ETSI TS 126 243 V10.0.0 (2011-04)

Technical Specification

**Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
ANSI C code for the fixed-point distributed speech recognition extended advanced front-end
(3GPP TS 26.243 version 10.0.0 Release 10)**

Reference

RTS/TSGS-0426243va00

Keywords

GSM, 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

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 (http://webapp.etsi.org/IPR/home.asp).

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.

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 code for DSR Extended Advanced Front-end. The ANSI-C code is necessary for a bit exact implementation of DSR Extended Advanced Front-end.

2 References

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

[1] ETSI ES 202 050: "Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithm", Oct 2002.

[2] ETSI ES 202 212 "Distributed Speech Recognition; Extended Advanced Front-end Feature Extraction Algorithm; Compression Algorithm, Back-end Speech Reconstruction Algorithm", Nov 2003.

[3] 3GPP TS 26.177: "Speech Enabled Services (SES); Distributed Speech Recognition (DSR) extended advanced front-end test sequences".

3 Definitions and abbreviations

3.1 Definitions

Definition of terms used in the present document, can be found in [1], [2]

3.2 Abbreviations

For the purpose of the present document, the following abbreviations apply:

ANSI	American National Standards Institute
I/O	Input/Output
RAM	Random Access Memory
ROM	Read Only Memory
AFE	Advanced Front-end
X-AFE	eXtended Advanced Front-end
DSR	Distributed Speech Recognition

4 C code structure

This clause gives an overview of the structure of the bit-exact C code and provides an overview of the contents and organization of the C code attached to this document.

The C code has been verified on the following systems:

- Sun Microsystems workstations and GNU gcc compiler
- IBM PC compatible computers with Linux operating system and GNU gcc compiler.

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

4.1 Contents of the C source code

The distributed files with suffix "c" contain the source code and the files with suffix "h" are the header files.

Makefiles are provided for the platforms in which the C code has been verified (listed above).

4.2 Program execution

There are separate executables for the FrontEnd and Vector Quantization, with and without Extensions. The command line options are described below.

<> - indicates parameters for the given option for running the executable
() – indicates default parameter.

FrontEnd w/ Extension:
USAGE: bin/ExtAdvFrontEnd infile HTK_outfile pitch_outfile class_outfile [options]
OPTIONS:

-q	Quiet Mode (FALSE)
-F format	Input file format <*NIST,HTK,RAW*> (NIST)
-fs freq	Sampling frequency in kHz <*8,16*> (8)
-swap	Change input byte ordering (Native)
-noh	No HTK header to output file (FALSE)
-noc0	No c0 coefficient to output feature vector (FALSE)
-nologE	No logE component to output feature vector (FALSE)

-skip_header_bytes n - Skip header, first n bytes (Only for -F RAW)

-noh, -noc0, -nologE and –skip_header_bytes are not used and should not be changed.

FrontEnd w/o Extension:
USAGE: bin/AdvFrontEnd infile HTK_outfile [options]
OPTIONS: - Same as FrontEnd w/ Extension

Vector Quantization w/ Extension:
Usage: extcoder htk_file_in pitch_file_in class_file_in bitstream_file_out pitch_file_out txt_file_out -freq x -VAD/No_VAD

htk_file_in	Input mel-frequency cepstral coefficient file in HTK MFCC format.
pitch_file_in	Input pitch period file.
class_file_in	Input classification file.
bit_file_out	Output binary bitstream.
pitch_file_out	Output quantised pitch period file.
txt_file_out	Vector quantiser output in text format.
-freq x	Sampling frequency in kHz (8 or 16).
-VAD	Use voice activity detector data. Voice activity input file must have same name as htk_file, but extension .vad
-No_VAD	Do not incorporate voice activity detector information in output bitstream.

Vector Quantization w/o Extension:
Usage: coder htk_file_in bitstream_file_out txt_file_out -freq x -VAD/No_VAD

htk_file_in	Input mel-frequency cepstral coefficient file in HTK MFCC format.
bit_file_out	Binary output bitstream.
txt_file_out	Vector quantiser output in text format.
-freq x	Sampling frequency in kHz (8 or 16).
-VAD	Use voice activity detector data. Voice activity input file must have same name as htk_file, but extension .vad
-No_VAD	Do not incorporate voice activity detector information in output bitstream.

File extension descriptions as generated by the sample script:
.cep – Binary file containing cepstral features in HTK format. Output from the FrontEnd, input to the vector quantizer.
.pitch – Binary file containing pitch information. Output from the FrontEnd, input to the vector quantizer. Only used for Extension.
.class – Ascii file containing class information. Output from the FrontEnd, input to the vector quantizer. Only used for Extension.
.bs – Binary file containing the bitstream. Output from the vector quantizer.
.log – Log files from the different executables.

4.3 Code hierarchy

Tables 1 to 3 are call graphs that show the functions used for AFE (table 1), VQ (table 2), and Extension (table 3).

Each column represents a call level and each cell a function. The functions contain calls to the functions in rightwards neighboring cells. The time order in the call graphs is from the top downwards as the processing of a frame advances. All standard C functions: printf(), fwrite(), etc. have been omitted. Also, no basic operations (add(), L_add(), mac(), etc.) or double precision extended operations (e.g. L_Extract()) appear in the graphs.

The basic operations are not counted as extending the depth, therefore the deepest level in this software is level 7.

