Conception and implementation of a test system for automated tests of audio outputs with the method of acoustic fingerprinting

by

Prof. Dr.-Ing. Klaus Solbach
University of Duisburg-Essen
Department of
Microwave and RF-Technology

Ali Sen
University of Duisburg-Essen
Department of
Microwave and RF-Technology

Dipl.-Ing. Michael Reiner
Nokia Automotive Business Line
R&D Center Bochum

Contents

- Introduction
- Overview of the Test System
- Development Process
- Audio Fingerprinting Theory
- Function of Audio Fingerprinting
- Implementation in Nokia Test Tool
- Result of Tests with a Car kit
- Conclusion

Introduction

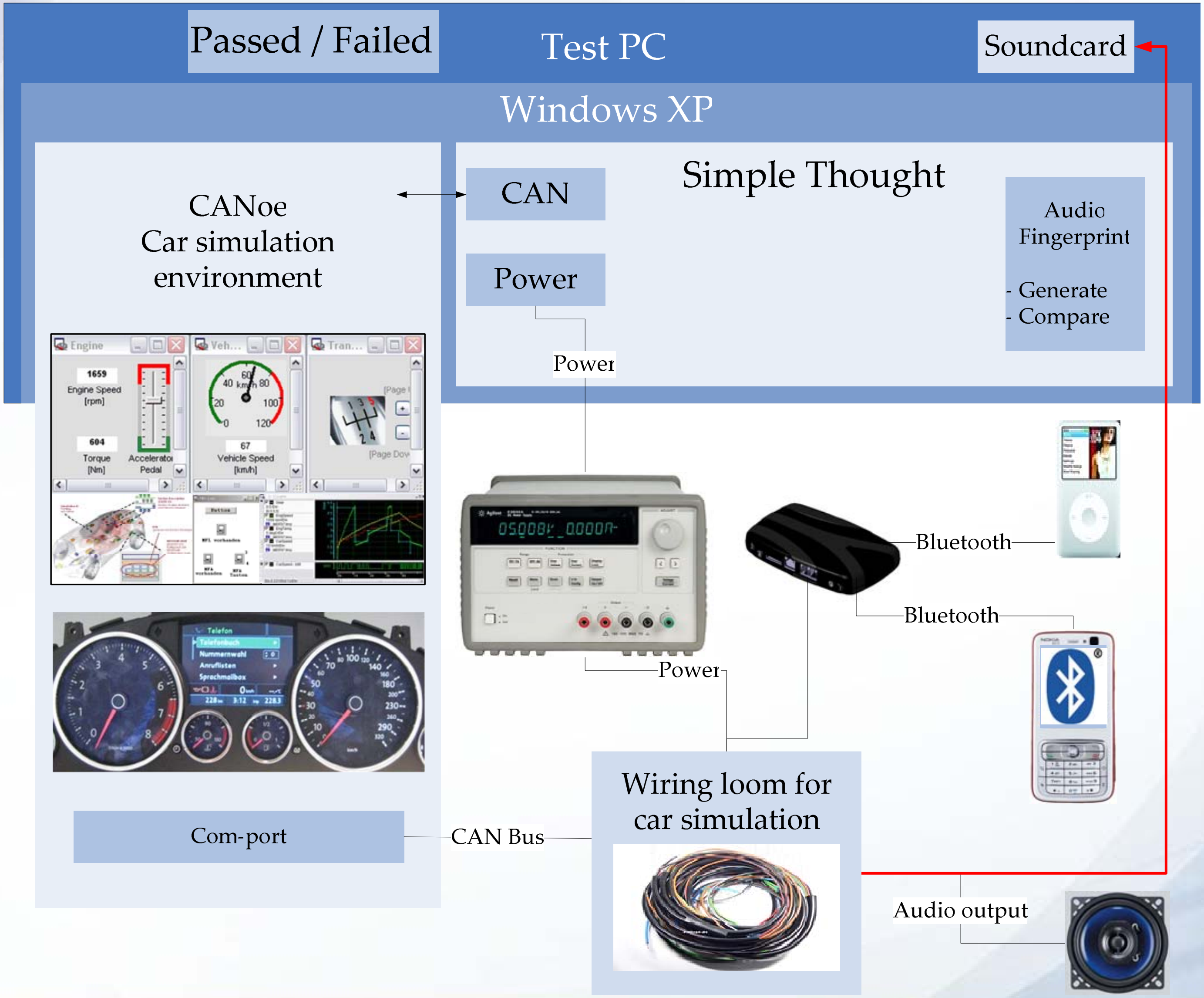
What is an audio fingerprint ?

- Small distinguishing feature of an audio signal
- Independent from artist or song name (Metadata)
- Consists of the main perceptual properties of the audio signal

What are the requirements for audio fingerprint comparison:

- Only the plain audio is needed
- Reliability
- Real-time audio comparison
- Support for short audio samples
- Support for both music and speech commands
- Limited implementation effort

Overview of the Test System



Passed / Failed

Test PC

Soundcard

Windows XP

CANoe
Car simulation
environment

CAN

Power

Simple Thought

Audio
Fingerprint

- Generate
- Compare

Power

Bluetooth

Bluetooth

Power

Com-port

CAN Bus

Wiring loom for
car simulation

Audio output

UNIVERSITÄT DUISBURG ESSEN

NOKIA

Development Process

First stage of development:

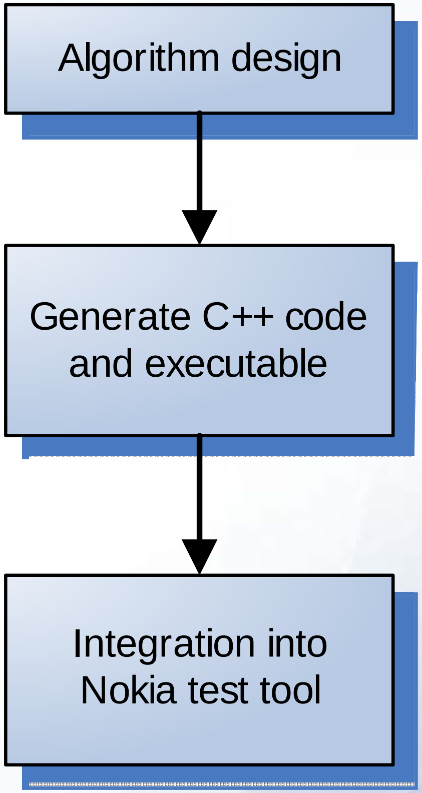
- Used existing fingerprint theory and create fingerprint program in “MATLAB “

