quality assurance – an assurance that the system meets the specifications and requirements for its intended use and performance. System testing is the most practical process of executing a program with the explicit intention of finding errors that make the programs fail.

5.1 IMPLEMENTATION AND MAINTENANCE

Once the physical system has been designed in detail, the next stage is to turn the design into a working system and then to monitor the operation of the system to ensure that it continues to work efficiently and effectively

MODULE SPECIFICATION

The system is divided into modules such as

1) Audio Analysis Module
2) Audio Training Module
3) Synchronized Sounds Module

4) Voice Converter Module
5) Audio Mixer Module
6) Text to Speech Module
7) Utilities Module
8) Help Module

Audio Analysis Module

A wave sound to be analyzed is opened first. This module checks whether the file is wave file and then checks whether it is a Stereo or Mono file. Once this is ascertained, the function in the module plots the frequency. For a stereo wave file, both the left and right channels are plotted. If the file is a mono one, only a single channel is plotted. This module also displays rate of sampling of a file, number of channels, 8 bit or 16 bit, file size, number of milliseconds needed to play the file completely. Provisions for playing, pausing, and stopping the audio file are implemented. There is also facility for continuous playing of the audio file.

Once the wave file is plotted, the user can select a particular portion for analysis. The user has to simple click and drag the mouse in the wave plotted area to select the starting and end positions. The selected area is marked in a different color for easy identification. Details of the selected portion such as samples, time, starting position, end position, and total length are displayed in label controls.

The selected portion of the wave file can be zoomed a number of times for close observation. The selected portion as well as the zoomed portion can be copied and pasted into a new form. The selected portion can also be played back.

Audio Training Module

When the wave file is opened, the frequency rates are also displayed in a list control present in the module. The values displayed in the list control are used for analysis and comparison of different wave files. This module also has facility for recording sounds for comparison.

Synchronized Sounds Module

In this module the user is presented with an interface that contains different selection options to choose multiple MP3 format files. Once the songs are selected, any or all of the songs can be played simultaneously. This is very useful for DJs, and sound studios for mixing different sounds. Using this synchronization, a studio can effectively mix different songs to produce a single song that contains traces of the other songs. In short this plays any number of sound / song files at the same time.

Voice Converter Module

The Voice Converter Module is used to convert tones and pitches into different modes by adding additional sample bits and deleting sample bits from the original file. This results in the conversion of a tone from bass to treble or vice versa. The process is graphically plotted. The parameters are displayed on the form. The VCM uses the technique of adding additional sampled bits to the existing file. This results in the extension of the new file. When the new file is played back, the sound is heard as a bass (male) tone. The VCM also uses the technique of deleting samples bits from the existing file thereby reducing the samples. This results in the tone being heard as a treble one (female).

Audio Mixer Module

The audio mixer module contains controls that interact with the sound hardware of the computer by way of low level APIs. This module describes how to detect if a system has a sound card capable of playing back audio files. The multimedia application programming interface has a function which returns the number of devices in the system capable of playing back audio data. Using this module a user can set various parameters like Bass, Treble, Master Line in, Volume, CD Volume, Wave Volume, etc. The parameters set in this module persist even after closing the application.

Text to Speech Module

This module contains a speech engine that converts written text in to spoken words. The True Voice Speech Engine is a collection of DLLs that formulate the written text in to speech by analyzing different syllables and their pronunciation. The TTS synthesis by highlighting its digital signal processing and natural language processing components is a pc - based system that is to read text aloud, whether it was directly introduced in the computer by an operator or scanned and submitted. This can be used to perform for those who are visually retarded.

Utilities Module

This module contains an audio reminder utility that can be used to set a song to be played at a predefined alarm time. The song file can be a Wave file or an MP3 file type. The selected song will play at the time set by the user.

Maintenance

The first maintenance activity occurs since it is unreasonable to assume that testing will uncover all errors in a large system. The process of including the diagnosis and correction of one or more errors is called corrective maintenance.

The second activity that contributes a definition of maintenance occurs since rapid change is encountered in every aspect of computing.

The third activity involves recommendations for new capabilities, modifications to the existing functions and general enhancements when the software is used. Software is changed to improve future maintainability or reliability. This is called as preventive maintenance.

Maintenance Characteristics

The only available elements of a software configuration are source code; maintenance activity begins with an evolution of the code, often complicated by poor internal documentation. The suitable characteristics such as program structure, global data structure, system interfaces, and performance and design constraints are difficult to handle and are often misinterpreted.

Maintenance Cost

The cost of software maintenance has increased steadily during the past several years. One intangible cost of software maintenance is the development opportunity that is postponed or lost since the available resources must be channeled to maintenance tasks.

Problems

Most of the problems associated with software maintenance can be traced. The problems are:

It is often difficult or impossible to trace the evaluation of the software was created.

It is difficult to understand some one else's program without adequate remarks in the code module.

The maintenance has not been viewed as glamorous work. Much of these perceptions come from the high frustrations level associated with maintenance work.

Maintainability

Maintainability can be defined as the case with which software can be, corrected, adopted and enhanced.

Controlling Features

Availability of qualified software staff.

Understandable system structure.

Ease of system handling.

Use of standardized programming languages.

Use of standardized operating system.

Standardized structure of documentation.

5.2 TESTS PERFORMED

INTRODUCTION:

A number of experiments are conducted to test the software for bugs. These experiments are listed below. The experiments are classified depending on the inputs that the user has to specify and the computations and processing that are done with the code.

OPERATING SYSTEM & HARDWARE:

Any software needs memory to run and also the processor utilization for the software is also very important. We have tested our software under different processor speeds and on different operating system platforms. The test is also extended by checking the memory utilization by running the same software by varying the RAM capacities. We have also noted the effects of different sound cards, microphones, and speakers and their impact on the ultimate analysis of sounds. The results are as follows:

These are the results that occurred on testing the software under different operating systems.

Windows Platforms

Microsoft Windows 95	Works fine
Microsoft Windows 98	Works fine
Microsoft Windows ME	Works fine
Microsoft Windows XP	Works fine

Processor Speed

Pentium I 166 MHz.	Works Well (With 16/32 Mb SD Ram)
Pentium II 233 MHz.	Works Well (With 32 Mb SD Ram)
Pentium II 500 MHz.	Works Well (With 32 Mb SD Ram)
Pentium III 500 MHz.	Works Well (With 64 Mb SD Ram)
Pentium III 833 MHz.	Works Well (With128 Mb SD Ram)
Pentium IV 1.4 GHz.	Works Well (With 128 Mb SD Ram)

Sound Card

Since the project's theme is 'Analysis of Sound', a very good sound card is imperative. We checked the software in systems containing the following sound cards.

Creative Sound Blaster	Works fine
Lancer Sound Card	Works fine

Microphone

A high-fidelity microphone is needed for recording the sounds which are used for comparing audio sounds and also for Audio Conversion.

Kartz1003 Microphone	Works very well

We came to the conclusion that any Hardware and OS fulfilling the above criteria works fine for this software.

MODULE TESTING

Each individual program module was tested for any possible logical effort. They were also tested for specifications to see if they are working as per what the program is supported to do and how it should perform under various conditions.

VOLUME TESTING

The user has provided the test data for this kind of test. This was made to check whether the hardware and software are functioning correctly when large amount of data is supplied.

USABILITY AND DOCUMENT TESTS

This is to verify the user-friendliness of the system developed. Normal operating and effort handling procedures are related to this. Accuracy and completeness of documentation is checked here.

PROGRAM TESTING

This test is used to check the errors in syntax and logic. To detect errors, the actual output is compared with the expected output. When a mismatch occurs the instruction sequence is traced to detect the error.