Table 1: AFE call structure

main()				
	AdvProcessInit_B()			
		DoNoiseSupInit_B()		
		DoWaveProcInit_B()		
		DoCompCepsInit_B()		
		DoPostProcInit_B()		
		DoVADInit_F()		
		Do16kProcInit_B()		
			QMF_FIR_Init_B()	
				fir_initialization_B()
				DP_HP_filters_B()
		BufIn32Alloc()		
	AdvProcessAlloc_B()			
		DoNoiseSupAlloc_B()		
		DoWaveProcAlloc_B()		
		DoCompCepsAlloc_B()		
		DoPostProcAlloc_B()		
		DoVADAlloc_F()		
		Do16kProcAlloc_B()		
	FlushAdvProcess_B()			
		DoVADFlush_F()		
		CvFeatInt2Float()		
	AdvProcessDelete_B()			
		DoNoiseSupDelete_B()		
		DoWaveProcDelete_B()		
		DoCompCepsDelete_B()		
		DoPostProcDelete_B()		
		DoVADDelete_B()		
		BufIn32Free()		
	DoAdvProcess_B()			
		Do16kProcessing_B()		
		DoNoiseSup_B()		
			Get16k_p_bufferData16k_B()	
			Get16k_bufData16kSize_B()	
			Get16k_p_BandsForCoding16k_B()	
			Get16k_p_CodeForBands16k_B()	
			Get16k_dataHP_B()	
			VAD_F()	
				Log_2()
			DoSigWindowing16_F1()	
			DoSigWindowing16_F2()	
			ff4NRFix32_B()	
				GetL15()
				GetH15()
				Mult16x32()
				Add_Mult16x16_16()
				Sub_Mult16x16_16()
				Permut()
			FFTtoPSD_F()	
				Square24d2_B()
				Square24_B()
			Get16k_BFC_dec_B()	
			GetBandsForCoding16k_B()	
			PSDMean_F()	
			NoiseEstimation_F1()	
				Sqrt_2()
				Sqrt16_2()
			NoiseEstimation_F2()	
				Sqrt_2()
				Sqrt16_2()
			FilterCalc_F()	
			SpeechQVar()	
			FilterBank16()	
			SpeechQSpec()	
			SpeechQMel()	
			DoGainFact_F1()	
				Log_2()
			DoGainFact_F2()	
				Log_2()
			DoMelIDCT_F16()	
			ApplyWF()	
			Get16k_dec1()	
			Get16k_dec2()	
			Get16k_dec3()	
			DoSigWindowing16_F3()	
			ff4NRFix32_B()	
				GetL15()
				GetH15()
				Mult16x32()
				Add_Mult16x16_16()
				Sub_Mult16x16_16()
				Permut()
			FFTtoPSD_F()	
				Square24d2_B()
				Square24_B()
			DoMelFB_B()	
			CodeBands16k_B()	
			DoSpecSub16k_B()	
				Log_2()
			UpDateDecal()	
			ApplyDecal()	
			DCOffsetFil_F()	
			Get16k_hpBandsSize_B()	

	Get16k_p_hpBands_B()	
	Get16k_p_bufferCodeForBands16k_B()	
	Get16k_p_CodeForBands16k_B()	
	Get16k_p_bufferCodeWeights_B()	
	Get16k_p_codeWeights_B()	
	Set16k_hpBands_dec_B()	
DoWaveProc_B()		
	TeagerEng()	
	GetTeagerFilter()	
		GetMaximaPositions()
DoCompCeps_B()		
	CepsCompute()	
		Get16k_p_bufferCodeWeights_B()
		Get16k_p_bufferCodeForBands16k_B()
		PreEmphHamm()
		ff4NB16_B()
		GetBandsForDecoding16k_B()
		DecodeBands16k_B()
		FilterBank()
		Get16k_hpBands_dec_B()
		Get16k_p_hpBands_B()
		MergeSSandCoded_B()
		CorrectEnergy_B()
		CosInv16Khz()
		cosInv() (only for 8kHz)
DoPostProc_B()		
DoVADProc_F()		
	focalpoint()	

Table 2: VQ call structure

main()			
	quantize_and_print()		
		get_best_dataframe()	
			best_centroid()
		quant_pitch_abs()	
		get_class_bit()	
		quant_pitch_diff()	
		get_class_bit()	
		mfcc_crc_encode()	
		pc_crc_encode()	

Table 3: Extension call structure

main()							
	RVC_ConstructPitchRom_be()						
	RVC_ConstructPitchMeter_be()						
		Allocate_InterpolatedDft_be()					
		RVC_ResetPitchMeter_be()					
	RVC_DestructPitchRom_be()						
	RVC_DestructPitchMeter_be()						
		Deallocate_InterpolatedDft_be()					
	DoAdvProcess_B()						
		DoPitchExtract()					
			FilterBank()				
			dsr_afe_vad()				
				get_vm()			
					fnLog2()		
			IsLowBandNoise()				
			get_zcm()				
			pre_process()				
				iir_d()			
				iir_s()			
			RVC_MeasurePitch_be()				
				ClearPitch_be()			
				DirichletInterpolation_be()			
				IsLowLevelInput_be()			
				Finalize_be()			
					IsContinuousPitch_be()		
						Mpy_lw_sw()	
				Mpy_lw_sw()			
				PrepareSpectralPeaks_be()			
					CalcSpectrum_be()		
						Mpy_lw_sw()	
						Mpy_lw_sw_Add()	
					FindPeaks_be()		
					Prelim_ScaleDownAmpsOfHighFreqPeaks_be()		
					qsort_be()*		
						swap()	
					CompareIpointAmp_be()		
					RefineSpectralPeaks_be()		
						sqrt_l_fix()	
					Final_ScaleDownAmpsOfHighFreqPeaks_be()		
					Mpy_lw_sw()		
				FindPitchCandidates_be()			
					NormalizeAmplitudes_be()		
					CalcUtilityFunction_be()		
						CreatePieceWiseConstantFunction_be()	
							L_Extract()
							Mpy_32_16()
						qsort_be()*	
							swap()
						Compare_ARRAY_OF_XPOINTS_be()	
						LinkArrayOfPoints_be()	
						AddSortedArrayOfPoints_be()	
							LinkArrayOfPoints_be()
						ConvertLinkedListOfDiffPointsToUtilFunc_be()	
					FindDominantLocalMaximaInUtilityFunction_be()		
						Mpy_lw_sw()	
					UtilityFunctionAtGivenPitchFreq_be()		
					qsort_be()*		
						swap()	
					ComparePitchFreqAscending_be()		
					SelectTopPitchCandidates_be()		
						Mpy_lw_sw()	
					compute_pcorr_be()		

