Release Notes for G.723.1 Decoder and Encoder

ABSTRACT:

Release Notes for G.723.1 Decoder and Encoder

KEYWORDS:

Multimedia codecs, speech, G.723.1

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
1.0	19-Jan-2005	Ashok Kumar	Final release 1.0
2.0	16-May-2005	Ashok Kumar	Release 2.0 tested on board
2.1	09-Sep--2005	Raja	Build Procedure changes for RVDS2.2
3.0	06-Feb-2006	Lauren Post	Using new format
3.1	19-Jan-2007	Sukruth	Updated for ARM9 Optimizations
3.2	10-july-2008	Jackiea pan	Update release notes

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 Speech Signal processing and G.723.1 codec.

1.4 References

1.4.1 Standards

- **ITU-T Recommendation G.723.1 (03/96)** –Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbits/s.
- **ITU-T Recommendation G.723.1 Annex A (11/96) –**Silence compression scheme
- **ITU-T Recommendation G.723.1 Implementers Guide (25 October 2002)** – Implementers' Guide for G.723.1
- **ITU-T G.723.1 Test vectors (1996)** –Description of the digital test sequences for the verification of the G.723.1 algorithm.

1.4.2 Freescale Multimedia References

- G.723.1 Codec Application Programming Interface – g723_codec_api.doc
- G.723.1 Codec Requirements Book – g723_codec_reqb.doc
- G.723.1 Codec Test Plan - g723_codec_test_plan.doc
- G.723.1 Codec Release notes - g723_codec_release_notes.doc
- G.723.1 Codec Test Results – g723_codec_test_results.doc
- G.723.1 Codec Performance Results – g723_codec_perf_results.doc
- G.723.1 Codec datasheet – g723_codec_datasheet.doc
- G.723.1 Interface Common Header – g723_com_api.h
- G.723.1 Interface Decoder Header – g723_dec_api.h
- G.723.1 Interface Encoder Header – g723_enc_api.h
- G.723.1 Decoder Application Code – g723_dectest.c

- G.723.1 Encoder Application Code – g723_enctest.c

1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
API	Application Programming Interface
ARM	Advanced RISC Machine
CNG	Comfort Noise Generation
DTX	Discontinuous Transmission
FSL	Freescale
ITU	International Telecommunication Union
MIPS	Million Instructions per Second
OS	Operating System
PCM	Pulse Code Modulation
SID	Silence Insertion Descriptor
RVDS	ARM RealView Development Suite
TBD	To Be Determined
UNIX	Linux PC x/86 C-reference binaries
VAD	Voice Activity Detection

1.6 Document Location

docs/g.723.1

.

2 Release History

RELEASE NUMBER	DELIVERABLES	FEATURES
1.0	• Documentation • Interface header file for encoder and decoder • ELINUX and RVDS libraries and test applications for decoder and encoder • 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. • 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 ITU-T standard test vectors. Sample application can be used to build executables
2.2	• Same	• Shared Library Support • Bus Alignment Fixes • Upgrade to RVDS 2.2
2.3	• Same	• ARM9 Optimized Library

Table 1. Details of the release

2.1 Assumptions and Known Problems

None

2.2 Contacts

Please report any problems to the following email address: mmsw@freescale.com

3 List of Deliverables

3.1 Documentation

Base directory: / fsl_mad_multimedia_codec /

Subdirectory	Files
docs/g.723.1	g723_codec_api.doc g723_codec_reqb.doc g723_codec_test_plan.doc g723_codec_test_results.doc g723_codec _release_notes.doc g723_codec_perf_results.doc g723_codec_datasheet.doc

3.2 Public Headers

Base directory: / fsl_mad_multimedia_codec /

Sibdirectory	Files	Description
ghdr	g723_common_api.h g723_enc_api.h g723_dec_api.h	G.723.1 common, encoder and decoder header file

3.3 Test Application Source

Base directory: / fsl_mad_multimedia_codec /

Subdirectory	Files
test/g.723.1	"Makefile" makefile for building RVDS, UNIX and ELINUX board executables.
test/g.723.1/hdr	*.h, application headers.
test/g.723.1/c_src	*.c, application code.
utils/g.723.1	Batch files to be run on the board and RVDS

