ETSI TS 126 443 V12.5.0 (2016-04)

TECHNICAL SPECIFICATION

Universal Mobile Telecommunications System (UMTS);
LTE;
Codec for Enhanced Voice Services (EVS);
ANSI C code (floating-point)
(3GPP TS 26.443 version 12.5.0 Release 12)

Reference
RTS/TSGS-0426443vc50

Keywords
LTE,UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document contains an electronic copy of the ANSI C floating-point code for the Enhanced Voice Services (EVS) Codec. This ANSI C code is the unique alternative reference specification besides the ANSI-C fixed-point code for the EVS Codec (3GPP TS 26.442) for a standard compliant implementation of the EVS Codec (3GPP TS 26.445), Voice Activity Detection (VAD) (3GPP TS 26.451), Comfort Noise Generation (CNG) (3GPP TS 26.449), Discontinuous Transmission (DTX) (3GPP TS 26.450), Packet Loss Concealment (PLC) of Lost Packets (3GPP TS 26.447), Jitter Buffer Management (JBM) (3GPP TS 26.448), and AMR-WB Interoperable Function (3GPP TS 26.446).

Requirements for any implementation of the EVS codec to be standard compliant are specified in 3GPP TS 26.444 (Test sequences).

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 26.445: "Codec for Enhanced Voice Services (EVS); Detailed Algorithmic Description".

[3] 3GPP TS 26.451: "Codec for Enhanced Voice Services (EVS); Voice Activity Detection (VAD)".

[4] 3GPP TS 26.449: "Codec for Enhanced Voice Services (EVS); Comfort Noise Generation (CNG) Aspects".

[5] 3GPP TS 26.450: "Codec for Enhanced Voice Services (EVS); Discontinuous Transmission (DTX)".

[6] 3GPP TS 26.447: "Codec for Enhanced Voice Services (EVS); Error Concealment of Lost Packets".

[7] 3GPP TS 26.448: "Codec for Enhanced Voice Services (EVS); Jitter Buffer Management".

[8] 3GPP TS 26.446: "Codec for Enhanced Voice Services (EVS); AMR-WB Backward Compatible Functions".

[9] IETF RFC 3550: "RTP: A Transport Protocol for Real-Time Applications".

[10] Recommendation ITU-T G.191 (03/10): "Software tools for speech and audio coding standardization".

[11] Recommendation ITU-T G.192: "A common digital parallel interface for speech standardization activities".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the definitions given in TR 21.905 [1] apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

ACELP	Algebraic Code-Excited Linear Prediction
AMR-WB	Adaptive Multi Rate Wideband (codec)
CNG	Comfort Noise Generator
DTX	Discontinuous Transmission
EVS	Enhanced Voice Services
FB	Fullband
FEC	Frame Erasure Concealment
IP	Internet Protocol
JBM	Jitter Buffer Management
MSB	Most Significant Bit
MTSI	Multimedia Telephony Service for IMS
NB	Narrowband
PS	Packet Switched
PSTN	Public Switched Telephone Network
SAD	Sound Activity Detection
SC-VBR	Source Controlled - Variable Bit Rate
SID	Silence Insertion Descriptor
SWB	Super Wideband
VAD	Voice Activity Detection
WB	Wideband
WMOPS	Weighted Millions of Operations Per Second

4 C code structure

This clause gives an overview of the structure of the floating-point C code and provides an overview of the contents and organization of the C code attached to the present document.

The ANSI-C code has been verified on the following platforms:

- IBM PC compatible computers with Windows 7 operating systems and Microsoft Visual C++ 2010 compiler, 32-bit.

ANSI-C was selected as the programming language because portability was desirable.

4.1 Contents of the C source code

The C code is organized as follows:

Table 1: Source code directory structure

Directory	Description
README.txt	information on how to compile
Makefile	UNIX style encoder Makefile
Workspace_msvc/	Directory for the MSVC 2008 (or newer) project files
lib_com/	Source code files used both in encoder and decoder
lib_dec/	Source code files used solely in the decoder
lib_enc/	Source code files used solely in the encoder

The distributed files with suffix "c" contain the source code and the files with suffix "h" are the header files. The ROM data is contained in files named "rom_xxx" with suffix "c".

Makefiles are provided for the platforms in which the C code has been verified (listed above). Once the software is installed, this directory will have a compiled version of the encoder (named EVS_cod) and the decoder (named EVS_dec).

4.2 Program execution

The codec for Enhanced Voice Services is implemented in two programs:

- EVS_cod: encoder;
- EVS_dec: decoder.

The programs should be called like:

- EVS_cod [encoder options] <input file> <bitstream file>;
- EVS_dec [decoder options]<bitstream file> < output file>.

The input and output files contain 16-bit linear encoded PCM samples and the bitstream file contains encoded data.

The encoder and decoder options will be explained by running the programs without any input arguments. See the file readme.txt for more information on how to run the *encoder* and *decoder* programs.

5 File formats

This clause describes the file formats used by the encoder and decoder programs. The test sequences defined in [1] also use the file formats described here.

5.1 Input/output file format

Input files read by the encoder and output files written by the decoder consist of 16-bit integer words per each data sample. The byte order in each word depends on the host architecture (e.g. LSB first on PCs, etc.). Both the encoder and the decoder program process complete frames corresponding to multiples of 20 ms. The remaining samples are discarded.

The encoder will pad the last frame to integer multiples of 20ms frames, i.e. n speech frames will be produced from an input file with a length between [(n-1)*20ms+1 sample; n*20ms]. The files produced by the decoder will always have a length of n*20ms.