			interpolate_be()	
				Mpy_lw_sw()
				Mpy_lw_lw()
				sqrt_l_fix()
			find_most_energetic_window_be()	
			accumulate_be()	
			find_most_energetic_window2_be()	
			Mpy_lw_sw()	
	SelectFinalPitch_be()			
		qsort_be()*		
			swap()	
		ComparePitchFreqDescending_be()		
		ClearPitch_be()		
		GOOD_ENOUGH_be()		
		CLOSELY_LOCATED_be()		
			Mpy_lw_sw()	
		BETTER_be()		
		IsContinuousPitch_be()		
			Mpy_lw_sw()	
	CalculateDoubleWindowDft_be()			
classify_frame()				

* qsort_be() is a recursive function

4.5 Variables, constants and tables

The data types of variables and tables used in the fixed point implementation are signed integers in 2's complement representation, defined by:

- **Word16** 16 bit variable;
- **Word32** 32 bit variable.

4.5.1 Description of constants used in the C-code

Table 5a: Global constants for AFE

Constant	Value	Description
NS_SPEC_ORDER_16K	64	Noise suppression Array length
NS_HANGOVER_16K	15	Noise suppression hangover count
NS_MIN_SPEECH_FRAME_HANGOVER_16K	4	Noise suppression minmum speech frame hangover count
NS_ANALYSIS_WINDOW_16K	80	Noise suppression analysis window
PERC_CODED	0.7	lambda merge (empirically set constant)
LAMBDA_NSE16k	0.99	Noise estimation Lambda
NS_NB_FRAME_THRESHOLD_NSE	100	Noise suppression number of frame threshold used for NSE
LENGTH_QMF	118	QMF filter length
f24	1	multiplier for QMF filter coefficients
SHFF_H	8	shift to get higher value
L_H	16	shift to get lower value
HP16k_MEL_USED	3	Higher frequnecy band Mel used
NB_LP_BANDS_CODING	3	Lower frequency band used in coding
NE16k_FRAMES_THRESH	100	Noise estimation frames threshold
NB_TOPOSTPROC	12	Number of coefficients to postprocess
CEP_FRAME_LENGTH	200	Frame length for cepstral coefficients
CEP_NB_COEF	13	Number of cepstral coefficients (including c0)
CEP_NB_CHANNELS	23	Number of filters used for cepstral coefficients
CEP_FFT_LENGTH	256	FFT length for cepstral coefficients
FRAME_BUF_SIZE	241	Denoised Output buffer size
FRAME_SHIFT	80	WaveProcessing input frame shift
FRAME_LENGTH	200	WaveProcessing frame size
NS_SPEC_ORDER	65	Noise suppression array length (8khz)
NS_BUFFER_SIZE	180	Noise suppression past frame size
NS_FRAME_SHIFT	80	Noise suppression input frame shift
NS_HALF_FILTER_LENGTH	8	Noise suppression filter half size
NS_NB_FRAME_THRESHOLD_LTE	10	Noise suppression long term energy forgetting factor threshold (in frames)
NS_NB_FRAME_THRESHOLD_NSE	100	Noise suppression spectrum estimate forgetting factor threshold (in frames)
NS_MIN_FRAME	10	Number of frame threshold to update average energy for Nosie suppression VAD
NS_FFT_LENGTH	256	FFT length for noise suppression
WF_MEL_ORDER	25	Noise suppression Wiener filter order
SHFT_NOISE	14	shift applied to noise spectrum estimate
SHFT_FACT_MUL	14	shift applied to gain coefficient (nosie suppression gain factoriization)
IDCT_ORDER	25	Noise suppression idct order
NS_BETA	0.98	Noiseless signal suppression factor
NS_RSB_MIN	0.079432823	Minimum a priori SNR
NS_LAMBDA_NSE	0.99	Forgetting factor for noise spectrum estimate
NS_LOG_SPEC_FLOOR	-10.0	average energy minimum threshold
NS_SNR_THRESHOLD_VAD	15	SNR threshold for noise suppression VAD
NS_SNR_THRESHOLD_UPD_LTE	20	Long term energy update threshold for noise suppression VAD
NS_ENERGY_FLOOR	80	Energy Minimum threshold for noise suppression VAD
MaxPos	10	Maximum number of maxima in waveprocessing
WP_EPS	0.2	weigthing value added or substracted for waveprocessing