Second stage of development:

- Use “MATLAB” automatic C++ code generation and implement the fingerprint program in “Visual Studio”

Last stage of development:

- Implementation of the fingerprint program in “Nokia test tool”


Algorithm design
Generate C++ code and executable
Integration into Nokia test tool

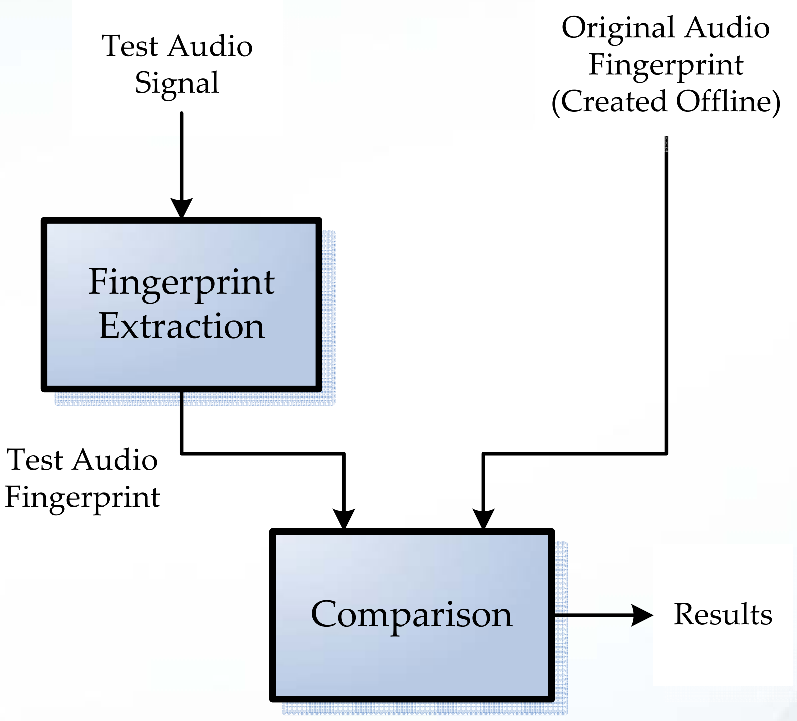
Contents

- Introduction
- Overview of the Test System
- Development Process
- Audio Fingerprinting Theory
    - Music Identification using Audio Fingerprinting
    - Overview of the Fingerprint Algorithm
    - The Pro and Cons
- Function of Audio Fingerprinting
- Implementation in Nokia Test Tool
- Result of Tests with a Car kit
- Conclusion

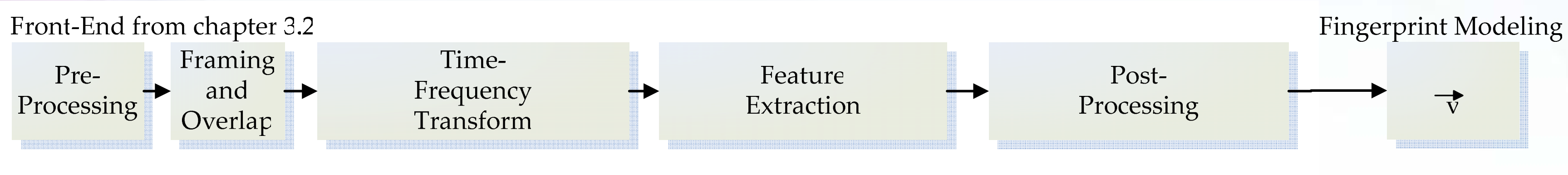
Music Identification using Audio Fingerprinting

Music identification process:

1. Create audio fingerprint from audio file
2. Read second audio signal from soundcard and extract fingerprint
3. Compare both fingerprints



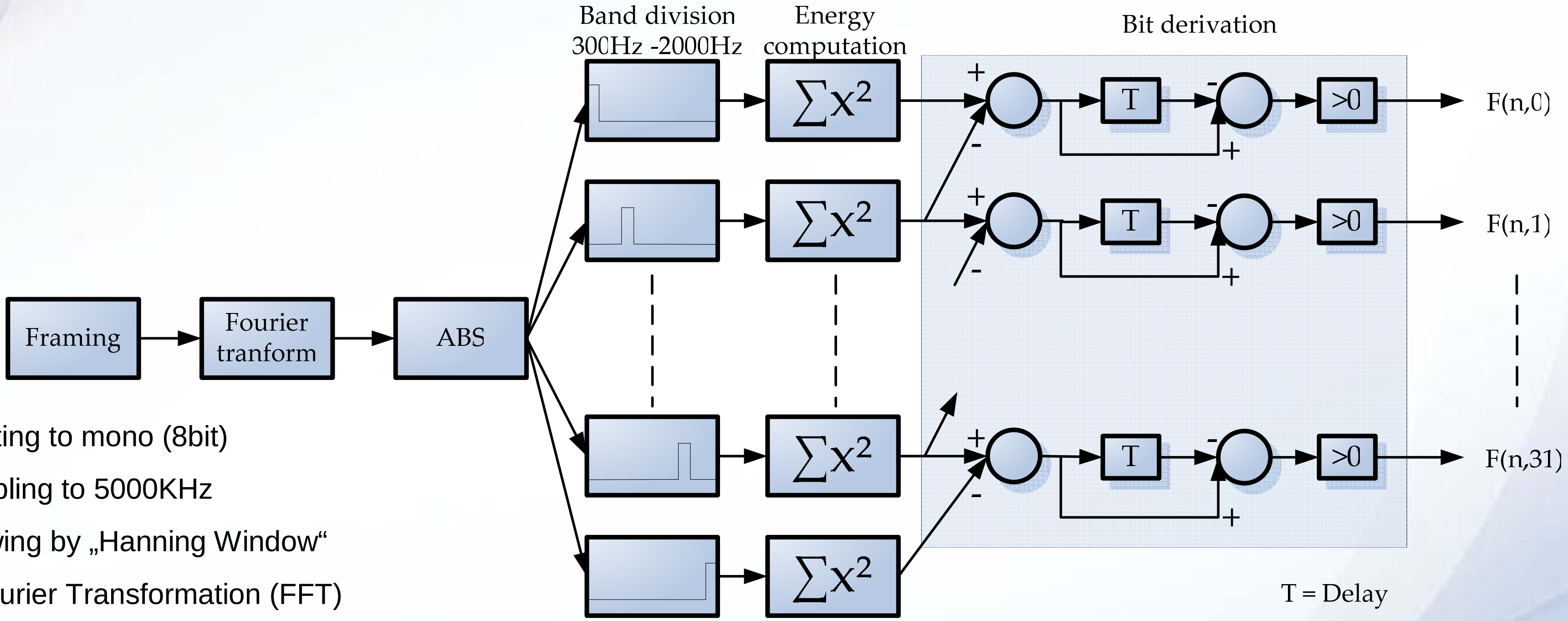
Overview of the Fingerprinting Algorithm