STRING TESTING

Here each portion of the system is tested against the entire module with the test data provided by the user. This is done because programs are related to one another and they interact in a total system.

QUALITY ASSURANCE

Quality assurance is the review of software and related documentation for correctness, accuracy, reliability, maintainability and expandability. This also includes assurances that the system meets the specification and the requirements for its intended use and performance.

CONCLUSION

.0 CONCLUSION

This project provided me the opportunity to learn different aspects of 'Audio Digital Processing'. The different features present in this project make this a highly useful one in the audio profession. The Significant additional functionality, performance gains, and low overhead by the advanced APIs contributed a lot in the successful completion of this project. Usage of the Visual Basic Language and its RAD (Rapid Application Development) features enabled me to complete the project with in the allotted time.

Usage of this language has not in any way diminished the operability and performance of the difficult functions because of the performance gains provided by the DLLs and APIs. From formation of the idea to finalization, this project provided me the thrust for trying something different.

SCOPE FOR FUTURE DEVELOPMENT

.0 SCOPE FOR FUTURE DEVELOPMENT

This project could be ported to the Visual Studio.net framework. Even though such onversion will not provide any significant performance gains, the fact that the project will be vailable for performance in different platform will enhance the chance of its Functionality. Also uture development involves the building of an integration module so that the system could be ntegrated with several standard packages.

BIBLIOGRAPHY

8.0 BIBLIOGRAPHY

Books:

Childers D.G., 'Gender recognition from speech. Part II: Fine analysis', J. Acoust Soc, 1991, pp. 1841-1856.

Klatt, Dennis and Klatt, 'Analysis, synthesis, and perception of voice quality variations among female and male talkers', 1990, pp. 820-857.

Markel, John D, 'Digital inverse filtering - a new tool for formant trajectory estimation' IEEE Trans. Audio and Electro acoustics, 1972, pp. 129-137.

Jan P. H. van Santen, 'Prosodic modeling in text-to-speech synthesis', Proceedings of EuroSpeech97, 1997, Keynote Speech.

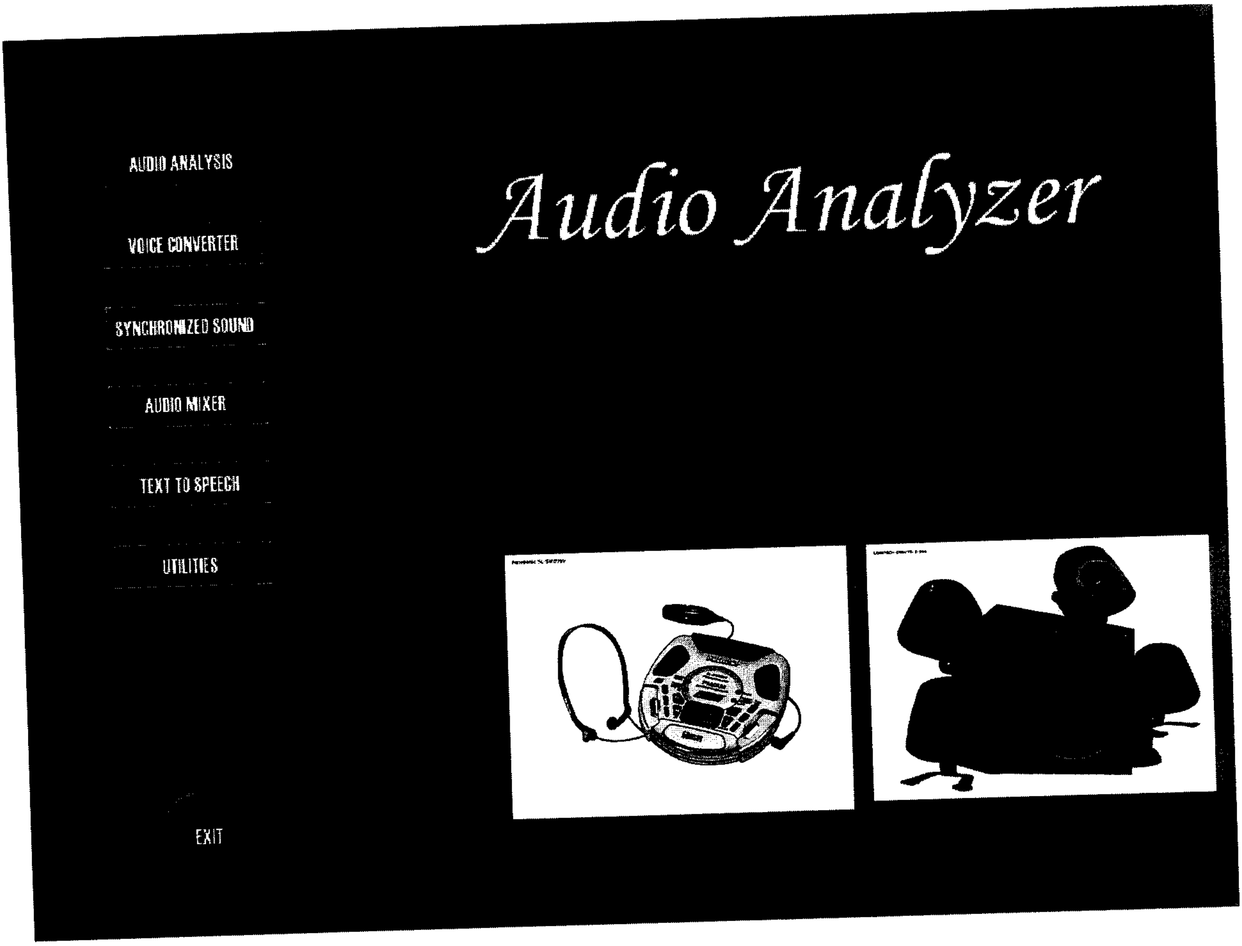
Childers, D.G., 'Voice Conversion and Speech Communication', Hicks, D.M., 1989, pp. 230-312

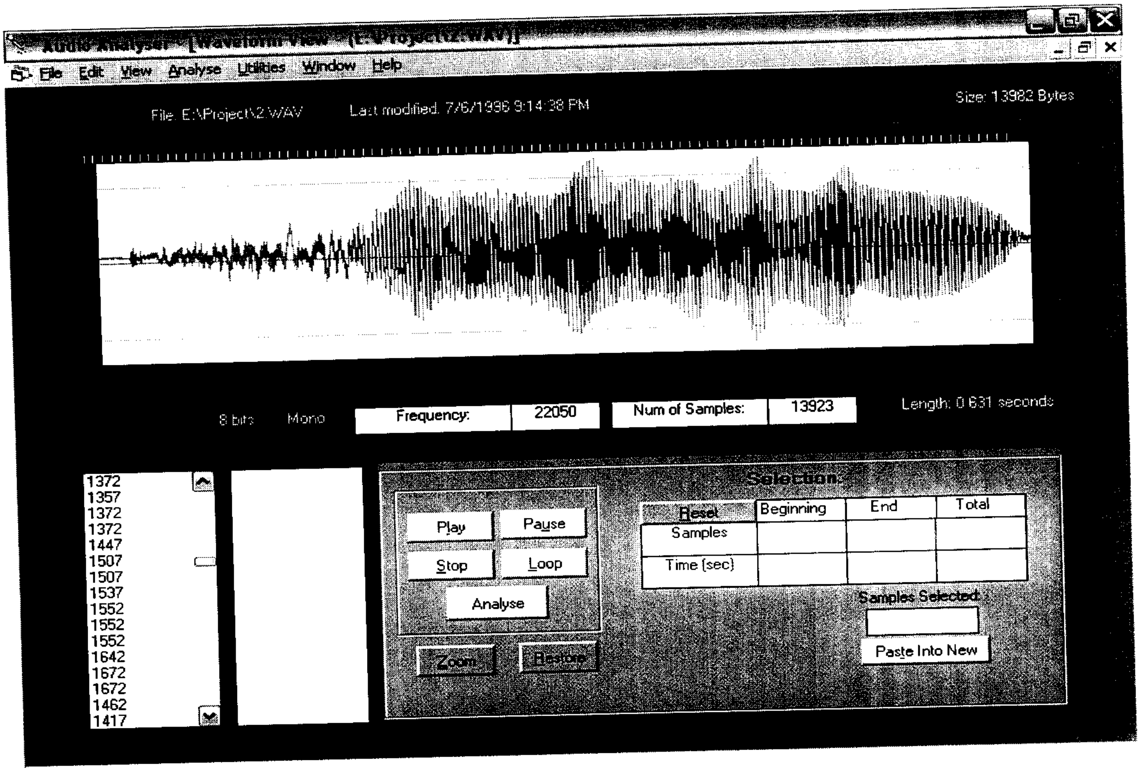
Prof. Garyth Nair, 'Voice-Tradition and Technology', A State-of-the-Art Studio, 1990, 00. 47-95