Table 5b: Global constants for VQ

Constant	Value	Description
MIN_PERIOD	1245184	Minimum pitch period allowed
MAX_PERIOD	9175040	Maximum pitch period allowed
NUM_MULTI_LEVELS_1	26	number of levels in pitch quantization
NUM_MULTI_LEVELS_2	24	number of levels in pitch quantization
UNVOICED_CODE	0	init value for Qpindex

Table 5c: Global constants for Extension

Constant	Value	Description
HISTORY_LEN	100	History length - past samples for pitch extraction
DOWN_SAMP_FACTOR	4	Down-sampling factor - used in computing correlation
NO_OF_DFT_POINTS	128	Number of DFT points
BREAK_POINT	12	Break point - marks the end of low frequency band
LBN_HIST_WEIGHT	32440	Low band noise history weight
LBN_CURR_WEIGHT	328	Low band noise current weight (32768 - LBN_HIST_WEIGHT)
LBN_MAX_THR	124518	Low band noise maximum threshold
LBN_LOW_ENR_LEVEL_MANT	32000	Low band noise low energy level mantissa
LBN_LOW_ENR_LEVEL_SHFT	22	Low band noise low energy level shift
RVC_OK	0	Return code for success
RVC_ERR	-1	Return code for unspecified error
RVC_ERR_NOT_ENOUGH_MEMORY	-2	Return code for not enough memory
RVC_ERR_ILLEGAL_ARGUMENT	-3	Return code for an illegal input / output argument
RVC_ERR_IO_FAILED	-4	Return code for failed input / output to a file
RVC_ERR_BAD_FILE_FORMAT	-5	Return code for a bad file header
RVC_ERR_NOT_INITIALIZED	-6	Return code for failure due to improper initialization
RVC_ERR_ILLEGAL_USAGE	-7	Return code for illegal usage of a function
RVC_ERR_NOT_ENOUGH_SAMPLES	-8	Return code for insufficient number of samples
RVC_ERR_NOT_IMPLEMENTED	-9	Return code for an unimplemented function

