08-7023-RN-ZCH66

OCT 17, 2008

1.0

Release Notes for G.729AB Decoder and Encoder

ABSTRACT:

Release Notes for G.729AB Decoder and Encoder

KEYWORDS:

Multimedia codecs, speech, G.729AB

APPROVED:

Shang Shidong

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
0.1	20-Aug-2008	Bing Song	Draft Version
1.0	17-Oct-08	Qiu Cunshou	Update for v1.4

Table of Contents

Introduction **4**
1.1 Purpose 4
1.2 Scope 4
1.3 Audience Description 4
1.4 References 4
1.4.1 Standards 4
1.4.2 Freescale Multimedia References 4
1.5 Definitions, Acronyms, and Abbreviations 5
1.6 Document Location 5

2 Release History **6**
2.1 Assumptions and Known Problems 6
2.2 Contacts 6

3 List of Deliverables **7**
3.1 Documentation 7
3.2 Public Headers 7
3.3 Test Application Source 7
3.4 Library Source 7
3.5 Common Makefiles 8

4 Software Setup & Tools used **9**

5 Build Procedure **10**
5.1 Library 10
5.2 Test Application 11

6 Test Application Execution **13**
6.1 Scripts 13
6.2 ELINUX 13
6.2.1 ARM9 Encoder 13
6.2.2 ARM9 Decoder 13
6.2.3 ARM11 Encoder 13
6.2.4 ARM11 Decoder 13
6.3 RVDS 14
6.3.1 ARM9 Encoder 14
6.3.2 ARM9 Decoder 14
6.3.3 ARM11 Encoder 14
6.3.4 ARM11 Decoder 14
6.4 UNIX Reference 14
6.4.1 Encoder 14
6.4.2 Decoder 15

7 Pre compilation Options **16**
7.1 Test application 16
7.2 Library 16

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.729AB codec.

1.4 References

1.4.1 Standards

- **ITU-T Recommendation G.729 (01/2007)** – Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction.

1.4.2Freescale Multimedia References

- G.729AB Codec Application Programming Interface – g729_codec_api.doc
- G.729AB Codec Requirements Book – g729_codec_reqb.doc
- G.729AB Codec Test Plan - g729_codec_test_plan.doc
- G.729AB Codec Release notes - g729_codec_release_notes.doc
- G.729AB Codec Test Results – g729_codec_test_results.doc
- G.729AB Codec Performance Results – g729_codec_perf_results.doc
- G.729AB Codec datasheet – g729_codec_datasheet.doc
- G.729AB Interface Common Header – g729_com_api.h
- G.729AB Interface Decoder Header – g729_dec_api.h
- G.729AB Interface Encoder Header – g729_enc_api.h
- G.729AB Decoder Application Code – g729_dectest.c
- G.729AB Encoder Application Code – g729_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.729ab

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 • 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 • Shared Library Support

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/g.729ab	g729_codec_api.doc g729_codec_reqb.doc g729_codec_test_plan.doc g729_codec_test_results.doc g729_codec _release_notes.doc g729_codec_perf_results.doc g729_codec_datasheet.doc

3.2 Public Headers

Base directory: / fsl_mad_multimedia_codec /

Sibdirectory	Files	Description
ghdr	g729_common_api.h g729_enc_api.h g729_dec_api.h	G.729AB common, encoder and decoder header file

3.3 Test Application Source

Base directory: / fsl_mad_multimedia_codec /

Subdirectory	Files
test/g.729ab	“Makefile” makefile for building RVDS, UNIX and ELINUX board executables.
test/g.729ab/hdr	*.h, application headers.
test/g.729ab/c_src	*.c, application code.
utils/g.729ab	Batch files to be run on the board and RVDS