5.2 Rate switching profile (encoder input)

The encoder program can optionally read in a rate switching profile file which specifies the encoding bitrate for each frame of the input data. The rate switching profile is a binary file, generated by 'gen-rate-profile' tool, which is part of STL 2009, as contained in ITU-T G.191 [10]. The rate switching profile contains 32-bit integer words where each word represents the encoding bitrate for each particular frame. The rate switching profile is recycled if it contains less entries than the total number of frames in the input file. The rate switching profile can contain EVS primary mode bitrates and AMR-WB IO mode bitrates arbitrarily. I.e. switching between the two modes can be specified by the rate switching profile.

5.3 Parameter bitstream file (encoder output / decoder input)

The files produced by the speech/audio encoder/expected by the speech decoder contain an arbitrary number of frames in the following available formats.

5.3.1 ITU-T G.192 compliant format

SYNC_WORD	DATA_LENGTH	B1	B2	…	Bnn

Each box corresponds to one `Word16` value in the bitstream file, for a total of 2+nn words or 4+2nn bytes per frame, where nn is the number of encoded bits in the frame. Each encoded bit is represented as follows: Bit 0 = 0x007f, Bit 1 = 0x0081. The fields have the following meaning:

- `SYNC_WORD:` Word to ensure correct frame synchronization between the encoder and the decoder. It is also used to indicate the occurrences of bad frames.

  In the encoder output: (0x6b21)

  In the decoder input: Good frames (0x6b21)

  Bad frames (0x6b20)

- `DATA_LENGTH:` Length of the speech data. Codec mode and frame type is extracted in the decoder using this parameter

5.3.2 Compact storage format file

The encoder and decoder programs can optionally write and read a file in the octet-based compact storage format. The compact storage format is specified in Annex A.2.6 of [2].

5.4 VoIP parameter bitstream file (decoder input)

Packet size	Arrival time	RTP header	G.192 format (see 6.3.1)

The fields have the following size and meaning:

- Packet size: 32 bit unsigned integer (= 12 + 2 + DATA_LENGTH).

- Arrival time: 32 bit unsigned integer in ms.

- RTP header: 96 bits (see RFC 3550 [9]), including RTP timestamp and SSRC.

5.5 Bandwidth switching profile (encoder input)

The encoder program can optionally read in a bandwidth switching profile, which specifies the encoding bandwidth for each frame of speech processed. The file is a text file where each line contains 'nb_frames B'. B specifies the signal bandwidth that is one of the supported four bandwidths, i.e. NB, WB, SWB or FB. And 'nb_frames' is an integer number of frames and specifies the duration of activation of the accompanied signal bandwidth B.

5.6 Channel-aware configuration file (encoder input and decoder output)

The encoder program can optionally read in a configuration file which specifies the values of FEC indicator p and FEC offset o, where FEC indicator, p: LO or HI, and FEC offset, o: 2, 3, 5, or 7 in number of frames. Each line of the configuration file contains the values of p and o separated by a space.

The channel-aware configuration file is meant to simulate channel feedback from a receiver to a sender, i.e. the decoder would generate FEC indication and FEC offset values for receiver feedback that correspond to the current transmission channel characteristics, thereby allowing optimization of the transmission by the encoder which applies the FEC offset and FEC indication when in the channel-aware mode.

5.7 JBM trace file (decoder output)

The decoder can generate a JBM trace file with the –Tracefile switch as a by-product of the decoder operation in case of JBM operation (which is triggered with the –VOIP switch on the decoder side).

The trace file is a CSV file with semi-colon as separator. The trace file starts with one header line that contains the column names in the following order:

```
rtpSeqNo;rtpTs;rcvTime;playtime;active
```

For each played out speech frame one entry is written to the trace file. The interval of the playtime values is usually 20ms, but may differ, depending on the JBM operation. Each entry is a line in the trace file that contains values as specified in Table 1.

Table 1: JBM trace file entry format

Name	Unit	Description
rtpSeqNo	1	RTP sequence number of played out speech frame. -1 if no corresponding RTP packet for the speech frame exists.
rtpTs	ms	RTP time stamp of played out speech frame. -1 if no corresponding RTP packet for the speech frame exists
rcvTime	ms	Absolute reception time of the RTP packet that corresponds to the speech frame. -1 if no corresponding RTP packet for the speech frame exists.
playtime	ms	Absolute play time (i.e. the time at which the PCM data is made available by the decoder). Can be floating-point value.
active	0 or 1	Binary entry, which is set to 1 for active speech frames (i.e. frames that are neither SID nor NO_DATA)

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2014-12	66	SP-140729			Presented to TSG SA#66 for approval		1.0.0
2014-12	66					1.0.0	12.0.0
2015-03	67	SP-150084	0001	1	Corrections to the text of the specification	12.0.0	12.1.0
2015-03	67	SP-150084	0002	2	Bugfixes to EVS Floating Point Source Code	12.0.0	12.1.0
2015-03	67	SP-150084	0003	3	Implementation of the compact storage format in the EVS Floating Point Source Code	12.0.0	12.1.0
2015-06	68	SP-150201	0005		Bugfixes to EVS Floating Point Source Code	12.1.0	12.2.0
2015-09	69	SP-150434	0006	1	Corrections to EVS Floating-Point Source Code	12.2.0	12.3.0
2015-12	70	SP-150639	0007		Corrections to EVS Floating-Point Source Code	12.3.0	12.4.0
2016-03	71	SP-160064	0008		Corrections to EVS Floating-Point Source Code	12.4.0	12.5.0

History

Document history		
V12.0.0	January 2015	Publication
V12.1.0	April 2015	Publication
V12.2.0	July 2015	Publication
V12.3.0	October 2015	Publication
V12.4.0	January 2016	Publication
V12.5.0	April 2016	Publication