RVC_ERR_FAIL_OPEN_FILE	-10	Return code for failure to open a file
UB_ENRG_FRAC	59	Upper band energy fraction
ZCM_THLD	87	Zero crossing measure threshold
SQRT_ONE_HALF	0x5A82	Square root of 0.5 (0.707)
FRAME_LEN_DS	50	Frame length downsampled (200/4)
FRAME_LEN_DS_BY_2	25	Frame length downsampled divided by 2
HISTORY_LEN_DS	25	History length downsampled (100/4)
WINDOW_LENGTH	18	Window length used in computing correlation
INV_WINDOW_LENGTH	1820	Inverse of window length (1/18 = 0.05556)
NUM_CHAN	23	Number of channels or Mel-frequency bands
MIN_CH_ENRG_MANTISSA	20000	Minimum channel energy mantissa
MIN_CH_ENRG_SHIFT	25	Minimum channel energy shift
INIT_SIG_ENRG_MANTISSA	30518	Initial signal energy mantissa
INIT_SIG_ENRG_SHIFT	8	Initial signal energy shift
CE_SM_FAC	18022	Channel energy smoothing factor
CE_SM_FAC_COMPL	14746	Channel energy smoothing factor complement
CNE_SM_FAC	3277	Channel noise energy smoothing factor
CNE_SM_FAC_COMPL	29491	Channel noise energy smoothing factor complement
LO_GAMMA	22938	Low gamma value
LO_GAMMA_COMPL	9830	Low gamma value complement
HI_GAMMA	29491	High gamma value
HI_GAMMA_COMPL	3277	High gamma value complement
LO_BETA	31130	Low beta value
HI_BETA	32702	High beta value
INIT_FRAMES	10	Initial number of frames (considered to be noise frames)
SINE_START_CHAN	4	Sine start channel (for sine wave detection)
PEAK_TO_AVE_THLD	10	Peak to average threshold
DEV_THLD	1523942	Deviation threshold
HYSTER_CNT_THLD	9	Hysteresis count threshold
F_UPDATE_CNT_THLD	500	Forced update count threshold
NON_SPEECH_THLD	32	Non-speech threshold
FIX_34	24576	(short) (32768.0 * 3.0/4.0)
FIX_18	4096	(short) (32768.0 * 1.0/8.0)
FIX_INVSQRT2	-23170	1 / sqrt(2)
swTHIRD_REF_BANDWIDTH	85	One third of the reference bandwidth
swTWO_THIRDS_REF_BANDWIDTH	171	Two thirds of the reference bandwidth
MIN_ENERGY_MANTISSA	25600	Minimum energy mantissa
MIN_ENERGY_SHIFT	18	Minimum energy shift
swREF_SAMPLE_RATE_Q0	0x1F40	Reference sampling rate in Q0 format
swCLOSE_FACTOR_Q14	0x4CCD	Closeness factor in Q14 format
swFD_SCORE_THLD1_Q15	0x63D7	Frequency domain score threshold 1 in Q15 format
swFD_SCORE_THLD2_Q15	0x570A	Frequency domain score threshold 2 in Q15 format
swCORR_THLD_Q15	0x651F	Correlation threshold in Q15 format
swSUM_THLD_Q14	0x6667	Sum threshold in Q14 format
lwCRIT0_OFFSET_Q15	0x0000170A	Offset for finding a better pitch candidate in Q15 format
swCANDCORR_THLD1_Q15	0x799A	Pitch candidate correlation threshold 1 in Q15 format
swCANDCORR_THLD2_Q15	0x599A	Pitch candidate correlation threshold 2 in Q15 format
swCANDCORR_THLD3_Q15	0x6CCD	Pitch candidate correlation threshold 3 in Q15 format
swCANDAMP_THLD3_Q15	0x68F6	Pitch candidate amplitude threshold 3 in Q15 format
swSTARTFREQ_COEFF	0x553F	Start frequency coefficient (for candidate search)
swENDFREQ_COEFF	0x4666	End frequency coefficient (for candidate search)
DIRICHLET_KERNEL_SPAN	8	Direchlet kernal span (for interpolation)
REF_SAMPLE_RATE	8000	Reference sampling rate
REF_BANDWIDTH	4000	Reference bandwidth
lwTHIRD_REF_BANDWIDTH	87381333	One third of the reference bandwidth
lwTWO_THIRDS_REF_BANDWIDTH	174762667	Two thirds of the reference bandwidth
swCENTER_WEIGHT	0x5000	Center weight
swSIDE_WEIGHT	0x1800	Side weight
swAMP_SCALE_DOWN1	0x5333	Amplitude scale down factor 1
swAMP_SCALE_DOWN2	0x399A	Amplitude scale down factor 2
swAMP_SCALE_DOWN2b	0x7333	Amplitude scale down factor 2b
swUDIST1	-4160	Utility function distance 1
swUDIST2	-6400	Utility function distance 2
swUSTEP	-16384	Utility function step
swFREQ_MARGIN1	0x4AE1	Frequency margin 1
swAMP_MARGIN1	0x07AE	Amplitude margin 1
swAMP_MARGIN2	0x07AE	Amplitude margin 2
MIN_STABLE_FRAMES	6	Minimum number of stable frames
MAX_TRACK_GAP_FRAMES	2	Maximum pitch track gap frames
swSTABLE_FREQ_UPPER_MARGIN	0x4E14	Stable frequency upper margin
swSTABLE_FREQ_LOWER_MARGIN	0x68EB	Stable frequency lower margin
UNVOICED	0	Pitch frequency of an unvoiced frame
lwMAX_PITCH_FREQ	0x01A40000L	Maximum pitch frequency
lwMIN_PITCH_FREQ	0x00340000L	Minimum pitch frequency
MAX_PITCH_FREQ	420	Maximum pitch frequency in Hz
MIN_PITCH_FREQ	52	Minimum pitch frequency in Hz
HIGHPASS_CUTOFF_FREQ	300	Highpass cut-off frequency in Hz
NO_OF_FRACS	77	Number of fractions in the frations table
lwSHORT_WIN_START_FREQ	0x00C80000L	Short window start frequency
lwSHORT_WIN_END_FREQ	0x01A40000	Short window end frequency
lwSINGLE_WIN_START_FREQ	0x00640000L	Single window start frequency
lwSINGLE_WIN_END_FREQ	0x00D20000L	Single window end frequency
lwDOUBLE_WIN_START_FREQ	0x00340000	Double window start frequency
lwDOUBLE_WIN_END_FREQ	0x00780000L	Double window end frequency
MAX_LOCAL_MAXIMA_ON_SPECTRUM	70	Maximum number of local maxima on the spectrum
MAX_PEAKS_FOR_SORT	30	Maximum number peaks for sorting
MAX_PEAKS_PRELIM	7	Maximum number of peaks (preliminary)
MIN_PEAKS	7	Minimum number of peaks
MAX_PEAKS_FINAL	20	Maximum number of peaks (final)
MAX_PRELIM_CANDS	4	Maximum number of preliminary candidates (pitch)
CREATE_PIECEWISE_FUNC_LOOP_LIM_SH	20	Create Piecewise function loop limit for short window
CREATE_PIECEWISE_FUNC_LOOP_LIM_SNG	30	Create Piecewise function loop limit for single window
CREATE_PIECEWISE_FUNC_LOOP_LIM_DBL	60	Create Piecewise function loop limit for double window

swSUM_FRACTION	0x799A	Sum fraction
swAMP_FRACTION	0x33F8	Amplitude fraction
MAX_BEST_CANDS	2	Maximum number of best candidates (pitch)
N_OF_BEST_CANDS_SHORT	2	Number of best candidates for short window
N_OF_BEST_CANDS_SINGLE	2	Number of best candidates for single window
N_OF_BEST_CANDS_DOUBLE	2	Number of best candidates for double window
N_OF_BEST_CANDS	6	Number of best candidates for all windows
SIZE_SCRATCH_DOPITCH	1090	Scratch memory size for DoPitch() function (This is the actual size required. The declared size in C simulation is 1632)
SIZE_SCRATCH_ADVPROCESS	825	Scratch memory size for DoAdvProcess() function (This is the actual size required. The declared size in C simulation is 1100)
RVC_PITCH_ROM_SIG	11031	Signature for RVC_PITCH_ROM structure
RVC_PITCH_METER_SIG	21053	Signature for RVC_PITCH_METER structure

4.5.2 Description of fixed tables used in the C-code

This section contains a listing of all fixed tables sorted by source file name and table name. All table data is declared as **Word16**.