Front-End from chapter 3.2

Pre-Processing → Framing and Overlap → Time-Frequency Transform → Feature Extraction → Post-Processing → $\vec{v}$

Fingerprint Modeling



1. Converting to mono (8bit)
2. Resampling to 5000KHz
3. Windowing by „Hanning Window“
4. Fast Fourier Transformation (FFT)
5. Absolute value (ABS) (Amplitude)
6. Divide the generated spectrum into Bark bands
7. Energy computation and energy differences

The Pros and Cons

Disadvantages:

- Reacting to single sinusoidal sounds and DTMF tones

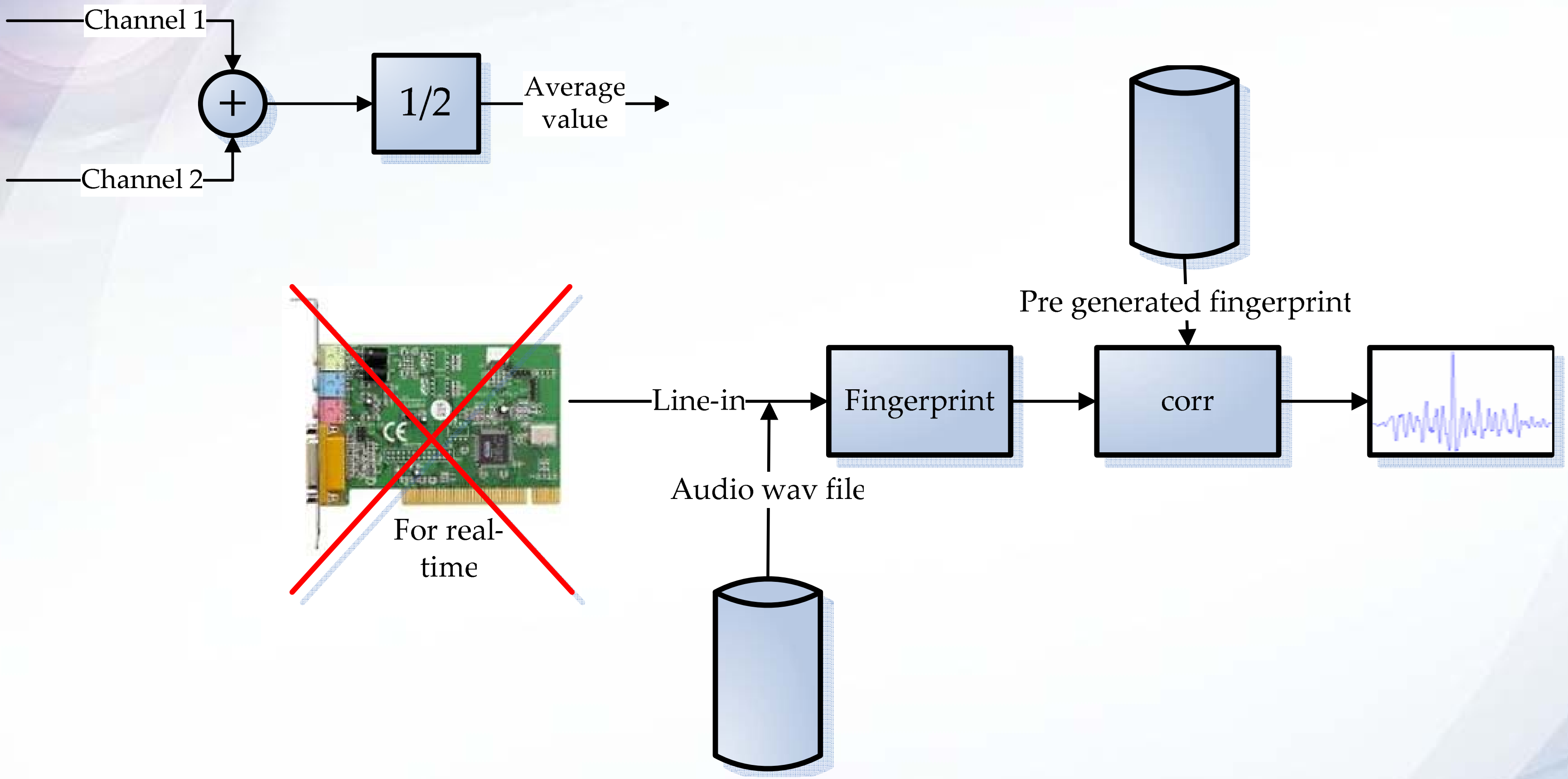
Advantages of this fingerprinting scheme:

- Adapted for all types of audio signals (speech, music, etc.)
- High accuracy regardless of small disturbance in recordings
- Independent from amplitude of the audio signal
- Easy “MATLAB”-implementation
- Good descriptive documents

Contents

- Introduction
- Overview of the Test System
- Development Process
- Audio Fingerprinting Theory
- Function of Audio Fingerprinting
  - Optimisation of Audio Fingerprint Comparison
  - Results of Comparison with different Settings
  - Audio Fingerprint Comparison I
  - Real-time Audio Matching
  - Possible Applications
  - DTMF Tones
  - DTMF Tone Detection
- Implementation in Nokia Test Tool
- Result of Tests with a Car kit
- Conclusion

Optimisation of Audio Fingerprint Comparison



Results of Comparison with Different Settings

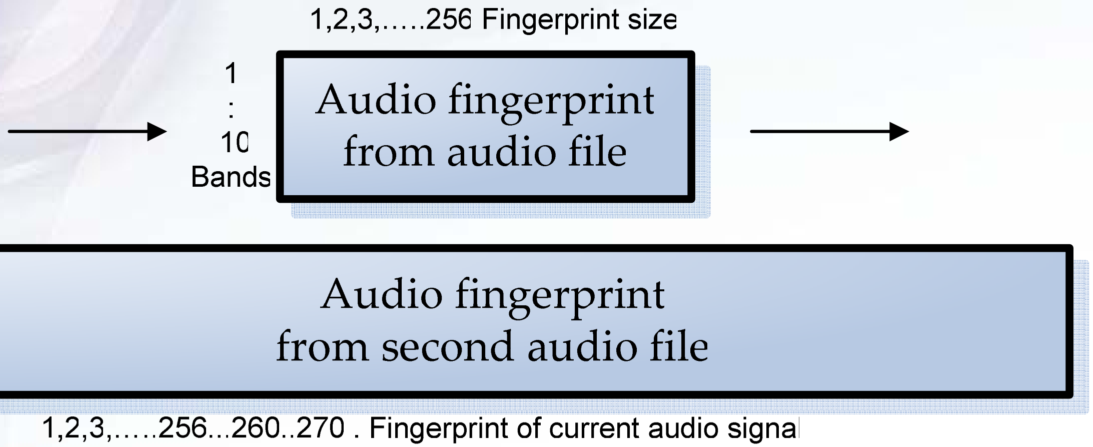
Final result:

Audio samples Settings	Speech commands	Music's like Pop or Techno	Classics	Sinusoidal tones	DTMF tones
Reference	92.13%	95.83%	98.50%	1.37%	3.65%

With different setup:

Audio samples Settings	Speech commands	Music's like Pop or Techno	Classics	Sinusoidal tones	DTMF tones
Reduce fingerprint size (N = 128)	92.13%	95.96%	98.96%	2.15%	2.47%
Increase fingerprint size (N = 384)	92.13%	96.05%	98.00%	1.82%	4.95%
Sample rate 3000 Hz	83.81%	84.62%	86.02%	1.52%	1.89%
Sample rate 4000 Hz	94.53%	94.01%	97.27%	2.91%	2.99%
Sample rate 7500 Hz	89.49%	1.80%	92.90%	3.15%	3.65%
Window size 1024	89.68%	92.71%	96.88%	2.65%	3.65%
Window size 4096	95.91%	95.96%	98.44%	3.56%	1.56%
Overlapping factor 32	92.42%	96.35%	98.63%	2.55%	2.15%
Overlapping factor 128	76.52%	79.21%	80.15%	5.54%	5.94%
Hamming window	92.70%	95.57%	98.57%	2.15%	1.56%
Blackman window	89.33%	95.31%	98.44%	1.22%	1.82%
Triangle window	91.76%	91.08%	97.85%	1.78%	3.78%
Only 10 bands	91.01%	95.75%	98.70%	0.95%	1.82%
Only 5 bands	85.96%	92.19%	93.75%	2.99%	3.32%
Vector elements 1 & -1	96.05%	96.86%	98.98%	3.29%	3.68%

Audio Fingerprint Comparison I



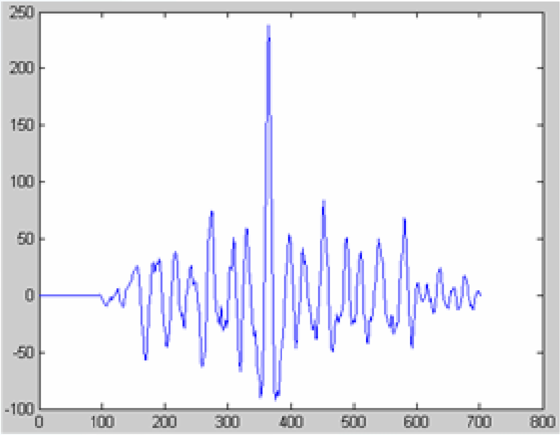
1,2,3,…..256 Fingerprint size

1
:
10
Bands

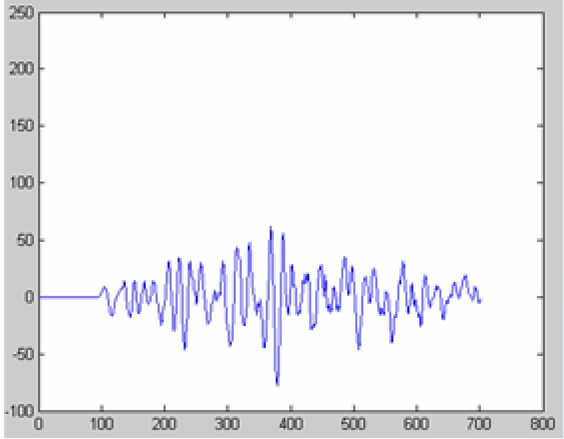
Audio fingerprint
from audio file

Audio fingerprint
from second audio file

1,2,3,…..256...260..270 . Fingerprint of current audio signal



True



False

Preparations of the audio file to improve the comparison results:

Audio files with silent ending:

Delete the silent part at the beginning and end of audio file

Extend to short audio files:

After resampling the window size of a short audio file is under 2048 of window size

Extract fingerprint at each part:

For tests with forwarding

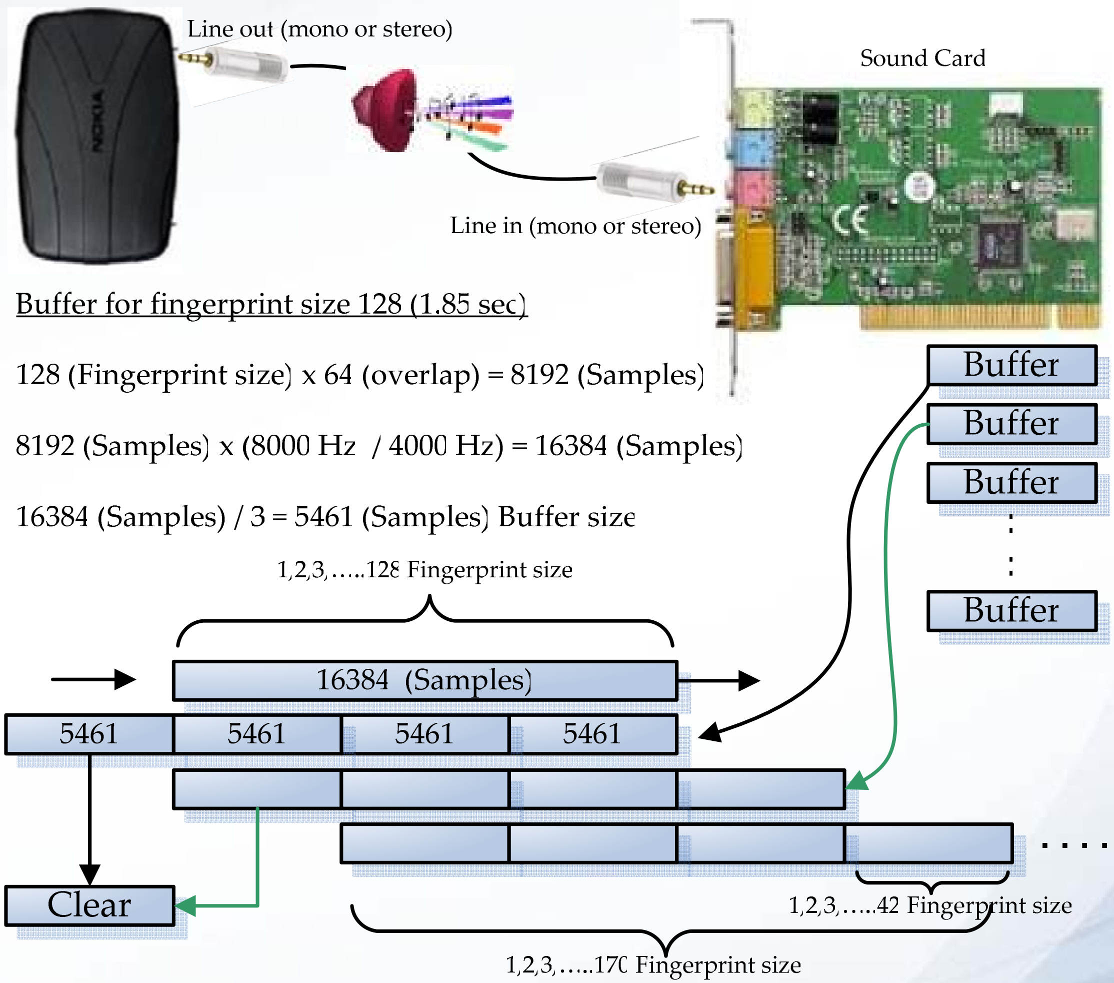
Real-time Audio Matching

Three different settings for real-time audio matching:

1. **Best setting:**
   Fingerprint size = 256
   (Need higher CPU speed)

2. **Optimal settings:**
   Fingerprint size = 128

3. **Fastest settings:**
   Fingerprint size = 128
   (Only 5 bands for reduce the buffer)


Line out (mono or stereo)
Sound Card
Line in (mono or stereo)

Buffer for fingerprint size 128 (1.85 sec)

128 (Fingerprint size) x 64 (overlap) = 8192 (Samples)

8192 (Samples) x (8000 Hz / 4000 Hz) = 16384 (Samples)

16384 (Samples) / 3 = 5461 (Samples) Buffer size

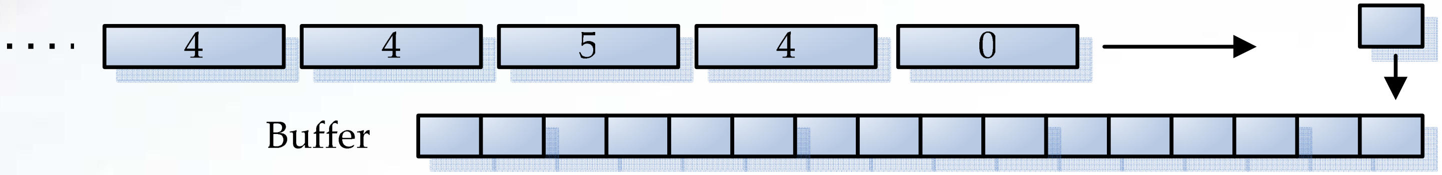
Buffer
Buffer
Buffer
Buffer
1,2,3,…..128 Fingerprint size
16384 (Samples)
5461
5461
5461
5461
Clear
. . . .
1,2,3,…..42 Fingerprint size
1,2,3,…..170 Fingerprint size

Buffer for fingerprint size 256 (3.6 sec) = 8192 (Samples) Buffer size

Possible Applications:

- Validation of voice recordings (no voice recognition!)