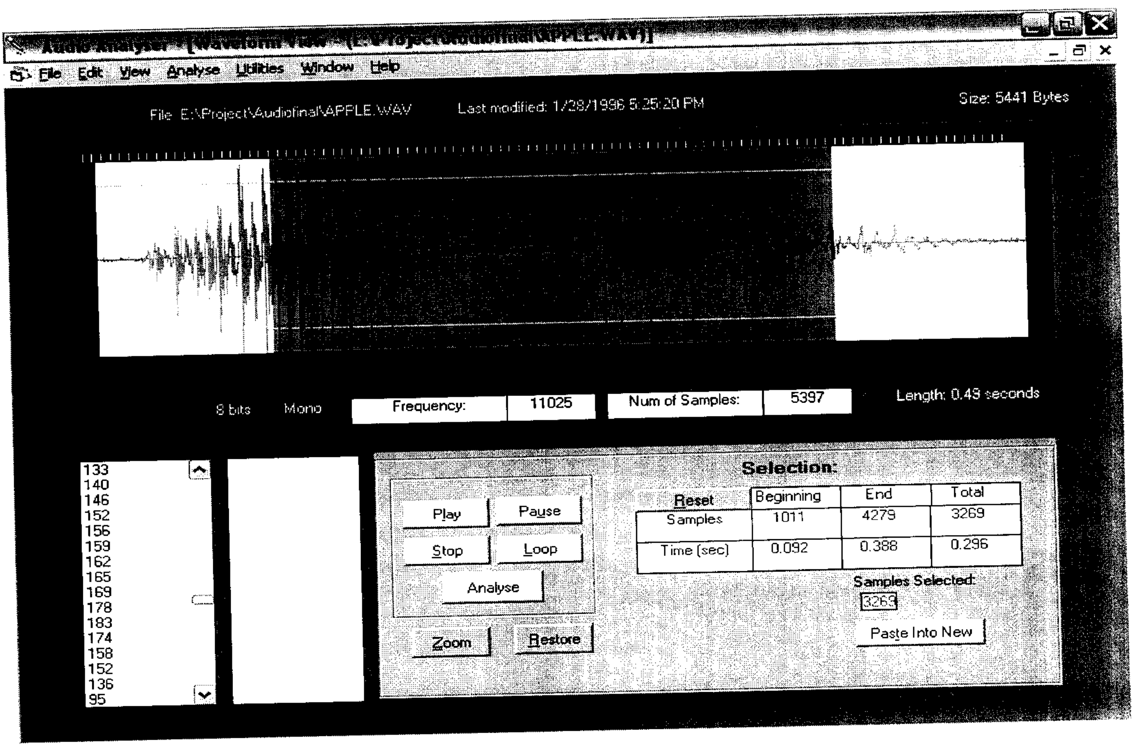
APPENDIX

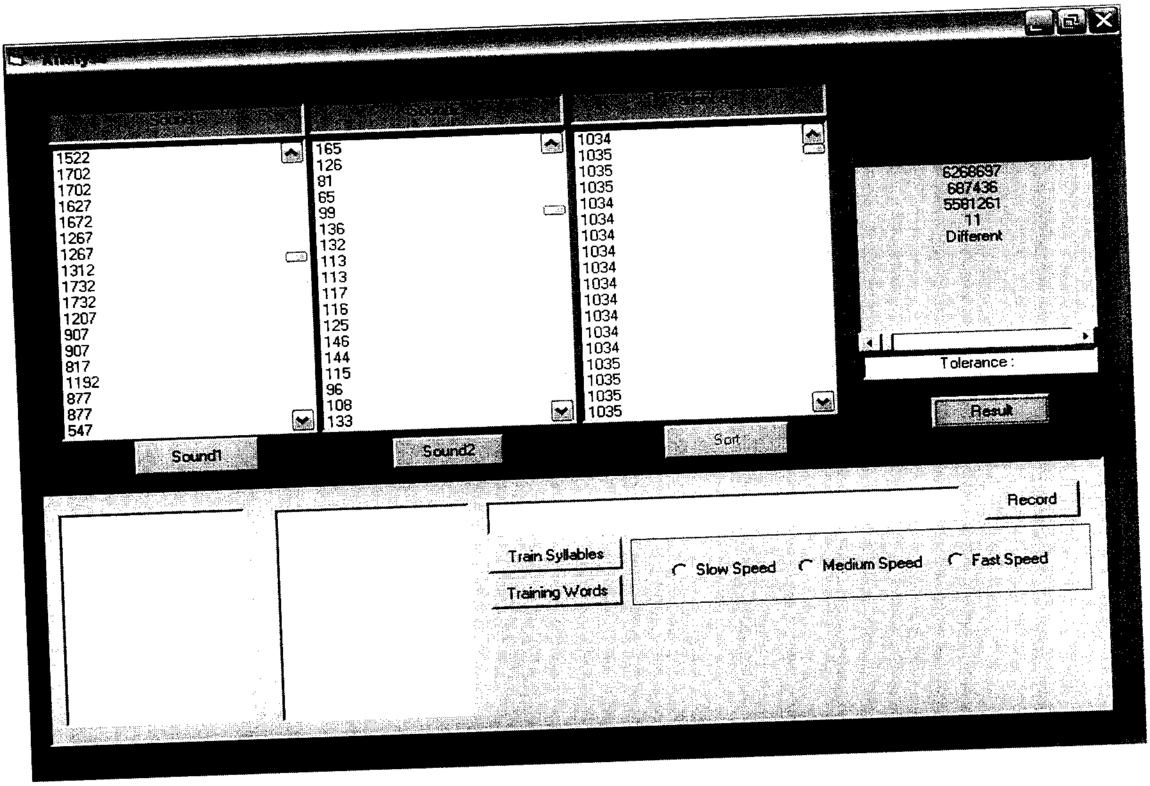
9.0 APPENDIX

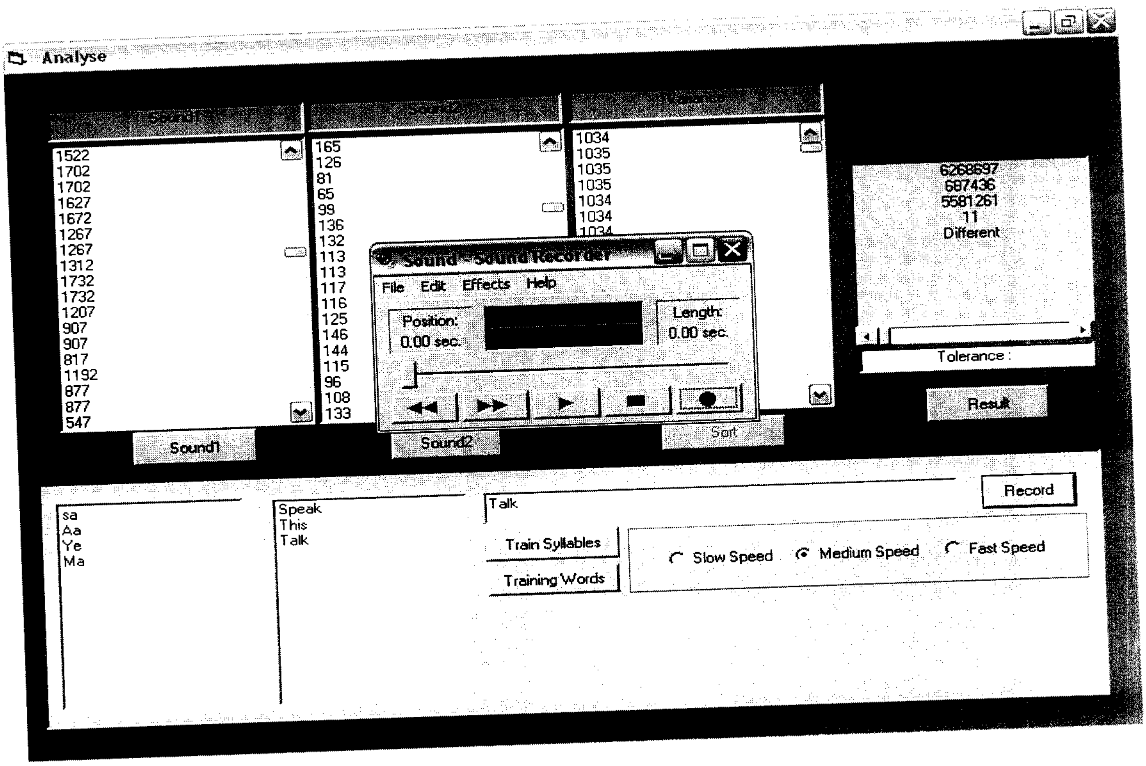
9.1 Sample screens

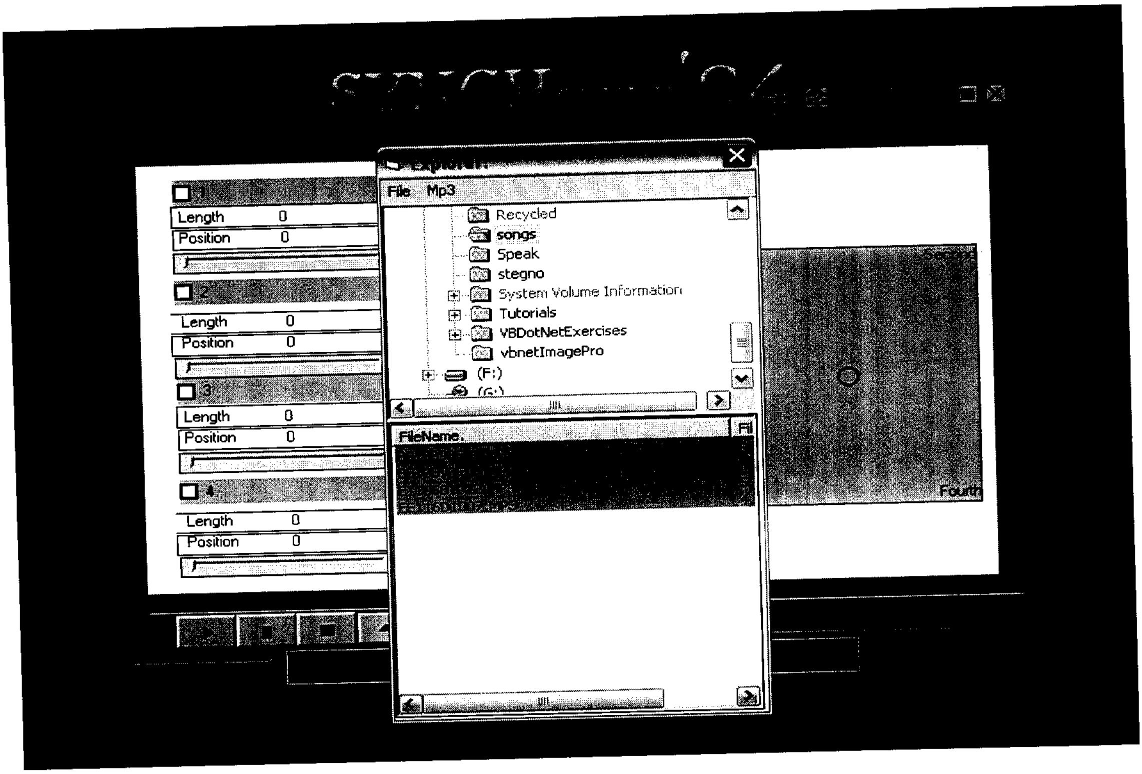


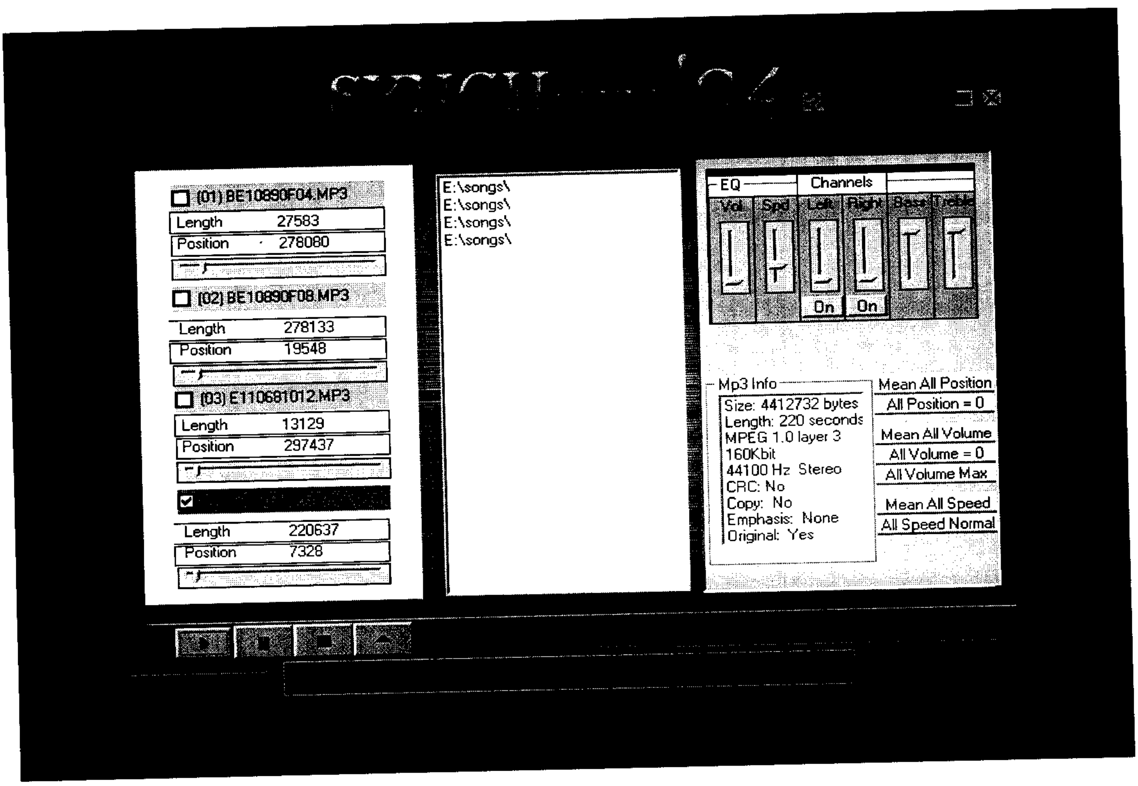

File Edit View Analyse Utilities Window Help
File: E:\Project\2.WAV
Last modified: 7/6/1996 9:14:38 PM
Size: 13982 Bytes
8 bits
Mono
Frequency: 22050
Num of Samples: 13923
Length: 0.631 seconds
1372
1357
1372
1372
1447
1507
1507
1537
1552
1552
1552
1642
1672
1672
1462
1417