Table 6a: Fixed tables for AFE

File	Table Name	Length	Description
16kHzProcessing_B.c	table_pow2	33	Table for square root
	LambdaNSEx2	100	Table used to compute first 100 LambdaNSE
	dp02_h	59	MSB of QMF filter coefficients
	dp02_l	43	LSB of QMF filter coefficients
PostProc_B.c	targetLMS16	12	Target for blind equalization
ComCeps_B.c	HalfHamming16	100	Hamming window coefficients
	CosMatrix16	144	Inverse cosinus coefficients at 8Khz (not used at 16khz)
	CosMatrix16_16khz	156	Inverse cosinus coefficients at 16Khz
	pondMelFilter	309	Mel bank coefficients
ff4nrFix16_B.c	tabSin	64	Sine table
	tabCos	64	Cosine table
MathFunc.c	tbInt0	48	Coefficients for computation of square root
ExtNoiseSup_B.c	lambda_1divX	20	Computation of 1/N
	Hann_sh32_hi	100	MSB of hanning window coefficients (32 bits)
	Hann_sh32_lo	100	LSB of hanning window coefficients (32 bits)
	Hann_sh24_hi	100	MSB of hanning window coefficients (24 bits)
	Hann_sh24_lo	100	LSB of hanning window coefficients (24 bits)
	pondMelFilterNoise	157	Mel-frequency scale coefficients (applied to the Wiener filter)
	idctMel16	234	Mel-warped inverse DCT coefficients
	pondMelFilter16k	134	Filter bank coefficients at 16Khz
	M1_LamdaLTE	8	Computation of 1/N
	M1_LambdaNSEx2	100	Computation of 2/N
	M1_LamdaNSE	9	Computation of 1/N
	mInvLambda16	10	Comutation od 2/N

Table 6b: Fixed tables for VQ

File	Table Name	Length	Description
coder_VAD.c	quantizer16kHz_0_1	128	vq table
	quantizer16kHz_2_3	128	vq table
	quantizer16kHz_4_5	128	vq table
	quantizer16kHz_6_7	128	vq table
	quantizer16kHz_8_9	128	vq table
	quantizer16kHz_10_11	64	vq table
	quantizer16kHz_12_13	512	vq table
	quantizer8kHz_0_1	128	vq table
	quantizer8kHz_2_3	128	vq table
	quantizer8kHz_4_5	128	vq table
	quantizer8kHz_6_7	128	vq table
	quantizer8kHz_8_9	128	vq table
	quantizer8kHz_10_11	64	vq table
	quantizer8kHz_12_13	512	vq table
	weight16kHz_c0_shift	1	vq weights
	weight16kHz_c0_norm	1	vq weights
	weight16kHz_logE	1	vq weights
	weight8kHz_c0_shift	1	vq weights
	weight8kHz_c0_norm	1	vq weights
	weight8kHz_logE	1	vq weights
	plwQuantLevels[127]	127*2	vq tables for pitch/class quantization
	ppplwQuantSections[8][3]	24*2	vq tables for pitch/class quantization
	plwQuantLevels[31]	31*2	vq tables for pitch/class quantization
	pplwQuantSections[4][3]	12*2	vq tables for pitch/class quantization
	pswRatioThld_1[4][6]	24	vq tables for pitch/class quantization
	piMultiLevelIndex[4]	4	vq tables for pitch/class quantization
	pswRatioThld_2[4][8]	32	vq tables for pitch/class quantization
	piMultiLevelIndex_2[4]	4	vq tables for pitch/class quantization
	swAlpha1	1	pitch/class constants
	swAlpha2	1	pitch/class constants

Table 6c: Fixed Tables for Extension

File	Table name	Length	Description
ExtNoiseSup_B.c	pswPePower	129	Coefficients to compute the pre-emphasis power spectrum
preProc_B.c	pswHpfCoef	15	High pass filter coefficients
preProc_B.c	pswLpfCoef	15	Low pass filter coefficients
preProc_B.c	pswLfeCoef	3	Low frequency emphasis filter coefficients
dsrAfeVad_B.c	piBurstConst	20	Burst length constants for different SNR's
dsrAfeVad_B.c	piHangConst	20	Hang length constants for different SNR's
dsrAfeVad_B.c	piVADThld	20	VAD voice metric thresholds for different SNR's
dsrAfeVad_B.c	piVMTable	90	Voice metric table as a function of SNR index
dsrAfeVad_B.c	piSigThld	20	Signal threshold table as a function of SNR
dsrAfeVad_B.c	piUpdateThld	20	Update threshold table as a function of SNR
dsrAfeVad_B.c	pswShapeTable	23	Spectral shape correction table
fix_mathlib.c	coeff_sqrt5_58	5	Coefficients for computation of square root
fix_mathlib.c	coeff_sqrt5_78	5	Coefficients for computation of square root
rvc_pitch_init_B.h	ROM_astFrac	312	Fractions table
rvc_pitch_init_B.h	ROM_pstWindowshiftTable	514	Complex exponents table for time shifting in frequency domain
rvc_pitch_init_B.h	ROM_aswDirichletImag	8	Imaginary part of the Dirichlet kernel

4.5.3 Static variables used in the C-code

In this section two tables that specify the static variables for the AFE, VQ, and Extension respectively are shown.

