08-6796-RN-ZCH66
JUNE 17, 2008
3.1

Release Notes for G.726 Decoder and Encoder on ARM9/ARM11 ELINUX

ABSTRACT:

Release Notes for G.726 Decoder and Encoder on ARM9/ARM11 ELINUX

KEYWORDS:

Multimedia codecs, speech, G.726

APPROVED:

Shang Shidong

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
1.0	30-Sep-2004	Ashok Kumar	Final release 1.0
2.0	28-Mar-2005	Ashok Kumar	Release 2.0 tested on board
2.1	06-Sep-2005	Anand	Build procedure changes for RVDS2.2
3.0	06-Feb-2006	Lauren Post	Using new format
3.1	17-Jun-2008	Qiu Cunshou	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 ARM9/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 Speech Signal processing and G.726 codec.

1.4 References

1.4.1 Standards

- **ITU-T Recommendation G.726 (1990)** – 40, 32, 24, 16 kbit/s Adaptive differential pulse code modulation (ADPCM).
- **ITU-T Recommendation G.711 (1988)** –Pulse code modulation (PCM) of voice frequencies.
- **G.726 Appendix II Test vectors (1991)** –Description of the digital test sequences for the verification of the G.726 40,32,24 and 16 kbps ADPCM algorithm.

1.4.2 Freescale Multimedia References

- G.726 Codec Application Programming Interface – g726_codec_api.doc
- G.726 Codec Requirements Book – g726_codec_reqb.doc
- G.726 Codec Test Plan - g726_codec_test_plan.doc
- G.726 Codec Release notes - g726_codec_release_notes.doc
- G.726 Codec Test Results – g726_codec_test_results.doc
- G.726 Codec Test Results – g726_codec_perf_results.doc
- G.726 Codec Data sheet – g726_codec_datasheet.doc
- G.726 Interface Common Header – g726_com_api.h
- G.726 Interface Decoder Header – g726_dec_api.h
- G.7261 Interface Encoder Header – g726_enc_api.h
- G.726 Decoder Application Code – g726_dectest.c
- G.726 Encoder Application Code – g726_enctest.c

1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
ADPCM	Adaptive Differential Pulse Code Modulation
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.726

2 Release History

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

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/

Directory	Files	Description
docs/g.726	g726_codec_api.doc g726_codec_reqb.doc g726_codec_test_plan.doc g726_codec_test_results.doc g726_codec_perf_results.doc g726_codec_release_notes.doc g726_codec_datasheet.doc	Application Programming Requirements Book Test Plan Test Results Performance Results Release Notes Data sheet

3.2 Public Headers

Base directory: /fsl_mad_multimedia_codec/

Directory	Files	Description
ghdr	g726_common_api.h g726_enc_api.h g726_dec_api.h	G.726 common, encoder and decoder header file

3.3 Test Application Source

Base directory: /fsl_mad_multimedia_codec/

Directory	Files	Description
test/g.726/	Makefile	makefile to build executables for RVDS, board, Linux and Unix for decoder and encoder
test/g.726/c_src	common encoder decoder	Folders containing the c source file for the sample test application
test/g.726/hdr	*.h	Header files for the test application
utils/g.726	*.bat *.sh	Batch for running the test cases and output comparison on Wince Scripts for running the test cases and file comparison on the board and rvds

3.4 Library Source

Base directory: /fsl_mad_multimedia_codec/src/g.726/

Directory	Files	Description
.	Makefile	Library makefiles for encoder and decoder
c_src asm_arm	C and ASM files are provided in these folders. Both encoder and decoder files are given separately in /encoder and /decoder folders. Common files are provided in /common folder.	C and ASM files are provided in this folder
hdr	*.h	Header files for g.726 codec

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 for building the test application. This file includes the common options used by the all the codecs.

3.6 Test Vectors

Base Directory: multimedia_vectors/test_vectors

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

Subdirectory	Description
g.726	All ITU-T 120 test vectors and reference vectors, including a-law, mu-law and linear vectors.

4 Software Setup & Tools used

- ARM RVDS 3.0 (build 441) should be installed in the PC.
- Freescale Linux OS Release L26.1.17 must be running on the evaluation board.
- Intel based Red Hat Linux Machine must have the devtek toolchain installed on it.
    - devtek Toolchain gcc 4.1.1 glibc 2.4 nptl 6
- '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 (ARM926EJ-S and ARM1136J-S). The details for the build procedure are described below.

Note: The build procedure is explained with encoder as an example. The library for the decoder will be created by the same procedure.

5.1 Library

To build the library, run 'make' on 'Makefile' (for both encoder and decoder) from src 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** : This option builds both static library 'lib_g.726_enc_arm9_elinux.a' and shared library 'lib_g.726_enc_arm9_elinux.so', for testing on the board.

- **BUILD=ARM11ELINUX** : This option builds both static library 'lib_g.726_enc_arm11_elinux.a' and shared library 'lib_g.726_enc_arm11_elinux.so', for testing on the board.