Selection:
Play
Pause
Stop
Loop
Analyse
Zoom
Restore
Reset
Beginning
End
Total
Samples
Time (sec)
Samples Selected:
Paste Into New

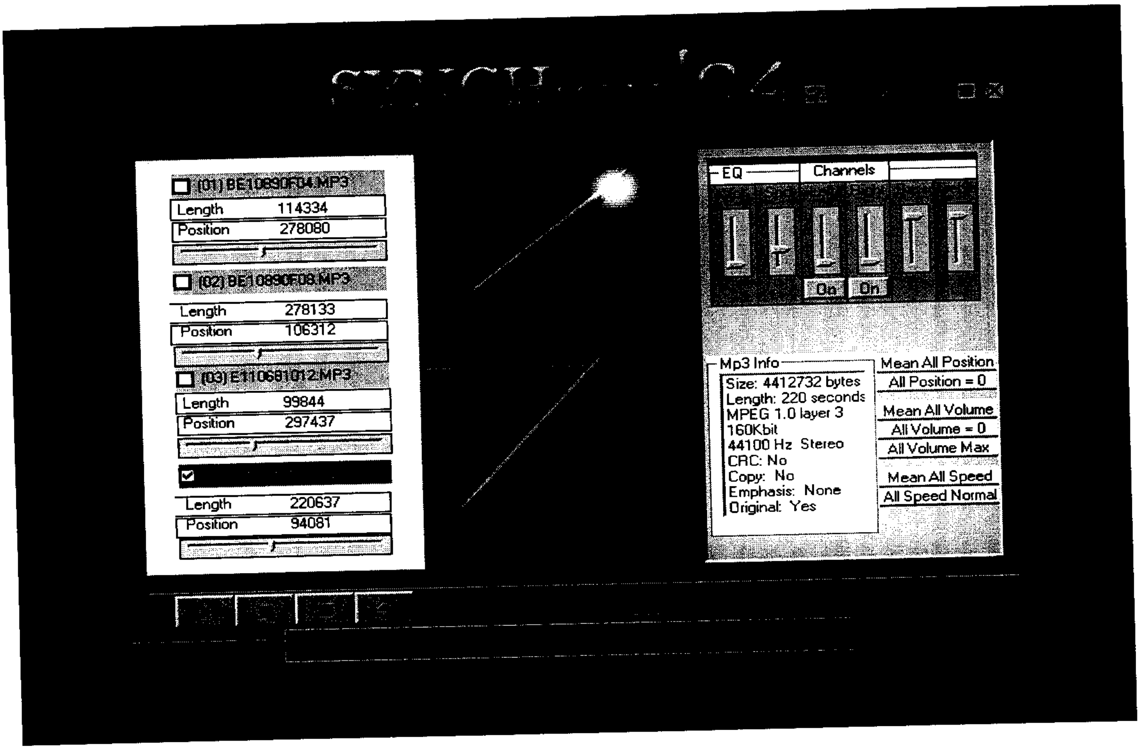

File Edit View Analyse Utilities Window Help
File: E:\Project\Audiofinal\APPLE.WAV
Last modified: 1/28/1996 5:25:20 PM
Size: 5441 Bytes
8 bits
Mono
Frequency: 11025
Num of Samples: 5397
Length: 0.49 seconds
133
140
146
152
156
159
162
165
169
178
183
174
158
152
136
95
Selection:
Play
Pause
Stop
Loop
Analyse
Zoom
Restore
Reset
Beginning
End
Total