3.4 Library Source

Base directory: / fsl_mad_multimedia_codec /

Subdirectory	Files
src/g.729ab	Makefile “Makefile” for building RVDS, UNIX, and ELINUX libraries. lib_g.729ab_arm9_elinux.a: static library for ARM9 lib_g.729ab_arm11_elinux.a: static library for ARM11 lib_g.729ab_arm9_elinux.so: dynamic library for ARM9 lib_g.729ab_arm11_elinux.so: dynamic library for ARM11 lib_g.729ab_arm11_lervds.a: ARM11 LE RVDS library

	lib_g.729ab_arm9_lervds.a: ARM9 LE RVDS library lib_g.729ab_x86_unix.a : library for Linux x/86 – c reference code
src/g.729ab/c_src	*.c, G.729AB source code
src/g.729ab/hdr	*.h G.729AB 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 3.0 (build 586) 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.
    - 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.729ab 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= ARM11ELINUX** : It builds both static as well as dynamic libraries, 'lib_g.729ab_dec_arm11_elinux.a ' and shared library 'lib_ g.729ab_dec_arm11_elinux.so' , for testing on the board.

- **BUILD=ARM11LERVDS**: This option builds the static library 'lib_g.729ab_arm11_lervds.a', for testing on ARM11 LE RVDS (Armulator).

- **BUILD= ARM9ELINUX:** It builds both static as well as dynamic libraries, 'lib_g.729ab_dec_arm9_elinux.a ' and shared library 'lib_ g.729ab_dec_arm9_elinux.so' , for testing on the board.

- **BUILD=ARM9LERVDS**: This option builds the static library 'lib_g.729ab_arm9_lervds.a', for testing on ARM9 LE RVDS (Armulator).

- **BUILD=UNIX**: This option builds the static library 'lib_g.729ab_x86_unix.a', for testing on UNIX/Linux machine.

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, src/g.729ab.

Target	Compilation Environment	Build Options	Library Name
Board (MX31)	PC(Using Cygwin)	BUILD=ARM11ELINUX	lib_g.729ab_dec_arm11_elinux.a lib_g.729ab_enc_arm11_elinux.a lib_g.729ab_dec_arm11_elinux so

			lib_g.729ab_enc_arm11_elinux.so
RVDS	PC(Using Cygwin)	BUILD=ARM9LERVDS	lib_g.729ab_dec_arm9_lervds.a lib_g.729ab_enc_arm9_lervds.a
RVDS	PC(Using Cygwin)	BUILD=ARM11LERVDS	lib_g.729ab_dec_arm11_lervds.a lib_g.729ab_enc_arm11_lervds.a
Unix/ Linux	Unix/Linux machine	BUILD=UNIX	lib_g.729ab_x86_dec_unix.a lib_g.729ab_x86_enc_unix.a
Board (MX27)	Linux/Unix machine	BUILD= ARM9ELINUX	lib_g.729ab_dec_arm9_elinux.a lib_g.729ab_enc_arm9_elinux.a lib_g.729ab_dec_arm9_elinux.so lib_g.729ab_enc_arm9_elinux.so

5.2 Test Application

To build the test application, run 'make' from the test/g.729ab 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**:
   - **BUILD=ARM11ELINUX**: This option builds the executable 'test_g.729ab_arm11_elinux', for MX31 board.
   - **BUILD=ARM11LERVDS**: This option builds the executable 'test_g.729ab_arm11_lervds ' for the ARM11 LE RVDS (Armulator).
   - **BUILD=ARM9ELINUX**: This option builds the executable 'test_g.729ab_arm9_elinux', for MX27 board.
   - **BUILD=ARM11LERVDS**: This option builds the executable 'test_g.729ab_arm9_lervds ' for the ARM11 LE RVDS (Armulator).
   - **BUILD=UNIX**: This option builds the executable 'test_g.729ab_x86_unix' for the Unix/Linux machine.