- **BUILD=ARM11LERVDS**: This option builds the static library 'lib_g.726_enc_arm11_lervds.a' for testing on RVDS (Armulator).

- **BUILD=ARM9LERVDS**: This option builds the static library 'lib_g.726_enc_arm9_lervds.a' for testing on RVDS (Armulator).

- **BUILD=UNIX**: This option builds the static library 'lib_g.726_enc_UNIX.a' for testing on UNIX/Linux machine.

  Eg: make BUILD=ARM9ELINUX (for board)
  make BUILD=ARM9LERVDS (for Armulator)
  make BUILD=UNIX (for Unix/Linux machine)

b) **clean options**:

- **clean**: Deletes all the object files and RVDS,UNIX and ELINUX libraries.

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

The library that is built is saved as lib_g.726_enc_arm11_rvds.a for RVDS build, and lib_g.726_enc_arm11_elinux.a and lib_g.726_enc_arm11_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= ARM9ELINUX	lib_g.726_dec_arm9_elinux.a lib_g.726_enc_arm9_elinux.a lib_g.726_dec_arm9_elinux.so lib_g.726_enc_arm9_elinux.so
Board	PC (Using Cygwin)	BUILD= ARM11ELINUX	lib_g.726_dec_arm11_elinux.a lib_g.726_enc_arm11_elinux.a lib_g.726_dec_arm11_elinux.so lib_g.726_enc_arm11_elinux.so
RVDS	PC (Using Cygwin)	BUILD=ARM9LERV DS	lib_g.726_dec_arm9_lervds.a lib_g.726_enc_arm9_lervds.a
RVDS	PC (Using Cygwin)	BUILD=ARM11LER VDS	lib_g.726_dec_arm11_lervds.a lib_g.726_enc_arm11_lervds.a
Unix/ Linux	Unix/Linux machine	BUILD=UNIX	lib_g.726_dec_unix.a lib_g.726_enc_unix.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 board, RVDS, Linux and Unix. The executables test_g.726_enc_arm11_lervds (test_g.726_dec_arm11_lervds for decoder) for RVDS, test_g.726_enc_arm11_elinux (test_g.726_dec_arm11_elinux for decoder) for board and test_g.726_enc_unix (test_g.726_dec_unix for decoder) for Linux and Unix platforms are stored under test/exe 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=ARM9ELINUX**: This option builds the executable 'test_g.726_enc_arm9_elinux', for the board.
   - **BUILD=ARM11ELINUX**: This option builds the executable 'test_g.726_enc_arm11_elinux', for the board.
   - **BUILD=ARM9LERVDS**: This option builds the executable 'test_g.726_enc_arm9_lervds' for the RVDS (Armulator).
   - **BUILD=ARM11LERVDS**: This option builds the executable 'test_g.726_enc_arm11_lervds' for the RVDS (Armulator).
   - **BUILD=UNIX**: This option builds the executable 'test_g.726_enc_unix' for the Unix/Linux machine.

Eg: make BUILD=ARM9ELINUX (for board)
make BUILD=ARM9LERVDS (for Armulator)
make BUILD=UNIX (for Unix/Linux machine)

2) **clean options**:
   - **clean**: Deletes all the object files and RVDS,UNIX ELINUX executables.

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	Redhat Linux Machine	BUILD=ARM9ELINUX	test_g.726_dec_arm9_elinux test_g.726_enc_arm9_elinux
Board	Redhat Linux Machine	BUILD=ARM11ELINUX	test_g.726_dec_arm11_elinux test_g.726_enc_arm11_elinux
RVDS	PC (Using Cygwin)	BUILD=ARM9LERVDS	test_g.726_dec_arm9_lervds test_g.726_enc_arm9_lervds
RVDS	PC (Using Cygwin)	BUILD=ARM11LERVDS	test_g.726_dec_arm11_lervds test_g.726_enc_arm11_lervds
UNIX/ Linux	Unix/Linux machine	BUILD=UNIX	test_g.726_dec_unix test_g.726_enc_unix

6 Test Application Execution

6.1 Scripts

In the utils/g.726 directory, a script file exists for doing batch processing on several vectors. The script can be modified or parameters set to specify the binaries to use.

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 ARM9 or ARM11

a) Go to the directory utils/g.726" 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/exe to the current directory
d) copy the required script file (.sh) from test_util/scripts into the working directory on the board
e) Compare output of encoder and decoder using diff script provided in utils/g.726.

6.3 RVDS

The batch files to test encoder and decoder on RVDS are provided in utils/g.726. 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.726_enc_unix and test_g.726_dec_unix respectively

7 Pre compilation Options

The following C options need to be set

C Defines	Description	Remarks
TIME_PROFILE	Uses SYSTEM_TIME for RVDS and UNIX builds and TIME_PROFILE for ELINUX builds	For Elinux and RVDS
DBG_BUILD=1	Enalbe debug mode for mips test	For Elinux and RVDS
OPTIM_LEVEL=-O0	Set optimal level to O0 for mips test	For Elinux and RVDS