Samples 1011 4279 3269
Time (sec) 0.092 0.388 0.296
Samples Selected:
3269
Paste Into New


Analyse
Sound1
Sound2
Sort
Result
Tolerance :
6268697
687436
5581261
11
Different
Record
Train Syllables
Training Words
Slow Speed
Medium Speed
Fast Speed

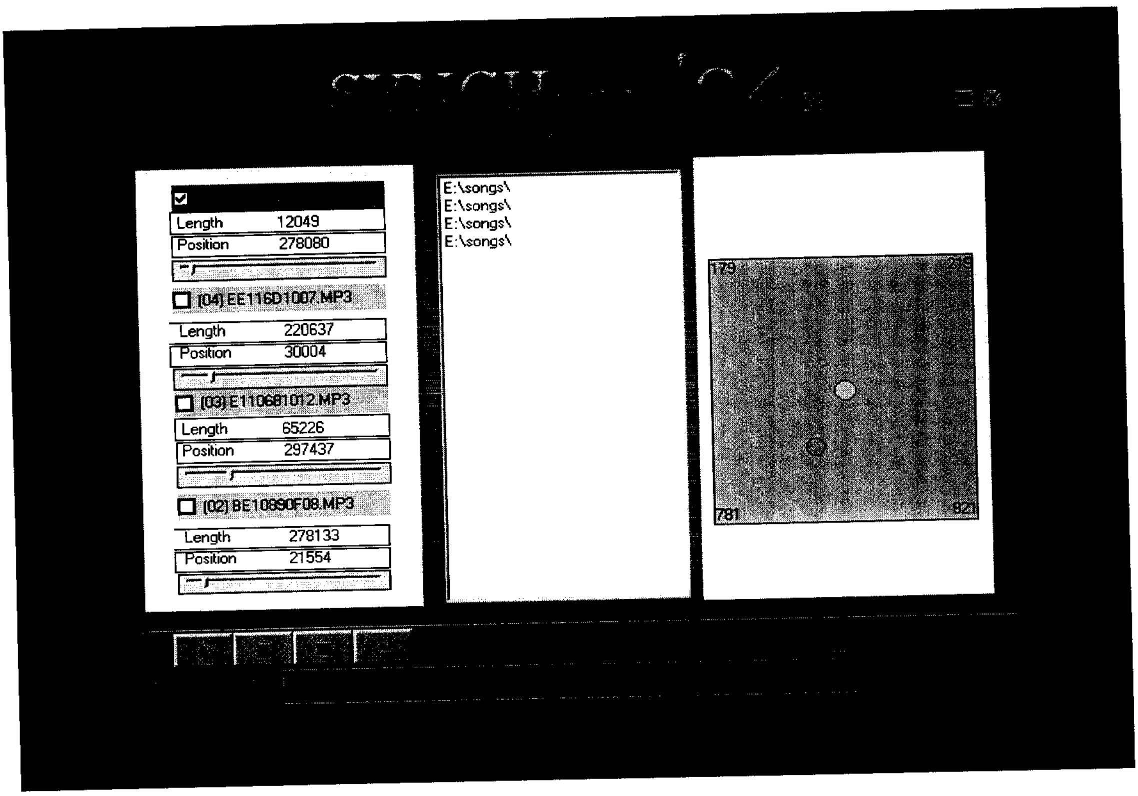

Analyse
Sound1
Sound2
Sort
Result
Tolerance :
6268697
687436
5581261
11
Different
File
Edit
Effects
Help
Position:
0.00 sec.
Length:
0.00 sec.
sa
Aa
Ye
Ma
Speak
This
Talk
Talk
Record
Train Syllables
Training Words
Slow Speed
Medium Speed
Fast Speed