3.4 Library Source

Base directory: / fsl_mad_multimedia_codec /

Subdirectory	Files
src/g.723.1	Makefile "Makefile" for building RVDS, UNIX, and ELINUX libraries. lib_g.723.1_arm9_elinux.a: static library for MX21 lib_g.723.1_arm11_bervds.a: ARM11 BE RVDS library lib_g.723.1_arm9_bervds.a: ARM9 BE RVDS library lib_g.723.1_arm11_elinux.a: static library for MX31 lib_g.723.1_arm11_elinux.so: dynamic library for MX31

	lib_g.723.1_arm11_lervds.a: ARM11 LE RVDS library lib_g.723.1_arm9_lervds.a: ARM9 LE RVDS library lib_g.723.1_x86_unix.a : library for Linux x/86 – c reference code
src/g.723.1/c_src	*.c, G.723.1 source code
src/g.723.1/hdr	*.h G.723.1 library header files

3.5 Common Makefiles

Base Directory: fsl_mad_multimedia_codec /

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.

4 Software Setup & Tools used

- ARM RVDS 2.2 (build 503) should be installed in the PC.
- Freescale Linux OS Release L26.1.15 must be running on the evaluation board.
- Intel based Red Hat Linux Machine must have the Montavista toolchain installed on it.
  - o 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/ ARM926EJ-S). The details for the build procedure are described below.

Note: The build procedure is explained with decoder as an example. To build library for the encoder applies the same procedure given below, with the makefile 'Makefile'.

5.1 Library

To build the library, run 'make' on 'Makefile' from src/g.723.1 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**:
  - o **BUILD= ARM11ELINUX** : It builds both static as well as dynamic libraries, 'lib_g.723.1_dec_arm11_elinux.a ' and shared library 'lib_ g.723.1_dec_arm11_elinux.so' , for testing on the board.

  - o **BUILD=ARM11LERVDS**: This option builds the static library 'lib_g.723.1_arm11_lervds.a', for testing on ARM11 LE RVDS (Armulator).

  - o **BUILD=ARM11BERVDS**: This option builds the static library 'lib_g.723.1_arm11_bervds.a', for testing on ARM11 BE RVDS (Armulator).

  - o **BUILD= ARM9ELINUX:** It builds static library, 'lib_g.723.1_arm9_elinux.a' for testing on the board.

  - o **BUILD=ARM9LERVDS**: This option builds the static library 'lib_g.723.1_arm9_lervds.a', for testing on ARM9 LE RVDS (Armulator).

  - o **BUILD=ARM9BERVDS**: This option builds the static library 'lib_g.723.1_arm9_bervds.a', for testing on ARM9 BE RVDS (Armulator).

  - o **BUILD=UNIX**: This option builds the static library 'lib_g.723.1_x86_unix.a', for testing on UNIX/Linux machine.

b) **clean options**:
  - o **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, src/g.723.1.

Target	Compilation Environment	Build Options	Library Name
Board (MX31)	PC(Using Cygwin)	BUILD=ARM11ELINUX	lib_g.723.1_dec_arm11_elinux.a lib_g.723.1_enc_arm11_elinux.a lib_g.723.1_dec_arm11_elinux so lib_g.723.1_enc_arm11_elinux.so
RVDS	PC(Using Cygwin)	BUILD=ARM11LERVDS BUILD=ARM11BERVDS BUILD=ARM9LERVDS BUILD=ARM9BERVDS	lib_g.723.1_dec_arm11_lervds.a lib_g.723.1_enc_arm11_lervds.a lib_g.723.1_dec_arm11_bervds.a lib_g.723.1_enc_arm11_bervds.a lib_g.723.1_dec_arm9_lervds.a lib_g.723.1_enc_arm9_lervds.a lib_g.723.1_dec_arm9_bervds.a lib_g.723.1_enc_arm9_bervds.a
Unix/ Linux	Unix/Linux machine	BUILD=UNIX	lib_g.723.1_x86_dec_unix.a lib_g.723.1_x86_enc_unix.a
Board (MX21)	Linux/Unix machine	BUILD= ARM9ELINUX	lib_g.723.1_dec_arm9_elinux.a, lib_g.723.1_enc_arm9_elinux.a

5.2 Test Application

To build the test application, run 'make' from the test/g.723.1 directory. This makefile can create executables for testing on Linux x86, the ARM11/ARM9 board and RVDS for ARM11.. The following commands should be invoked so as to build the executables.

Note: The build procedure is explained with decoder as an example.To build library for the encoder applies the same procedure given below, with the makefile 'Makefile'.