Table 7a: AFE static variables

Struct Name	Variable	Type[Length]	Description
QMF_FIR			
	lengthQMF	Word32	QMF Filter length
	*dp_l	Word16	QMF filter low frequency Coeff
	*dp_h	Word16	QMF filter high frequency Coeff
	*T	Word16	Temporary QMF filter buffer
	T_dec	Word16	Multiplier for T
DataFor16kProc_B			
	FrameLength	Word32	Input Frame length
	FrameShift	Word32	Shift value for the frame
	numFramesInBuffer	Word32	Number of frames in buffer
	SamplingFrequency	Word32	Sampling frequency (8/16)
	Do16kHzProc	BOOLEAN	Flag to enable 16kHz processing
	*hpBands_B	Word32	Buffer for HP bands
	hpBandsSize	Word32	hpBands_B buffer size
	CodeForBands16k_B	Word32[9]	HP coding buffer
	bufferCodeForBands16k_B	Word32[27]	buffer used for HP coding
	codeWeights_B	Word16[3]	code Weights buffer
	bufferCodeWeights_B	Word16[9]	buffer used for code Weights
	* pQMF_Fir	QMF_FIR	Pointer to QMF_FIR structure
	*bufferData16k_B	Word32	temporary buffer to carry QMF LP data
	bufData16kSize	Word32	16k data buffer size
	*FirstWindow16k	MelFB_Window	pointer to MelFB_Window structure
	noiseSE16k_B	Word32[3]	noise spectrul energy variable
	noise_dec	Word16	Multiplier for noiseSE16k_B
	BandsForCoding16k_B	Word32[9]	buffer for storing Bands for Coding
	vadCounter16k	Word32	vad flag counter
	vad16k	Word32	vad flag
	nbSpeechFrames16k	Word32	number of speech frames counter
	hangOver16k	Word32	hang over used for VAD
	meanEn16k	Word32	mean Energy variable
	nb_frame_threshold_nse	Word32	threshold NSE for frame
	lambda_nse	Word16	lambda NSE variable
	*dataHP_B	Word32	buffer stores QMF HP value
	dec_16k	Word16[5]	Multiplier for dataHP_B buffer
	BFC_dec	Word16[1]	Multiplier for computing bands for coding
	fb16k_dec	Word16[3]	Buffer is used to store multiplier for current and pervious two frames
PostProcStructX			
	weightLMS	Word32[12]	Current LMS weight
CompCepsStructX			
	FFTLength	Word32	FFT size
	Do16khzProc	Word16	Flag to enable 16kHz processing
	*pData16k	Word32	Pointer to data for 16Khz processing
WaveProcStructX			
	*TeagerFilter16	Word32	Pointer to teager filter
	*TeagerWindow32	Word32	Pointer to teager window
	TeagerOnset	Word32	Unused
	FrameLength	Word32	Input frame length
ns_var_F			
	SampFreq	Word16	Sampling frequency (8/16)
	Do16khzProc	Word16	Flag to enable 16kHz processing
	buffers.nbFramesInFirstStage	Word32	number of frames in first stage
	buffers.nbFramesInFirstStage	Word32	number of frames in second stage
	buffers. nbFramesOutSecondStage	Word32	number of frames out og second stage
	buffers. FirstStageIn16Buffer	Word16[180]	First stage buffer
	buffers.SecondStageInBuffer32	Word32[180]	Second stage buffer
	buffers. SecondDecalSig	Word16[4]	Shift factor for each sub-frame of second stage buffer
	prevSamples32.lastSampleIn32	Word32	Last input sample of DC offset compensation
	prevSamples32.lastDCOut32	Word32	last output sample of DC offset compensation
	prevSamples32. oldShift	Word16	lprevious window shift factor of DC offset compensation
	spectrum.indexBuffer1	Word16	Where to enter new PSD for first stage, alternatively 0 and 1
	spectrum.indexBuffer2	Word16	Where to enter new PSD for second stage, alternatively 0 and 1
	spectrum.noiseSE1_32	Word32[65]	Noise spectrum estimate for first stage
	spectrum.noiseSE1_dec	Word16[65]	Shift factor for Noise spectrum estimate (first sage)
	spectrum.noiseSE2_32	Word32[65]	Noise spectrum estimate for second stage
	spectrum.noiseSE2_dec	Word16[65]	Shift factor for Noise spectrum estimate (second sage)
	spectrum.PSDMeanAntBuffer1	Word32[65]	1st stage PSD Mean buffer for precedent frame
	spectrum.nSigSE1Ant_dec	Word16[65]	Shift factor for PSD Mean buffer for precedent frame (1rst stage)
	spectrum.PSDMeanAntBuffer2	Word32[65]	2nd stage PSD Mean bufferfor precedent frame
	spectrum.nSigSE2Ant_dec	Word16[65]	Shift factor for PSD Mean buffer for precedent frame (2nd stage)
	spectrum.denSigSE1_32	Word32[65]	1st stage PSD Mean buffer
	spectrum. nSigSE1Cur_dec	Word16[65]	Shift factor for PSD Mean buffer (1rst stage)
	spectrum. denSigSE2_32	Word32[65]	2nd stage PSD Mean buffer
	spectrum. nSigSE2Cur_dec	Word16[65]	Shift factor for PSD Mean buffer (2^{nd} stage)
	vad_data_ns_F. nbFrame	Word16[2]	Nubmer of frames (for the 2 stages)
	vad_data_ns_F. flagVAD	Word16	Vad Flag (1 = SPEECH, 0 = NON SPEECH)
	vad_data_ns_F.hangOver	Word16	hangover
	vad_data_ns_F. nbSpeechFrames	Word16	Number of speech frames (used to set hangover)
	vad_data_ns_F.meanEn32	Word32	Mean energy for VAD
	vad_data_ca. flagVAD	Word16	Vad Flag (1 = SPEECH, 0 = NON SPEECH)
	vad_data_ca.hangOver	Word16	hangover
	vad_data_ca. nbSpeechFrames	Word16	Number of speech frames (used to set hangover)
	vad_data_ca.meanEn32	Word32	Mean energy for VAD
	vad_data_fd.MelMean	Word16	SpeechQMel (for frame dropping)
	vad_data_fd.VarMean	Word32	SpeechQVar (for frame dropping)