File Mp3
Recycled
songs
Speak
stegno
System Volume Information
Tutorials
VBDotNetExercises
vbnetImagePro
(F:)
FileName
Length 0
Position 0
Fourth

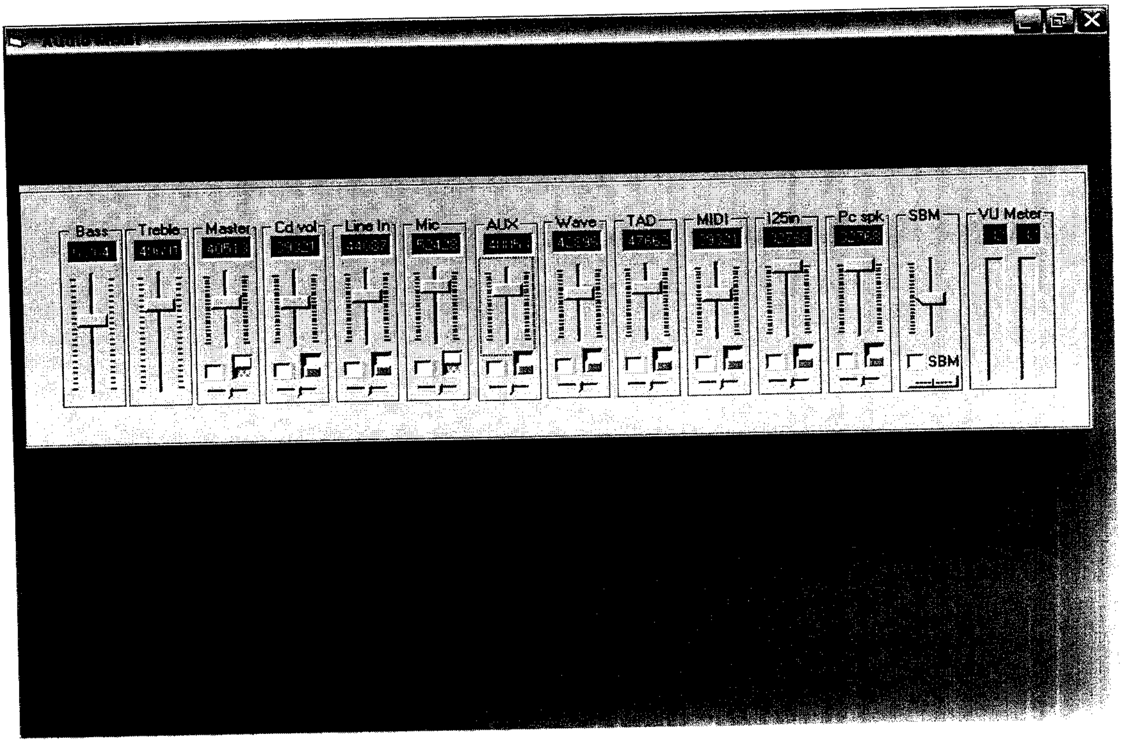

(01) BE10890F04.MP3
Length 27583
Position 278080
(02) BE10890F08.MP3
Length 278133
Position 19548
(03) E110681012.MP3
Length 13129
Position 297437
Length 220637
Position 7328
E:\songs\
E:\songs\
E:\songs\
E:\songs\
EQ
Channels
Vol
Spd
Left
Right
On
On
Mp3 Info
Size: 4412732 bytes
Length: 220 seconds
MPEG 1.0 layer 3
160Kbit
44100 Hz Stereo
CRC: No
Copy: No
Emphasis: None
Original: Yes
Mean All Position
All Position = 0
Mean All Volume
All Volume = 0
All Volume Max
Mean All Speed
All Speed Normal

