08-6532-RN-ZCH66

18/04/2008

1.2

Release Notes for SBC Encoder

ABSTRACT:

Release Notes for SBC Encoder

KEYWORDS:

Multimedia codecs, SBC

APPROVED:

Shang Shidong

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
1.0	31-August-07	Sunil Ramaswamy	Initial Version
1.1	27-March-08	Tao Jun	Update doc
1.2	18-Apr-08	Tao Jun	Update doc

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 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 A2DP and SBC encoding.

1.4 References

1.4.1 Standards

- *A2DP specification for Bluetooth, version 1.2*

1.4.2 Freescale Multimedia References

- SBC Encoder Application Programming Interface – sbc_enc_api.doc
- SBC Encoder Requirements Book – sbc_enc_reqb.doc
- SBC Encoder Release notes - sbc_enc_release_notes.doc
- SBC Encoder Test plan –sbc_enc_test_plan.doc
- SBC Encoder Test Summary – sbc_enc_test_results.doc
- SBC Encoder Interface Header – sbc_api.h

1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
API	Application Programming Interface
OS	Operating System

1.6 Document Location

docs/sbc_enc

.

2 Release History

RELEASE NUMBER	DELIVERABLES	FEATURES
1.0	• Documentation • Interface header file for encoder • ELINUX and RVDS libraries and test applications for encoder • UNIX/Linux x86 Reference library and test application • Makefiles and Source code for library and test application including optimized assembler for the ELINUX and RVDS libraries. • Test vectors	• Initial Release • Contains prototypes of interface function and data types • Details of feature and interface function can be found in these docs • Optimized C and assembly files • Contains standard test vectors. Sample application can be used to build executables
1.1	• Documentation • Interface header file for encoder • ELINUX and RVDS libraries and test applications for encoder • UNIX/Linux x86 Reference library and test application • Makefiles and Source code for library and test application including optimized assembler for the ELINUX and RVDS libraries.	• Optimized C code and assembly files based on ARM9E.

1.2	• Documentation • Interface header file for encoder • ELINUX and RVDS libraries and test applications for encoder • UNIX/Linux x86 Reference library and test application • Makefiles and Source code for library and test application including optimized assembler for the ELINUX and RVDS libraries.	• Optimized C code and assembly files based on both ARM9E and ARM11.

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: /multimedia_codecs/

Subdirectory	Files
docs/sbc_enc	sbc_enc_api.doc sbc_enc_reqb.doc sbc_enc_release_notes.doc sbc_enc_test_plan.doc sbc_enc_test_results.doc

3.2 Public Headers

Base directory: / multimedia_codecs/

Sibdirectory	Files	Description
ghdr/sbc	sbc_api.h sbc_typedefs.h sbc_defs.h	SBC encoder header files

3.3 Test Application Source

Base directory: / multimedia_codecs/

Subdirectory	Files
test/sbc_enc	"Makefile" makefile for building RVDS, UNIX and ELINUX board executables.
test/sbc_enc/hdr	*.h, application headers.
test/sbc_enc/c_src	*.c, application code.
utils/sbc_enc	Batch files to be run on the board

3.4 Library Source

Base directory: /multimedia_codecs/

Subdirectory	Files
src/sbc_enc	Makefile "Makefile" for building RVDS, UNIX, and ELINUX libraries. lib_sbc_enc_arm9_elinux.a: static library for ARM926EJ-S lib_sbc_enc_arm9_elinux.so: dynamic library for ARM926EJ-S lib_sbc_enc_arm9_lervds.a: ARM9 LERVDS library lib_sbc_enc_x86_unix.a : library for Linux x/86 – c reference

	code lib_sbc_enc_arm11_elinux.a: static library for ARM1136JF-S lib_sbc_enc_arm11_elinux.so: dynamic library for ARM1136JF-S lib_sbc_enc_arm11_lervds.a: ARM11 LERVDS library
src/sbc_enc/c_src	*.c, SBC encoder source code
src/sbc_enc/hdr	*.h SBC encoder library header files

3.5 Common Makefiles

Base Directory: / multimedia_codecs/

Makefile	**Description**
build/Makefile.init	This is a common makefile. To build libraries, it is included in the codec library makefile. This file includes common options used by all codecs.
build/ Makefile_test.init	This is the common makefile included in the codec test makefile building the test application. This file includes the common opti used by the all the codecs.

3.6 Test Vectors

Base Directory: TBD

The test vectors are provided in another location from the library and test source.

4 Software Setup & Tools used

- ARM RVDS 3.02 (build 586) should be installed in the PC.
- Freescale Linux OS Release L2.6.18.1 must be running on the evaluation board.
- Intel based Red Hat Linux Machine must have the GSO toolchain installed on it.
- ‘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 (ARM926EJ-S/ARM1136JF-S). The details for the build procedure are described below.

5.1 Library

To build the library, run 'make' on 'Makefile' from src/sbc_enc directory. This makefile can create libraries for testing on ARM board, RVDS, Linux and UNIX. 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 following options can be invoked so as to build the library

Options

a) **BUILD options**:

- **BUILD= ARM9ELINUX** : It builds both static as well as dynamic libraries, 'lib_sbc_enc_arm9_elinux.a ' and shared library 'lib_ sbc_enc_arm9_elinux.so' , for testing on the board.

- **BUILD=ARM9LERVDS**: This option builds the static library 'lib_sbc_enc_arm9_lervds.a', for testing on ARM9 LE RVDS (Armulator).

- **BUILD=UNIX**: This option builds the static library 'lib_sbc_enc_x86_unix.a', for testing on UNIX/Linux machine.