Options
1) **BUILD options**:
   o **BUILD=ARM11ELINUX**: This option builds the executable 'test_g.723.1_arm11_elinux', for MX31 board.

   o **BUILD=ARM11LERVDS**: This option builds the executable 'test_g.723.1_arm11_lervds ' for the ARM11 LE RVDS (Armulator).

   o **BUILD=ARM11BERVDS**: This option builds the executable 'test_g.723.1_arm11_bervds ' for the ARM11 BE RVDS (Armulator).

   o **BUILD=ARM9ELINUX**:   This option builds the executable 'test_g.723.1_arm9_elinux', for MX21 board.

   o **BUILD=ARM11LERVDS**: This option builds the executable 'test_g.723.1_arm9_lervds ' for the ARM11 LE RVDS (Armulator).

    o **BUILD=ARM11BERVDS**: This option builds the executable 'test_g.723.1_arm9_bervds' for the ARM11 BE RVDS (Armulator).

    o **BUILD=UNIX**: This option builds the executable 'test_g.723.1_x86_unix' for the Unix/Linux machine.

2) **LIBRARY options**:
    o **LIB_TYPE= STATIC**: This option builds the ELINUX test application linked with the ELINUX static library 'lib_g.723.1_arm11_elinux.a'.If nothing is specified ,the executable links with shared library 'lib_g.723.1_arm11_elinux.so'

      Eg: make BUILD=ARM11ELINUX LIB_TYPE=STATIC

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

<u>Note:</u>
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
Board (MX31)	Redhat Linux Machine	BUILD=ARM11ELINUX LIB_TYPE = STATIC	test_g.723.1_dec_arm11_elinux test_g.723.1_enc_arm11_ elinux
RVDS	PC (Using Cygwin)	BUILD=LERVDS BUILD=BERVDS	test_g.723.1_dec_arm11_lervds test_g723.1_enc_arm11_ lervds test_g.723.1_dec_arm11_bervds test_g723.1_enc_arm11_ bervds test_g.723.1_dec_arm9_lervds test_g723.1_enc_arm9_ lervds test_g.723.1_dec_arm9_bervds test_g723.1_enc_arm9_ bervds
UNIX/ Linux	Unix/Linux machine	BUILD=UNIX	test_g.723.1_dec__x86_unix test_g.723.1_enc__x86_unix
Board (MX21)	Redhat Linux Machine	BUILD=ARM9ELINUX	test_g.723.1_dec_arm9_elinux test_g.723.1_enc_arm9_elinux

6 Test Application Execution

6.1 Scripts

In the utils/g.723.1/ directory, a script file exists for doing

    a)   Regression, Performance on MX31 and MX21 (g.723.1_run_linux.sh)

    b)   Sanity on LE RVDS (g.723.1_run_rvds.sh)

6.2 ELINUX

The user is expected to be aware of the settings to be done for the hardware and to get Linux running on ARM11/ARM9

    a) Go to the directory utils/g.723.1" and edit scripts verify that paths are correct.
    b) Make sure the scripts are changed according to current test setup.
    c) create a working directory on the board and copy the executables from test/g.723.1 to the current directory
    d) copy the required script file (.sh) from utils/g.723.1into the working directory on the board
    e) Compare output of encoder and decoder using diff script provided in utils/g.723.1.

6.3 RVDS

The batch files to test encoder and decoder on RVDS are provided in utils/g.723.1. Run the script from PC (DOS) command prompt.
Note: Please verify the input, output and image path before running the script.

6.4 UNIX Reference

The script described in ELINUX execution can be used for C reference. Modify the script or pass in the parameter for ENCODER_EXE and DECODER_EXE which will be test_g.723.1_x86_enc_unix and test_g.723.1_dec_x86_UNIX respectively

7 Pre compilation Options

7.1 Test application

The following C options need to be set

C Defines	Description	Remarks
G723_BIG_ENDIAN	To run the code as big endian	
TIME_PROFILE	To run the code for profiling	ELINUX build only

7.2 Library

C Defines	Description	Remarks
G723_BIG_ENDIAN	To run the code as big endian	
GNU_INLINE_ASM	To use Inline assembly of Basic Opts in Gnu syntax	MX21 Board
ARM_INLINE_ASM	To use Inline assembly of Basic Opts in RVDS syntax	MX31 Board & RVDS
G723_C_VERSION	To compile C only code	
G723_ARM11_VERSION	Inline assembly instructions in ARMV6 architecture environment.	
G723_ARM9_VERSION	Inline assembly instructions in ARMV5E architecture environment.	
NO_INLINE_ASM	Use Assembly files instead of functions with inline assembly	This can be used with compilers not supporting inline assembly