08-6341-RN-ZCH66
NOV 29, 2007
1.1

Release Notes for MP3 Encoder on ARM11&ARM9 ELINUX

ABSTRACT:

Release Notes for MP3 Encoder on ARM11&ARM9 ELINUX

KEYWORDS:

Multimedia codecs, MP3, Audio

APPROVED:

Shang Shidong

Revision History

Version	Date	Author	Change Description
0.1	29-Nov-2007	Huang Shen	release 0.1
1.0	17-Jun-2008	Huang Shen	Add arm9 related support
1.1	31-Mar-2009	Wang Shengjiu	Update linux build

Table of Contents

Introduction

1.1 Purpose

The purpose of this document is to provide information on the package contents, instructions on building library and test applications and test execution on ARM11&ARM11 ELINUX, RVDS and Linux x86.

1.2 Scope

The scope is restricted to information on the package contents and instructions for building and testing. This document does not provide architecture or details about the APIs provided in the package. Performance data will be provided in another document as detailed in the Requirements Book.

1.3 Audience Description

The reader is expected to have basic understanding of Audio Signal processing and MP3 encoding.

1.4 References

1.4.1 Standards

- N/A

1.4.2 General References

- N/A

1.4.3 Freescale Multimedia References

- MP3 Encoder Application Programming Interface - mp3_enc_api.doc
- MP3 Encoder Requirements Book - mp3_enc_reqb.doc
- MP3 Encoder Test Plan - mp3_enc_test_plan.doc
- MP3 Encoder Release notes – mp3_enc_release_notes.doc
- MP3 Encoder Test Results – mp3enc_test.xls
- MP3 Encoder Interface header – mp3_enc_interface.h
- MP3 Encoder Application Code - test_mp3enc.c

1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
API	Application Programming Interface
ARM	Advanced RISC Machine
FSL	Freescale
OS	Operating System
RVDS	ARM RealView Development Suite
TBD	To Be Determined
UNIX	Linux PC x/86 C-reference binaries

1.6 Document Location

docs/mp3_enc

2 Release History

RELEASE NUMBER	DELIVERABLES	FEATURES
1.2	• Header file for the encoder (mp3_enc_interface.h) • Example Application (Name: test_mp3enc.c) • ELINUX and RVDS libraries and test applications • UNIX/Linux x/86 Reference library and test application • Makefiles and Source code for library and test application including optimized assembler for the ELINUX and RVDS libraries. • Project files for RVDS and ELINUX	• Supports sampling rate of 32, 44.1, 48KHz • Supports bit rates of 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320 kbps. • Stereo / Mono encoding.
	•	•
	•	

Table 1. Details of the release

2.1 Assumptions and Known Problems

None

2.2 Contacts

Please report any problems to Freescale customer representative

3 List of Deliverables

3.1 Documentation

Base directory: /fsl_mad_multimedia_codec/

Subdirectory	Files
docs/mp3_enc	mp3_enc_api.doc mp3_enc_reqb.doc mp3_enc_test_plan.doc mp3enc_test.xls mp3_enc_release_notes.doc

3.2 Public Headers

Base directory: /fsl_mad_multimedia_codec/ghdr/

Subdirectory	File
API_include	mp3_enc_interface.h

3.3 Test Application Source

Base directory: /fsl_mad_multimedia_codec/test/mp3_enc/

Subdirectory	Files
.	"Makefile" makefile for building RVDS, UNIX and ELINUX board executable.
c_src	*.c, application code.
hdr	*.h, application header files

3.4 Library Source

Base directory: /fsl_mad_multimedia_codec/src/mp3_enc/

Subdirectory	Files
.	Makefile "Makefile" for building RVDS, ELINUX libraries. lib_mp3_enc_arm11_lervds.a – Special options for simulator testing lib_mp3_enc_arm11_elinux.a - static library for board lib_mp3_enc_arm11_elinux.so – shared library for board
c_src	*.c, MP3 encoder source code
asm_arm	*.s assembly source
hdr	*.h, MP3 encoder library header files