2) **LIBRARY options**:
   - **LIB_TYPE= STATIC**: This option builds the ELINUX test application linked with the ELINUX static library 'lib_g.729ab_arm11_elinux.a'.If nothing is specified ,the executable links with shared library 'lib_g.729ab_arm11_elinux.so'

Eg: make BUILD=ARM11ELINUX LIB_TYPE=STATIC

3) **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
Board (MX31)	Redhat Linux Machine	BUILD=ARM11ELINUX LIB_TYPE = STATIC	test_g.729ab_dec_arm11_elinux test_g.729ab_enc_arm11_ elinux
RVDS	PC (Using Cygwin)	BUILD=ARM9LERVDS	test_g.729ab_dec_arm9_lervds test_g.729ab_enc_arm9_ lervds
RVDS	PC (Using Cygwin)	BUILD=ARM11LERVDS	test_g.729ab_dec_arm11_lervds test_g.729ab_enc_arm11_ lervds
UNIX/ Linux	Unix/Linux machine	BUILD=UNIX	test_g.729ab_dec_x86_unix test_g.729ab_enc_x86_unix
Board (MX27)	Redhat Linux Machine	BUILD=ARM9ELINUX	test_g.729ab_dec_arm9_elinux test_g.729ab_enc_arm9_elinux

6 Test Application Execution

6.1 Scripts

In the utils/g.729ab/ directory, a script file exists for doing

a) Regression, Performance on MX31 and MX27 (g.729ab_run_linux.sh)

b) Sanity on LE RVDS (g.729ab_run_rvds.sh)

6.2 ELINUX

6.2.1 ARM9 Encoder

test_g729ab_enc_arm9_elinux <InpFile> <OutFile> <VAD_flag>

Where:

InpFile	is the name of the file to be processed.
OutFile	is the name with the processed data.
VAD_flag	0: disable VAD, 1: enable VAD.

6.2.2 ARM9 Decoder

test_g729ab_dec_arm9_elinux <InpFile> <OutFile>

Where:

InpFile	is the name of the file to be processed.
OutFile	is the name with the processed data.

6.2.3 ARM11 Encoder

test_g729ab_enc_arm11_elinux <InpFile> <OutFile> <VAD_flag>

Where:

InpFile	is the name of the file to be processed.
OutFile	is the name with the processed data.
VAD_flag	0: disable VAD, 1: enable VAD.

6.2.4 ARM11 Decoder

test_g729ab_dec_arm11_elinux <InpFile> <OutFile>

Where:

InpFile	is the name of the file to be processed.

OutFile is the name with the processed data.

6.3 RVDS

6.3.1 ARM9 Encoder

test_g729ab_enc_arm9_lervds <InpFile> <OutFile> <VAD_flag>

Where:
InpFile is the name of the file to be processed.
OutFile is the name with the processed data.
VAD_flag 0: disable VAD, 1: enable VAD.

6.3.2 ARM9 Decoder

test_g729ab_dec_arm9_lervds <InpFile> <OutFile>

Where:
InpFile is the name of the file to be processed.
OutFile is the name with the processed data.

6.3.3 ARM11 Encoder

test_g729ab_enc_arm11_lervds <InpFile> <OutFile> <VAD_flag>

Where:
InpFile is the name of the file to be processed.
OutFile is the name with the processed data.
VAD_flag 0: disable VAD, 1: enable VAD.

6.3.4 ARM11 Decoder

test_g729ab_dec_arm11_lervds <InpFile> <OutFile>

Where:
InpFile is the name of the file to be processed.
OutFile is the name with the processed data.

6.4 UNIX Reference

6.4.1 Encoder

test_g.729ab_enc_x86_unix <InpFile> <OutFile> <VAD_flag>

Where:
InpFile is the name of the file to be processed.
OutFile is the name with the processed data.

VAD_flag 0: disable VAD, 1: enable VAD.

6.4.2 Decoder

test_g.729ab_dec_x86_unix <InpFile> <OutFile>

Where:
InpFile is the name of the file to be processed.
OutFile is the name with the processed data.

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	ELINUX build only

7.2 Library

C Defines	Description	Remarks
GNU_INLINE_ASM	To use Inline assembly of Basic Opts in Gnu syntax	MX27 Board
ARM_INLINE_ASM	To use Inline assembly of Basic Opts in RVDS syntax	MX31 Board & RVDS
G729_C_VERSION	To compile C only code	
G729_ARM11_VERSION	Inline assembly instructions in ARMV6 architecture environment.	
G729_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