	vad_data_fd.AccTest	Word32	SpeechQSpec (for frame dropping)
	vad_data_fd.AccTest2	Word32	
	vad_data_fd.SpecMean	Word32	SpecMean (for frame dropping)
	vad_data_fd.MelValues	Word16[2]	SpeechQMel (for frame dropping)
	vad_data_fd.SpecValues	Word32	SpeechQSpec (for frame dropping)
	vad_data_fd.SpeechInVADQ	Word16	Flag (for frame dropping)
	vad_data_fd.SpeechInVADQ2	Word16	Flag (for frame dropping)
	gainFact.logDenEn1_32	Word32[3]	Denoise frame energy for gain factorization
	gainFact.lowSNRtrack32	Word32	Low SNR level for gain factorization
	gainFact. alfaGF16	Word16	Wiener filter gain factorization coefficient
VADStructX_F			
	Focus	Word16	Position of circular buffe
	HangOver	Word16	Hangover length
	FlushFocus	Word16	Position in circular buffer when emptying at end
	H_CountDown	Word16	Main hangover countdown
	V_CountDown	Word16	Short hangover countdown
	**OutBuffer	Word32	outBuffer pointer pointer
	*OutBuffer	Word32[7]	outBuffer pointer
	OutBuffer	Word16[7x15]	outBuffer

Table 7b: VQ static variables

Struct Name	Variable	Type [Length]	Description
coder_VAD.c	four_frames[27]	Word16[27]	Previous frames used to build multiframe
	plwQPHistory[3]	Word32[3]	History of Pitch
	lReliableFlag	Word16	Pitch reliability flag

Table 7c: Extension static variables

Struct Name	Variable	Type[Length]	Description
	iFirstFrameFlag	Word16	First frame flag
	pswUBSpeech	Word16[200]	Upper band speech
	pswDownSampledProcSpeech	Word16[75]	Down-sampled processed speech
	lwCritMax	Word32	Maximum power ratio
	iOldPitchPeriod	Word16	Old pitch period value
	iOldFrameNo	Word16	Old frame number
PCORR_STATE_be	s_be		
	lwX1_X1	Word32	X1*X1
	lwZ1_Z1	Word32	Z1*Z1
	lwZ2_Z2	Word32	Z2*Z2
	lwX1_Z1	Word32	X1*Z1
	lwX1_Z2	Word32	X1*Z2
	lwZ1_Z2	Word32	Z1*Z2
	swX1_Sum	Word16	Sum of X1
	swZ1_Sum	Word16	Sum of Z1
	swZ2_Sum	Word16	Sum of Z2
	iBurstConst	Word16	Burst constant
	iBurstCount	Word16	Burst count
	iHangConst	Word16	Hang constant
	iHangCount	Word16	Hang count
	iVADThld	Word16	VAD threshold
	iFrameCount	Word16	Frame count
	iFUpdateFlag	Word16	Forced update flag
	iHysterCount	Word16	Hysteresis count
	iLastUpdateCount	Word16	Last update count
	iSigThld	Word16	Signal threshold
	iUpdateCount	Word16	Update count
	iChanEnrgShift	Word16	Channel energy shift
	iChanNoiseEnrgShift	Word16	Channel noise energy shift
	pswChanEnrg	Word16[23]	Channel energy
	pswChanNoiseEnrg	Word16[23]	Channel noise energy
	swBeta	Word16	Beta value
	swSnr	Word16	SNR value
NormSw	pnsLogSpecEnrgLong		
	swMantissa	Word16[23]	Mantissa
	iShift	Word16[23]	Shift
	swC0	Word16	C0 value
	swC1	Word16	C1 value
	swC2	Word16	C2 value
	pswHpfXState	Word16[6]	High pass filter input state
	pswHpfYState	Word16[12]	High pass filter output state
	pswLpfXState	Word16[6]	Low pass filter input state
	pswLpfYState	Word16[12]	Low pass filter output state
	pswLfeXState	Word16	Low frequency emphasis filter input state
	pswLfeYState	Word16[2]	Low frequency emphasis filter output state

5 File formats

This section describes the file formats used by the AFE, VQ & Extension programs.

5.1 Speech file

Speech files read by the X-AFE and written by the Extension consist of 16-bit words. The byte order depends on the host architecture (e.g. MSByte first on SUN workstations, LSByte first on PCs etc)

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2004-06	24	SP-040343			Version 6.0.0 approved at 3GPP TSG SA#24	2.0.0	6.0.0
2004-12	26	SP-040837	001	1	Software bug correction: Removal of Basicops simulation of "C" shift operator	6.0.0	6.1.0
2004-12	26	SP-040837	002	1	Software bug correction: Initialization of the variables lwc and i2aScale	6.0.0	6.1.0
2004-12	26	SP-040837	003	1	Software bug correction: Wrong assignment of the variables *piReliableFlag and *pcQPIndex	6.0.0	6.1.0
2004-12	26	SP-040837	004	2	Software bug correction: Use of incorrect variable fRefPeriod instead of iRefPeriod	6.0.0	6.1.0
2004-12	26	SP-040837	005		Add reference to test sequences document	6.0.0	6.1.0
2007-06	26				Version for Release 7	6.1.0	7.0.0
2008-12	42				Version for Release 8	7.0.0	8.0.0
2009-12	46				Version for Release 9	8.0.0	9.0.0
2011-03	51				Version for Release 10	9.0.0	10.0.0

History

Document history		
V10.0.0	April 2011	Publication