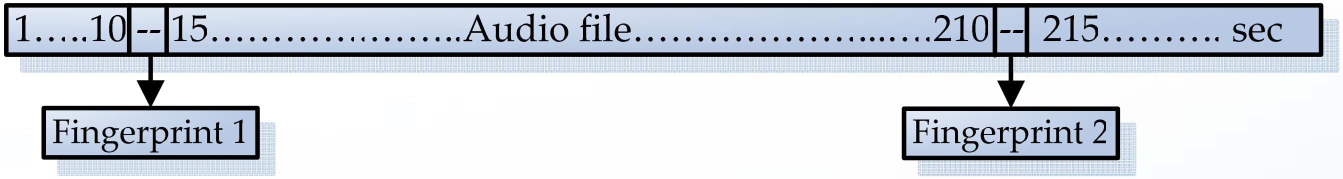
- Comparison of voice commands like phone numbers (0454418156)



· · · · 4 4 5 4 0

Buffer

- Multiple fingerprints for one audio file (different positions) to check different sound playback features (e.g. fast forward)



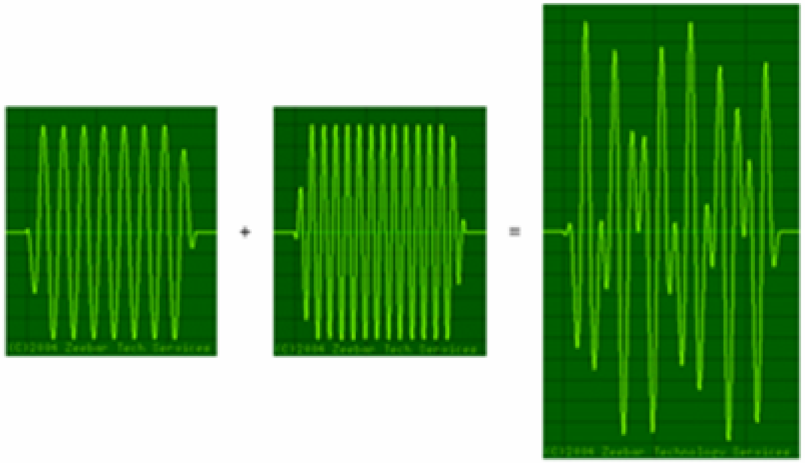
1…..10 -- 15………….……..Audio file…………………...….210 -- 215………. sec

Fingerprint 1

Fingerprint 2

- Search audio in database

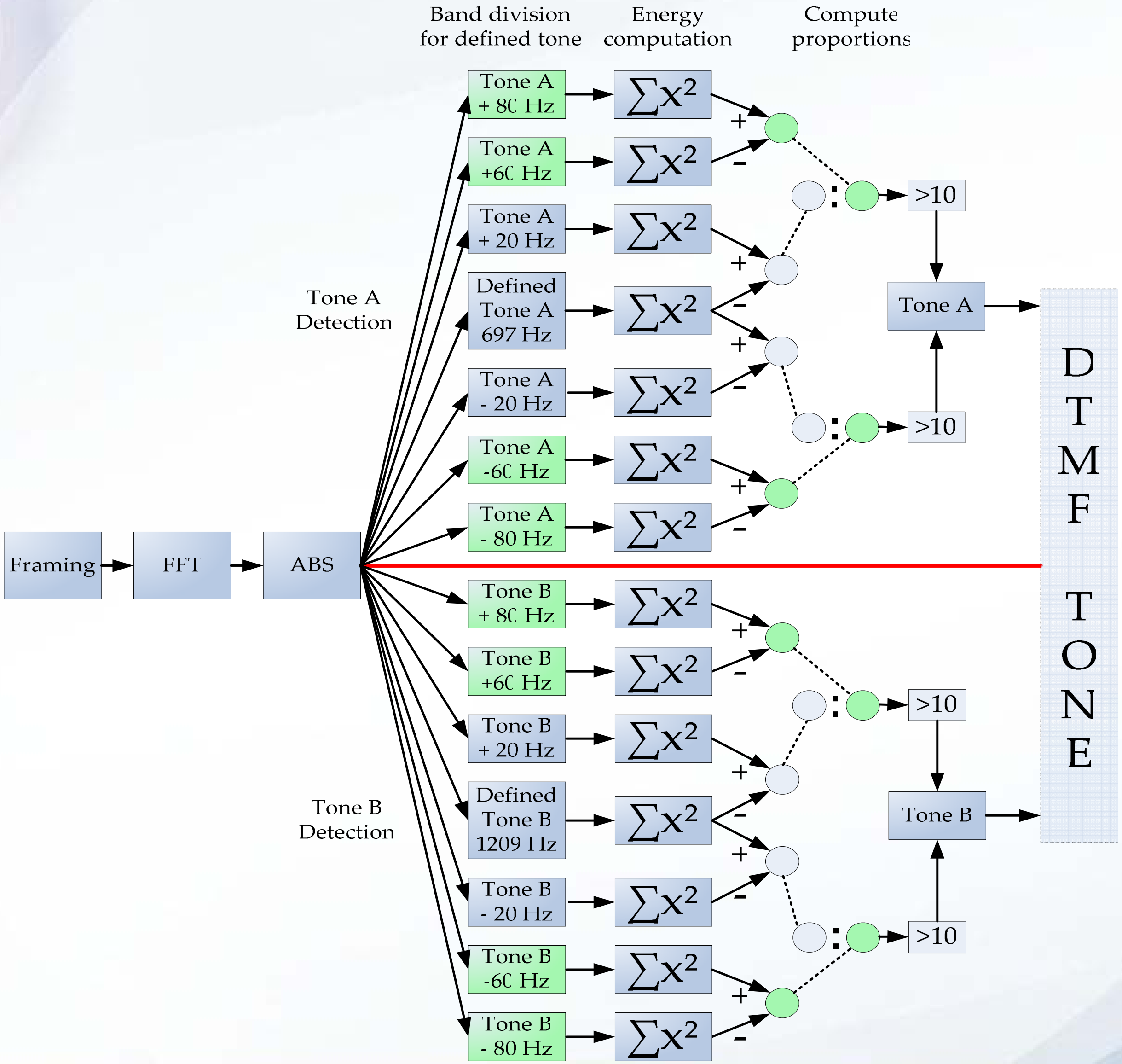
DTMF Tones



697 Hz Sine Wave + 1209 Hz Sine Wave = DTMF Tone "1"

DTMF keypad frequencies (with sound clips)				
	1209 Hz	**1336 Hz**	**1477 Hz**	**1633 Hz**
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