- **BUILD= ARM11ELINUX** : It builds both static as well as dynamic libraries, 'lib_sbc_enc_arm11_elinux.a ' and shared library 'lib_ sbc_enc_arm11_elinux.so' , for testing on the board.

- **BUILD=ARM11LERVDS**: This option builds the static library 'lib_sbc_enc_arm11_lervds.a', for testing on ARM11 LE RVDS (Armulator).

b) **clean options**:

- **clean**: Deletes all the object files and the library for specified BUILD option.

Note: Make appropriate changes in file 'Makefile.init' for the location of toolchains.

The libraries are saved in the current directory, release/lib.

Target	**Compilation Environment**	**Build Options**	**Library Name**
ELINUX	Unix/Linux machine	BUILD=ARM9ELINUX	lib_sbc_enc_arm9_elinux.a lib_sbc_enc_arm9_elinux so
ELINUX	Unix/Linux machine	BUILD=ARM11ELINUX	lib_sbc_enc_arm11_elinux.a lib_sbc_enc_arm11_elinux so

RVDS	Unix/Linux machine	BUILD=ARM9LERVDS	lib_sbc_enc_arm9_lervds.a
RVDS	Unix/Linux machine	BUILD=ARM11LERVDS	lib_sbc_enc_arm11_lervds.a
Unix/ Linux	Unix/Linux machine	BUILD=UNIX	lib_sbc_enc_x86 _unix.a

5.2 Test Application

To build the test application, run 'make' from the test/sbc_enc directory. This makefile can create executables for testing on Linux x86, the ARM9 board and RVDS for ARM9.. The following commands should be invoked so as to build the executables.

Options

1) **BUILD options**:
   - **BUILD=ARM9ELINUX**: This option builds the executable 'test_sbc_enc_arm9_elinux', for MXC91131 board.
   - **BUILD=ARM9LERVDS**: This option builds the executable 'test_sbc_enc_arm9_lervds ' for the ARM9 LE RVDS (Armulator).
   - **BUILD=UNIX**: This option builds the executable 'test_sbc_enc_x86_unix' for the Unix/Linux machine.
   - **BUILD=ARM11ELINUX**: This option builds the executable 'test_sbc_enc_arm11_elinux', for i.MX31 board.
   - **BUILD=ARM11LERVDS**: This option builds the executable 'test_sbc_enc_arm11_lervds ' for the ARM11 LE RVDS (Armulator).

2) **clean options**:
   - **clean:** Deletes all the object files and executable for the specified BUILD option

Note:
In 'Makefile_test.init', 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, it 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**
ELINUX	Redhat Linux Machine	BUILD=ARM9ELINUX	test_sbc_enc_arm9_elinux

ELINU X	Redhat Linux Machine	BUILD=ARM11ELINUX	test_sbc_enc_arm11_elinux
RVDS	Unix/Linux	BUILD=ARM9LERVDS	test_sbc_enc_arm9_lervds
RVDS	Unix/Linux	BUILD=ARM11LERVDS	test_sbc_enc_arm11_lervds
UNIX/ Linux	Unix/Linux machine	BUILD=UNIX	test_sbc_enc_x86_unix

6 Test Application Execution

6.1 Scripts

TBD

6.2 ELINUX SINGLE

test_sbc_enc_arm9_elinux/ test_sbc_enc_arm11_elinux < input_file> [-h] [-l<blk_len>] [-m<mode>] [-o<output_file>] [-n<subbands> [-p] [–r <rate>] [-b<bitpool>] [-s<sample_freq>] [-f<super_frame_size>]

input_file	***Specify input file***
input_file	***Specify input file***
-h	***Display this command line help and exit***
-l <blk_len>	***block length (4,8,12 or 16)***
-m <mode>	***mode (0=mono, 1=dual_channel, 2=stereo)***
-o output_file	***Specify output file. Output suppressed if unspecified.***
-n <subbands>	***number of subbands (4 or 8)***
-p	***Enable psycho-acoustic model [default is off]***
-r <rate>	***Bitrate in bps (cannot be combined with -b option)***
-b <bitpool>	***Bitpool value (2 to 250) (cannot be combined with -r option)***
	Recommended bitpool value:
	In DUAL MODE, if sample rate==16KHz, bitpool=5*subbands,
	if sample rate>16KHz,bitpool=4*subbands;
	In STEREO MODE, if sample rate==16KHz, bitpool=9*subbands,
	if Samperate>16kHz,bitpool=7*subbands.
-s <sampling_freq>	***Sampling Frequency (16000,32000,44100, or 48000 Hz)***
-f <super_frame_size>	***Size of input data frame (e.g. 2048) - optional***

NOTE: 1) Either bitpool or bitrate must be provided, but not both

2) Super frame size is optional. Encoder operates on data frame size that is a product of the number of subbands and number of blocks. If input data comes in frames of a certain predefined size, then the super_frame_size quantity should be set to that size.

3) Psychoacoustic model may or may not be used.

6.3 UNIX Reference

To execute on Linux x/86 type:

test_sbc_enc_ x86_unix < input_file> [-h] [-l<blk_len>] [-m<mode>] [-o<output_file>] [-n<subbands> [-p] [–r <rate>] [-b<bitpool>] [-s<sample_freq>] [-f<super_frame_size>]

7 Pre compilation Options

7.1 Test application

The following C options need to be set

C Defines	Description	Remarks
TIME_PROFILE	To run the code for profiling	Both RVDS and ELINUX platform are available
LIB_TYPE	To build binary with static or dynamic libray	Only available in ELINUX

7.2 Library

C Defines	Description	Remarks
ENABLE_ASM	To build the library enabling assembly files	Elinux and RVDS build only