3.5 Common Makefiles

Base Directory: /fsl_mad_multimedia_codec/build/

Subdirectory	Files
Makefile.init	This is a common makefile included in the codec library makefile for building the libraries. This file includes common options used by all codecs. Following flags can be overwritten or added to in the codec library makefile 1. Path to toolchain tools (TC_ROOT) 2. GNU header file path (HEADER_PATHS) 3. GNU library path (LIB_PATHS) 4. GNU Compiler/Assembler Options (GNU_CFLAGS, GNU_AFLAGS) 5. Endian Flags 6. Optimization Flags(OPTIM_LEVEL, OPTIM_TYPE) 7. Common options for RVDS,UNIX and ELINUX (CFLAGS,AFLAGS) 8. Build specific flags 9. Source directory of 'C' code 10. Source directory of 'assembly(.s)' code 11. Object directory for .o files 12. RVDS Compilation Tools 13. Codec header path 14. Arguments for librarian for UNIX builds 15. SHARED_ELINUX builds for libraries that must be linked using the toolchain because of external library includes.
Makefile_test.init	This is the common makefile included in the codec test makefile for building the test application. This file includes the common options used by the all the codecs. Following flags can be overwritten or added to in the codec test makefile 1. Toolchain path depending on the build option 2. Compiler Flags 3. Linker flags 4. Paths for c_source, exe and object directories 5. Codec header files' INCLUDES path 6. Endian Flags 7. CODEC_LIB generation

3.6 Test Vectors

N/A

4 Software Setup & Tools used

- ARM RVDS 3.0 should be installed in the PC.
- Freescale Linux OS Release must be running on the evaluation board.
- Intel based Red Hat Linux Machine must have the Montavista toolchain installed on it.
    - MontaVista 3.4.3-25.0.36.0501313 2005-08-21
- 'Cygwin' **Version** CYGWIN_NT-5.1, a freely downloadable linux emulator is installed in PC - **http://www.cygwin.com/**.
- 'make' utility available for targeted platforms

5 Build Procedure

All the required makefiles are provided under individual directories. The library can be built for windows / target processor (ARM1136J-S and ARM926E). The details for the build procedure are described below.

5.1 Library

To build the library, run 'make' on 'Makefile' from library directory. The makefile shall create the required directory to hold the object files. The makefile can be used if you want to build the library only. The same makefile can used to build libraries for both board, Linux and RVDS with different build options. The following options are available to build the library.

Options

a) **BUILD options**:

a. **BUILD= ARM11ELINUX** : This option builds static library 'lib_mp3_enc_arm11_elinux.a' for testing on the board.

b. **BUILD= ARM11ELINUXDLIB** : This option builds shared library 'lib_mp3_enc_arm11_elinux.so' for testing on the board.

c. **BUILD=ARM11LERVDS**: This option builds the static library 'lib_mp3_enc_arm11_lervds.a', for testing on RVDS (Armulator).

d. **BUILD=ARM9ELINUX:** This option builds static library 'lib_mp3_enc_arm9_elinux.a' , for testing on the board.

e. **BUILD=ARM9ELINUXDLIB:** This option builds shared library 'lib_mp3_enc_arm9_elinux.so' for testing on the board.

f. **BUILD=ARM9LERVDS:** This option builds the static library 'lib_mp3_enc_arm9_lervds.a', for testing on RVDS (Armulator).

Eg:

```
make BUILD=ARM11ELINUX
make BUILD=ARM11ELINUXDLIB
make BUILD=ARM11LERVDS
make BUILD=ARM9ELINUX
make BUILD=ARM9ELINUXDLIB
make BUILD=ARM9LERVDS
```

b) **clean options**:

- **clean_ARM11LERVDS**: Deletes all the object files and the RVDS library 'lib_mp3_enc_arm11_lervds.a'.
- **clean_ARM11ELINUX**: Deletes all the object file and the ELINUX libraries lib_mp3_enc_arm11_lervds.a and lib_mp3_enc_arm11_elinux.so.
- **clean**: Deletes all the object files and RVDS, ELINUX libraries.

Note: Make appropriate changes in file 'Makefile.init' at directory 'fsl_mad_multimedia_codec/build' for the location of toolchains.

The library that is built is saved as lib_mp3_enc_arm11_lervds.a or lib_mp3_enc_arm9_lervds.a for RVDS build, and lib_mp3_enc_arm11_elinux.so or lib_mp3_enc_arm9_elinux.so for board build. These libraries are saved in the current directory (the same directory in which the source and assembly directories are listed).