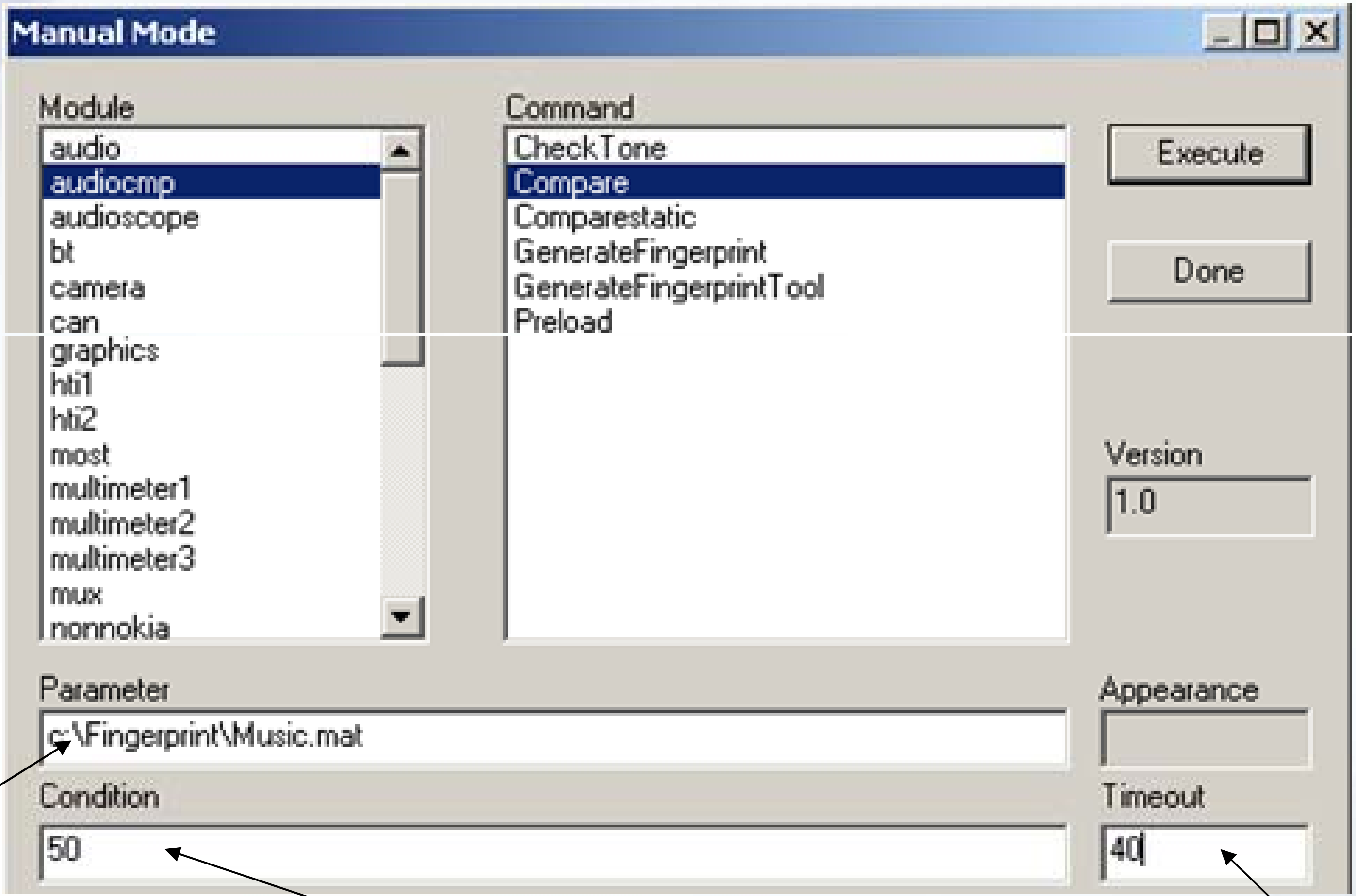
DTMF Tone Detection



Contents

- Introduction
- Overview of the Test System
- Development Process
- Audio Fingerprinting Theory
- Function of Audio Fingerprinting
- Implementation in Nokia Test Tool
    - Fingerprint Example in Manual Mode
    - Test Script
- Result of Tests with a Car kit
- Conclusion

Fingerprint example in Manual Mode



Version: 1.0

Parameter: c:\Fingerprint\Music.mat

Appearance:

Condition: 50

Timeout: 40

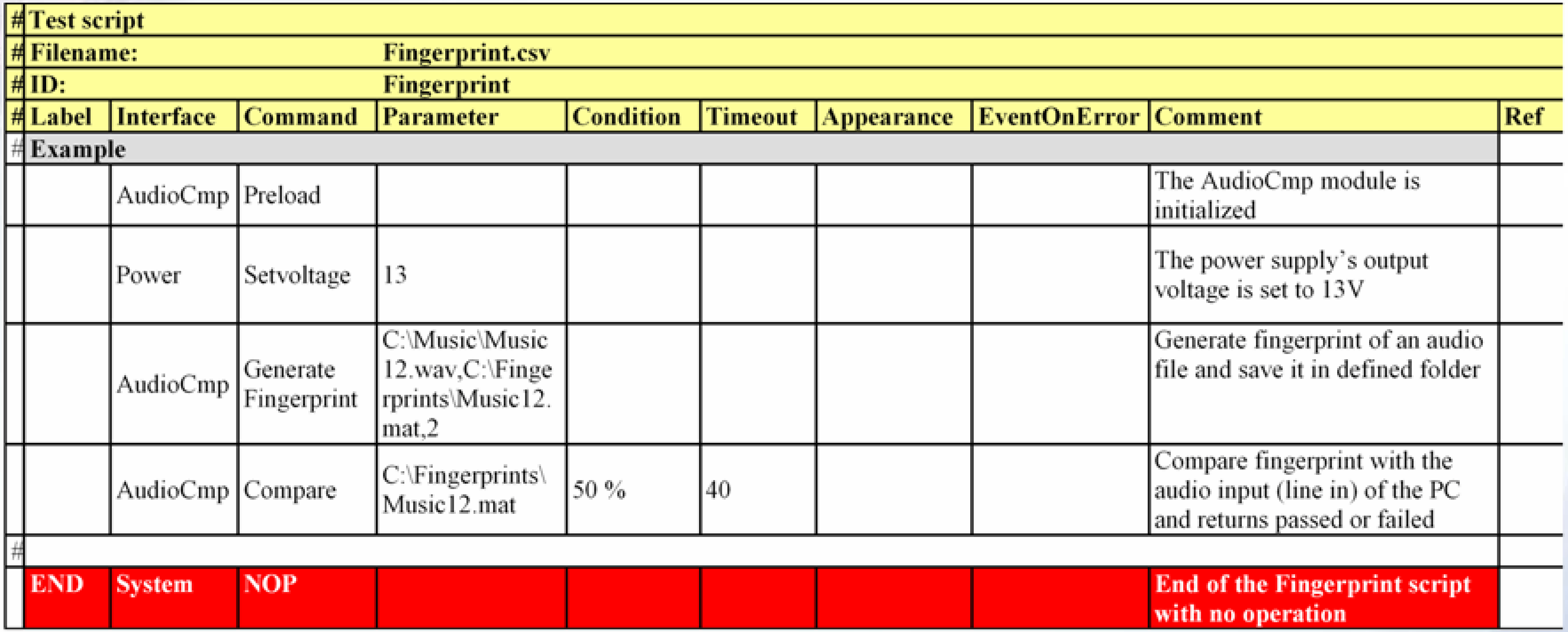
Fingerprint file

Quality settings

Duration time in sec

Test Script

Test script:



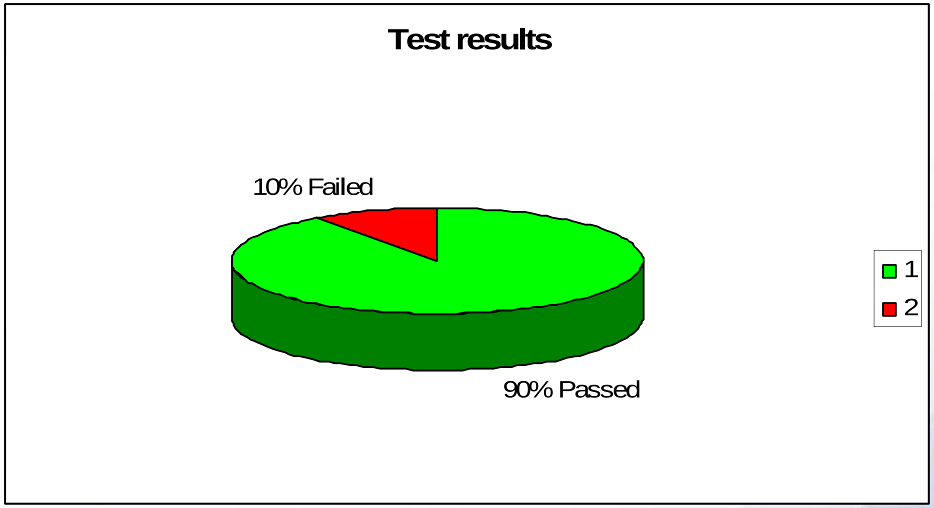
#	Test script									
#	Filename:			Fingerprint.csv						
#	ID:			Fingerprint						
#	Label	Interface	Command	Parameter	Condition	Timeout	Appearance	EventOnError	Comment	Ref
#	Example									
		AudioCmp	Preload						The AudioCmp module is initialized	
		Power	Setvoltage	13					The power supply's output voltage is set to 13V	
		AudioCmp	Generate Fingerprint	C:\Music\Music 12.wav,C:\Fingerprints\Music12.mat,2					Generate fingerprint of an audio file and save it in defined folder	
		AudioCmp	Compare	C:\Fingerprints\Music12.mat	50 %	40			Compare fingerprint with the audio input (line in) of the PC and returns passed or failed	
#										
	END	System	NOP						End of the Fingerprint script with no operation	

Contents

- Introduction
- Overview of the Test System
- Development Process
- Audio Fingerprinting Theory
- Function of Audio Fingerprinting
- Implementation in Nokia Test Tool
- Result of Tests with a Car kit
- Conclusion

Result of Tests with a Car kit

Recorded audio samples \ Original audio samples	Techno	Pop	Classic
Techno	93.13%	8.37%	9.55%
Pop	11.89%	95.03%	10.36%
Classic	13.53%	7.46%	98.58%



Tests:

- Play, Stop, Pause
- Repeat
- Intro scan
- Skip track
- Shuffle
- Fast forward and fast backward

Contents

- Introduction
- Overview of the Test System
- Development Process
- Audio Fingerprinting Theory
- Function of Audio Fingerprinting
- Implementation in Nokia Test Tool
- Tests with the Audio Identification System
- Conclusion

Conclusion

- Fingerprint can be extracted from audio file
- Audio signal can be identified
- Similarity rate of almost 90%
- Identification rate of exact 100%.
- Stability of the developed test system is given
- To advance a search and record function can be added in future

Questions ?!

Questions ?!?

Questions ?!?

Questions ?!?

Thank you for your attention!