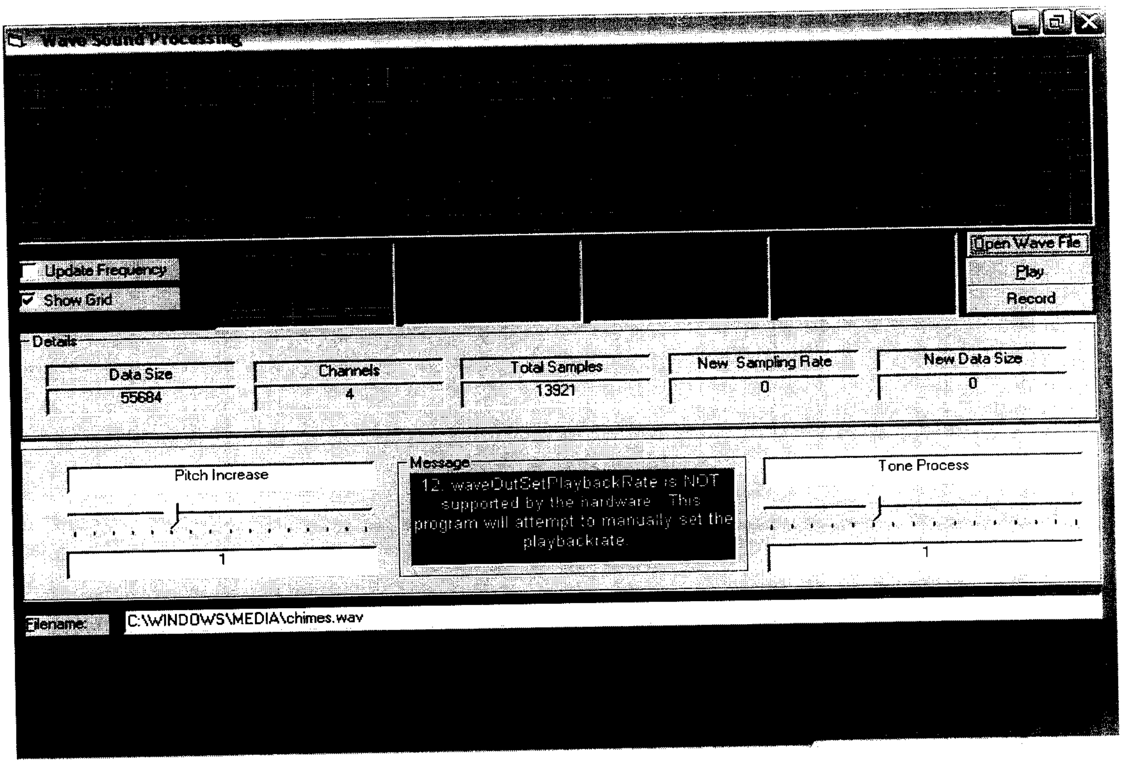


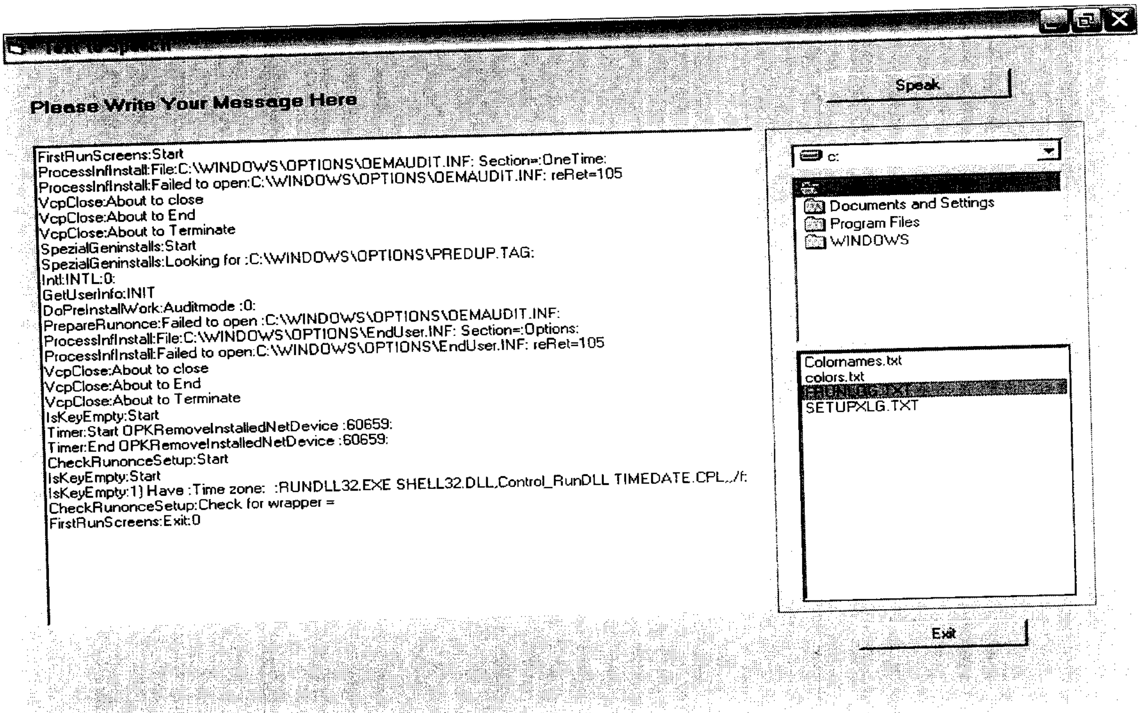
[01] BE10890F04.MP3
Length 114334
Position 278080
[02] BE10890F08.MP3
Length 278133
Position 106312
[03] E110681012.MP3
Length 99844
Position 297437
Length 220637
Position 94081
EQ
Channels
On
On
Mp3 Info
Size: 4412732 bytes
Length: 220 seconds
MPEG 1.0 layer 3
160Kbit
44100 Hz Stereo
CRC: No
Copy: No
Emphasis: None
Original: Yes
Mean All Position
All Position = 0
Mean All Volume
All Volume = 0
All Volume Max
Mean All Speed
All Speed Normal

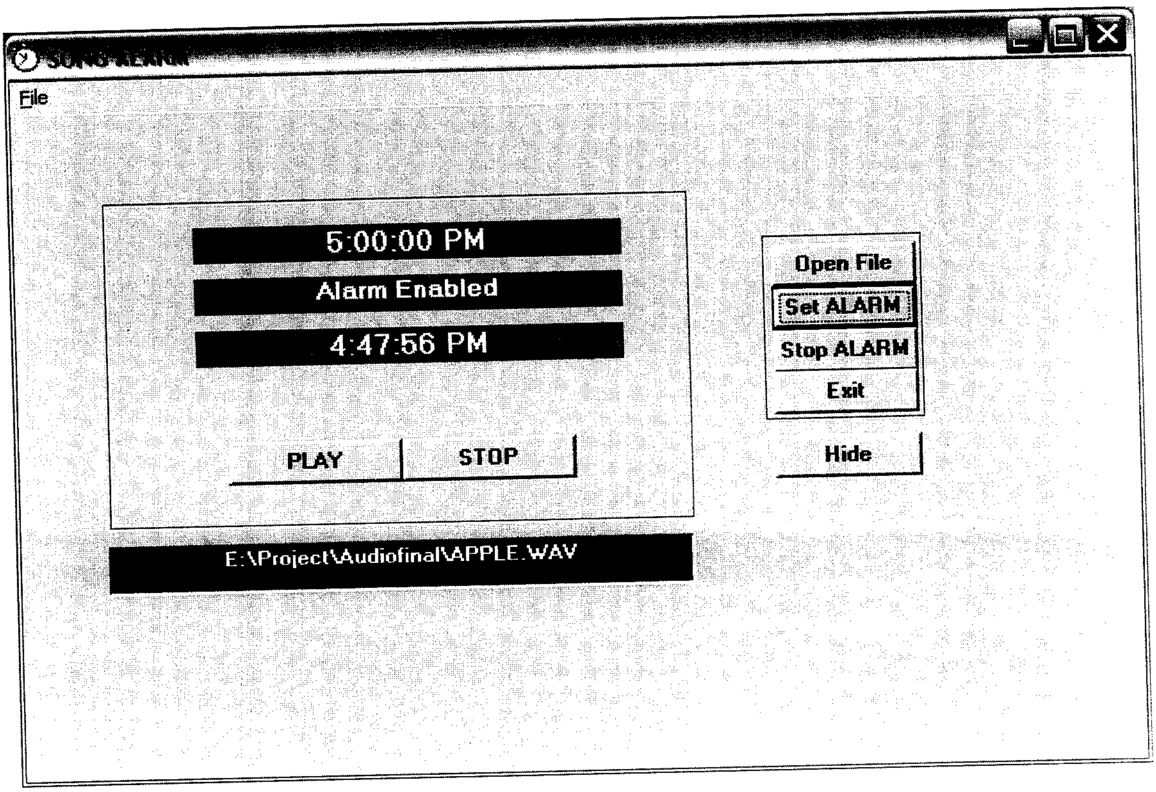


Length 12049
Position 278080
[04] EE116D1007.MP3
Length 220637
Position 30004
[03] E110681012.MP3
Length 65226
Position 297437
[02] BE10890F08.MP3
Length 278133
Position 21554
E:\songs\
E:\songs\
E:\songs\
E:\songs\
179
781
821


Bass
Treble
Master
Cd vol
Line In
Mic
AUX
Wave
TAD
MIDI
125in
Pc spk
SBM
VU Meter
SBM


Wave Sound Processing
Open Wave File
Play
Record
Update Frequency
Show Grid
Details
Data Size
55684
Channels
4
Total Samples
13921
New Sampling Rate
0
New Data Size
0
Pitch Increase
1
Message
12. waveOutSetPlaybackRate is NOT supported by the hardware. This program will attempt to manually set the playbackrate
Tone Process
1
Filename:
C:\WINDOWS\MEDIA\chimes.wav


Please Write Your Message Here
Speak
FirstRunScreens:Start
ProcessInfInstall:File:C:\WINDOWS\OPTIONS\OEMAUDIT.INF: Section=:OneTime:
ProcessInfInstall:Failed to open:C:\WINDOWS\OPTIONS\OEMAUDIT.INF: reRet=105
VcpClose:About to close
VcpClose:About to End
VcpClose:About to Terminate
SpezialGeninstalls:Start
SpezialGeninstalls:Looking for :C:\WINDOWS\OPTIONS\PREDUP.TAG:
Intl:INTL:0:
GetUserInfo:INIT
DoPreInstallWork:Auditmode :0:
PrepareRunonce:Failed to open :C:\WINDOWS\OPTIONS\OEMAUDIT.INF:
ProcessInfInstall:File:C:\WINDOWS\OPTIONS\EndUser.INF: Section=:Options:
ProcessInfInstall:Failed to open:C:\WINDOWS\OPTIONS\EndUser.INF: reRet=105
VcpClose:About to close
VcpClose:About to End
VcpClose:About to Terminate
IsKeyEmpty:Start
Timer:Start OPKRemoveInstalledNetDevice :60659:
Timer:End OPKRemoveInstalledNetDevice :60659:
CheckRunonceSetup:Start
IsKeyEmpty:Start
IsKeyEmpty:1} Have :Time zone: :RUNDLL32.EXE SHELL32.DLL,Control_RunDLL TIMEDATE.CPL,,/f:
CheckRunonceSetup:Check for wrapper =
FirstRunScreens:Exit:0
c:
Documents and Settings
Program Files
WINDOWS
Colornames.txt
colors.txt
SETUPXLG.TXT
Exit