Target	Compilation Environment	Build Options	Library Name
Board	PC (Using Cygwin)	BUILD= ARM11ELINUXDLIB	lib_mp3_enc_arm11_elinux.so
Board	PC (Using Cygwin)	BUILD= ARM9ELINUXDLIB	lib_mp3_enc_arm9_elinux.so
RVDS	PC (Using Cygwin)	BUILD=ARM11LERVDS	lib_mp3_enc_arm11_lervds.a
RVDS	PC (Using Cygwin)	BUILD=ARM9LERVDS	lib_mp3_enc_arm9_lervds.a

5.2 Test Application

To build the test application, run 'make' on 'Makefile' from the test directory. This makefile can create executables for testing on both board and RVDS for ARM11&ARM9. The executables are stored under current directory. The makefile shall create the required directory structure to hold the object files and executables. The following commands should be invoked so as to build the executables.

Options

1) **BUILD options**:
   - **BUILD=ARM11ELINUX**: This is the default option and builds the executable 'test_mp3_enc_arm11_elinux', for the board.
   - **BUILD=ARM11LERVDS**: This option builds the executable 'test_mp3_enc_arm11_lervds' for the RVDS (Armulator).
   - **BUILD=ARM9ELINUX:** This is the default option and builds the executable 'test_mp3_enc_arm9_elinux', for the board.
   - **BUILD=ARM9LERVDS:** This option builds the executable 'test_mp3_enc_arm9_lervds' for the RVDS (Armulator).

   Eg: make BUILD=ARM11ELINUX (for board)

make BUILD=ARM11LERVDS (for Armulator)
make BUILD=ARM9ELINUX (for board)
make BUILD=ARM9LERVDS (for Armulator)

2) **clean options**:
   - **clean_ARM11LERVDS**: Deletes all the object files and the RVDS executable 'test_mp3_enc_arm11_lervds' .
   - **clean_ARM11ELINUX**: Deletes all the object file and the ELINUX 'test_mp3_enc_arm11_elinux'.
   - **clean**: Deletes all the object files and RVDS, ELINUX executables.

Note:
In 'Makefile_test.init' at directory 'fsl_mad_multimedia_codec/build', the paths for the compiling and linking tools are hard coded for the current set-up. These paths may not be the same in the user's directory set up. Hence, the 'Makefile_test.init' should be modified to point to the directories where the linking and compilation tools are present before building the application for board.

The following table summarises the build options,

Target	Compilation Environment	Build Options	Executable Name
Board	Redhat Linux Machine	BUILD=ARM11ELINUX	test_mp3_enc_arm11_elinux
Board	Redhat Linux Machine	BUILD=ARM9ELINUX	test_mp3_enc_arm9_elinux
RVDS	PC (Using Cygwin)	BUILD=ARM11LERVDS	test_mp3_enc_arm11_lervds
RVDS	PC (Using Cygwin)	BUILD=ARM9LERVDS	test_mp3_enc_arm9_lervds

6 Test Application Execution

Test applications for ARM11 and ARM9 share the same input parameters. Below we use arm11 test application as an example.

6.1 Scripts

N/A

6.2 ELINUX

test_mp3_enc_arm11_elinux ***[+m mode] [+b bitrate] [+s sample_rate] [+c config] [+f format] wavfile outfile***

test_mp3_enc_arm9_elinux ***[+m mode] [+b bitrate] [+s sample_rate] [+c config] [+f format] wavfile outfile***

mode: m = mono

j = jointstereo(default)

bitrate: The following bit rates are supported: 32, 40, 48, 56, 64, 80, 96,112, 128,160,192, 224, 256, 320 kbps. (default = 128)

sample_rate: The following sampling rates are supported: 32000, 44100 and 48000 Hz. (default=44100)

format: i = input in L/R interleaved format (default)

l = input in contiguous L samples, followed by contiguous R samples

config: q = Optimized for Quality (default)

s = Optimized for Speed

wavfile: Input wavfile

outfile: The encoded MP3 file.

6.3 RVDS

Please refer ARM documentation regarding loading the image and configuring the RVDS debugger for ARM1136J-S

- RVDS :
  Once the image is loaded press "*F5*" or select the pull down menu option "*Debug -> run*" to run the loaded image.

7 Pre compilation Options

The following C options need to be set

Compiler switches for Library

C Defines	Description	Remarks
LOG_DEBUG	Enable for log data.	Set to 1 to Enable data log. Must be closed in Released version

Compiler switches for Test Application

C Defines	Description	Remarks
TIME_PROFILE	Log performance results	ELINUX build only
MHZ_MEASURE	Log performance results	RVDS build only

Compiler switches common between Library and Test Application

C Defines	Description	Remarks