File
5:00:00 PM
Alarm Enabled
4:47:56 PM
PLAY
STOP
Open File
Set ALARM
Stop ALARM
Exit
Hide
E:\Project\Audiofinal\APPLE.WAV

.2 Sample code

```
Private Sub Command1_Click()      'Play
  If PlayControls = False Then
     Picture2.SetFocus
     Exit Sub
  End If
  RepeatIt = 0
  Call MMControl1_PlayClick(False)
  MMControl1.Command = "Play"
  Picture2.SetFocus
  Command1.Enabled = False: Command4.Enabled = False
End Sub

Private Sub Command3_Click()      'Stop
  If PlayControls = False Then
     Picture2.SetFocus
     Exit Sub
  End If
  RepeatIt = 0
  MMControl1.Command = "Stop"
  Picture2.SetFocus
  Command1.Enabled = True: Command4.Enabled = True
End Sub

Private Sub Chk_CH_Left_Click()                    'left channel on/off
   With Chk_CH_Left
     If .Value = 1 Then
        .Caption = "OFF"
        setMP3channelState wichMp3, "left", "off" 'set left channel off
      Else 'NOT .VALUE...
        .Caption = "ON"
        setMP3channelState wichMp3, "left", "on" 'set left channel on
        setMp3channelVolume wichMp3, "left", Sli_CH_Left.Value 'show left channel volume
     End If
   End With 'CHK_CH_LEFT

End Sub
```

```
Private Sub Sli_CH_Left_Scroll() 'slider left channel scroll
  setMp3channelVolume wichMp3, "left", Sli_CH_Left.Value 'set the left channel volume
  Sli_volume.Value = statusVolume(wichMp3) 'display volume
  Sli_CH_Right.Value = statusChannelVolume(wichMp3, "right") 'display lft channel  volume
  Label2(5).Caption = Sli_volume.Value 'display volume
  Label2(2).Caption = Sli_CH_Left.Value 'display left channel volume
  Label2(1).Caption = Sli_CH_Right.Value 'display right channel volume
  Label2(6).Caption = Slider1.Value 'display speed
End Sub

Private Sub Sli_volume_Scroll()
  setMp3Volume wichMp3, Sli_volume.Value 'set volume
  Sli_CH_Left.Value = statusChannelVolume(wichMp3, "left") 'display left channel volume
  Sli_CH_Right.Value = statusChannelVolume(wichMp3, "right") 'display rt channel volume
  Label2(5).Caption = Sli_volume.Value 'display volume
  Label2(2).Caption = Sli_CH_Left.Value 'display left channel volume
  Label2(1).Caption = Sli_CH_Right.Value 'display right channel volume
  Label2(6).Caption = Slider1.Value 'display speed
End Sub

Private Sub Cmd_Open_Click() 'open tree view
  Me.MousePointer = 11 'hourglass
  frmExplore.Show (1) 'show it
  Me.MousePointer = 0 'default
End Sub

Private Sub Cmd_stop_Click()
  Timer1(wichMp3).Enabled = False 'disable selected mp3
  StopMp3 wichMp3 'stop selected mp3
  CloseMp3 wichMp3 'close selected mp3
  lblLengthA(wichMp3).Caption = "0" 'display nothing
  lblPositionA(wichMp3).Caption = "0" 'display nothing
  Chk_pick(wichMp3).Caption = "No Track loaded." 'display nothing
  Frame1(wichMp3).BackColor = vbWhite 'let user know not on
End Sub

Private Sub Slider1_Scroll() 'set speed
  setMp3Speed wichMp3, Slider1.Value 'set speed
  Label2(5).Caption = Sli_volume.Value 'display volume
  Label2(2).Caption = Sli_CH_Left.Value 'display left channel volume
  Label2(1).Caption = Sli_CH_Right.Value 'display right channel volume
```

```
 Label2(6).Caption = Slider1.Value 'display speed
nd Sub

rivate Sub Sli_Treble_Scroll() 'slider treble scroll
  setMp3Treble wichMp3, Sli_Treble.Value 'set treble
nd Sub

rivate Declare Function waveOutOpen Lib "winmm.dll" _
  (lphWaveOut As Long, ByVal uDeviceID As Long, _
  lpFormat As Any, ByVal dwCallback As Long, _
  ByVal dwInstance As Long, ByVal dwFlags As Long) As Long

rivate Declare Function waveOutClose Lib "winmm.dll" (ByVal hWaveOut As Long) As
ong

rivate Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" _
  (hpvDest As Any, hpvSource As Any, ByVal cbCopy As Long)

rivate Declare Function GlobalAlloc Lib "kernel32" (ByVal wFlags As Long, ByVal dwBytes
As Long) As Long

Private Declare Function GlobalFree Lib "kernel32" (ByVal hmem As Long) As Long

Private Declare Function mmioRead Lib "winmm.dll" (ByVal hmmio As Long, pch As Any,
ByVal cch As Long) As Long

Private Declare Function waveOutPrepareHeader Lib "winmm.dll" _
  (ByVal hWaveOut As Long, lpWaveOutHdr As WAVEHDR, ByVal uSize As Long) As Long

Private Declare Function waveOutWrite Lib "winmm.dll" _
  (ByVal hWaveOut As Long, lpWaveOutHdr As WAVEHDR, ByVal uSize As Long) As Long

Private Declare Function waveOutSetPitch Lib "winmm.dll" _
  (ByVal hWaveOut As Long, ByVal dwPitch As Long) As Long

Private Declare Function waveOutSetPlaybackRate Lib "winmm.dll" _
  (ByVal hWaveOut As Long, ByVal dwRate As Long) As Long

Private Declare Function waveOutSetVolume Lib "winmm.dll" _
  (ByVal uDeviceID As Long, ByVal dwVolume As Long) As Long

Private Declare Function waveOutReset Lib "winmm.dll" (ByVal hWaveOut As Long) As
Long
```

```
rivate Type waveFormat
  wFormatTag As Integer
  nChannels As Integer
  nSamplesPerSec As Long
  nAvgBytesPerSec As Long
  nBlockAlign As Integer
nd Type

rivate Type WAVEHDR
  lpData As Long
  dwBufferLength As Long
  dwBytesRecorded As Long
  dwUser As Long
  dwFlags As Long
  dwLoops As Long
  lpNext As Long
  Reserved As Long
End Type

Private Sub DrawGrid()
Dim i As Long
   For i = 0 To Picture1.ScaleWidth Step 200
      Picture1.Line (i, 0)-(i, Picture1.ScaleHeight), QBColor(7)
      Picture1.Line (0, i)-(Picture1.ScaleWidth, i), QBColor(7)
   Next i
End Sub

Private Sub Picture1_Click()
  Set Picture1.Picture = Nothing
  Picture1.Refresh
End Sub

Private Sub Slider1_Click()
  Label4 = Slider1.Value / 10
End Sub

Private Sub Text1_GotFocus(Index As Integer)
   Text1(Index).SelStart = 0
   Text1(Index).SelLength = Len(Text1(Index).Text)
End Sub
```