INTERNATIONAL TELECOMMUNICATION UNION

ITU-T G.722.2 Implementers Guide

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(25 October 2002)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Digital terminal equipments – Coding of analogue signals by methods other than PCM

Implementors' Guide for G.722.2

(Wideband coding of speech at around 16 kbit/s using Adaptive Multi-rate Wideband, AMR-WB)

Implementers Guide for Recommendation G.722.2

Contact Information

Rapporteur, ITU-T Study Group 16 / Question 7	Rosario Drogo De Iacovo Research & Innovation TILAB S.p.A. Via G. Reiss Romoli, 274 - Torino Italy	Tel: +39-011-228-6221 Fax: +39-011-228-7056 E-mail: rosario.drogodeiacovo@tilab.com
Editor, ITU-T Recommendation G.722.2 and Implementors' Guide	Jari Hagqvist Nokia Research Center Visiokatu 1 33720 Tampere Finland	Tel: +358 50 4835459 Fax: +358 7180 35888 E-mail: jari.hagqvist@nokia.com

SUMMARY

Implementors' Guide for Recommendation G.722.2

The changes that appear in this document are necessary to correct defects identified with the standard as well as to keep alignment of the text with the related 3GPP Recommendation, as reported and approved at SG 16's meeting on 15-25 October 2002.

This revision of the document contains a few editorial changes (contacts, misspellings) but added a missing correction to section 3.3 (list of acronyms).

Table of Contents

1 INTRODUCTION **1**

2 SCOPE **1**

3 DEFECT RESOLUTION PROCEDURE **1**

4 REFERENCES **1**

5 NOMENCLATURE **1**

6.1 TECHNICAL AND EDITORIAL CORRECTIONS TO ITU-T RECOMMENDATION G.722.2 (2002) 2

6.1.1 G.722.2 Section 5.4 ..... *2*

6.1.2 G.722.2 Section 5.7 ..... *2*

6.1.3 G.722.2 Section 5.8.1 ..... *2*

6.1.4 G.722.2 Section 5.8.2 ..... *3*

6.1.5 G.722.2 Section 5.8.3 ..... *3*

6.1.6 G.722.2 Section 5.9 ..... *3*

6.1.7 G.722.2 Section 5.10 ..... *3*

6.1.8 G.722.2 Section 2 ..... *4*

6.1.9 G.722.2 Figure 2 and 3 ..... *4*

6.1.10 G.722.2 Section 6.1 ..... *5*

6.1.11 G.722.2 Appendix I changes *5*

6.1.12 G.722.2 Section 1 and 10 ..... *6*

6.1.13 G.722.2 Annex C C-code changes *9*

6.1.14 Other changes to G.722.2 Annex C C-code *16*

1 Introduction

This document is a compilation of reported defects identified with the 2002 decided edition of ITU-T G.722.2 Recommendation. It must be read in conjunction with the Recommendation to serve as an additional authoritative source of information for implementers. The changes, clarifications and corrections defined herein are expected to be included in future versions of affected G.722.2 -series Recommendations.

2 Scope

This guide resolves defects in the following categories:

- editorial errors
- technical errors, such as omissions and inconsistencies
- ambiguities

In addition, the Implementers Guide may include explanatory text found necessary as a result of interpretation difficulties apparent from the defect reports.

This Guide will not address proposed additions, deletions, or modifications to the Recommendations that are not strictly related to implementation difficulties in the above categories. Proposals for new features should be made in through contributions to the ITU-T.

3 Defect Resolution Procedure

Upon discovering technical defects with any components of the G.722.2 Recommendation, please provide a written description directly to the editors of the affected Recommendation with a copy to the Q7/16 Rapporteur. Return contact information should also be supplied so a dialogue can be established to resolve the matter and an appropriate reply to the defect report can be conveyed. This defect resolution process is open to anyone interested in Recommendation G.722.2. Formal membership in the ITU is not required to participate in this process.

4 References

This document refers to the following G.722.2 series Recommendations:

- ITU-T Recommendation G.722.2 (2002), *Wideband coding of speech at around 16 kbit/s using adaptive multi-rate wideband (AMR-WB)*
- ITU-T Recommendation G.722.2 – Appendix I (2002), *Error concealment of erroneous or lost frames*

5 Nomenclature

In addition to traditional revision marks, the following marks and symbols are used to indicate to the reader how changes to the text of a Recommendation should be applied:

Symbol	Description

[Begin Correction]	Identifies the start of revision marked text based on extractions from the published Recommendations affected by the correction being described.
[End Correction]	Identifies the end of revision marked text based on extractions from the published Recommendations affected by the correction being described.
...	Indicates that the portion of the Recommendation between the text appearing before and after this symbol has remained unaffected by the correction being described and has been omitted for brevity.
--- SPECIAL INSTRUCTIONS --- {instructions}	Indicates a set of special editing instructions to be followed.

6.1 Technical and Editorial Corrections to ITU-T Recommendation G.722.2 (2002)

6.1.1 G.722.2 Section 5.4

Description:	Editorial correction

[Begin Correction]

Depending on the mode, open-loop pitch analysis is performed once per frame (each ~~10~~20 ms)

[End Correction]

6.1.2 G.722.2 Section 5.7

Description:	Editorial correction

[Begin Correction]

windowed sinc functions); one for interpolating the term in Equation (~~34~~37) with the sinc truncated at ±17 and the other

[End Correction]

6.1.3 G.722.2 Section 5.8.1

Description:	Editorial correction

[Begin Correction]

domain by combining the filter $F(z)$ with the weighed synthesis filter prior to the co~~d~~de~~d~~book search.

[End Correction]

6.1.4 G.722.2 Section 5.8.2

Description:	Editorial corrections

[Begin Correction]

For cases 0 and ~~1~~4, I_{AB} is given by

...

...

where k is the index of the coupled case (2 bits), I_{3pB} is the index of 3 pulses in Section B (3(M-1)+1 bits), and I_{3pA} is the

[End Correction]

6.1.5 G.722.2 Section 5.8.3

Description:	Editorial correction

[Begin Correction]

Before searching the positions, the sign of ~~at~~ pulse at a potential position n is set the sign of $b(n)$ at that position. Then the

[End Correction]

6.1.6 G.722.2 Section 5.9

Description:	Editorial correction

[Begin Correction]

where $[b_1\ b_2\ b_3\ b_4]$=[0.5~~,~~0.4,0.3,0.2] are the MA prediction coefficients, and $\hat{R}(k)$ is the quantized energy prediction

[End Correction]

6.1.7 G.722.2 Section 5.10

Description:	Editorial correction

[Begin Correction]

Equation (~~54~~57) for $n = 48,\ldots,63$. This saves two filterings.

[End Correction]

6.1.8 G.722.2 Section 2

Description:	Editorial correction

[Begin Correction]

~~[1] ITU-T Recommendation G.711 (1988): "Coding of analogue signals by pulse code modulation Pulse code modulation (PCM) of voice frequencies".~~

[1] ITU-T Recommendation G.722 (1988): "7 kHz Audio-coding within 64 kbit/s".

[End Correction]

6.1.9 G.722.2 Figure 2 and 3

Description:	Editorial correction (only correct versions of Figure 2 and 3 presented)

[Begin Correction]

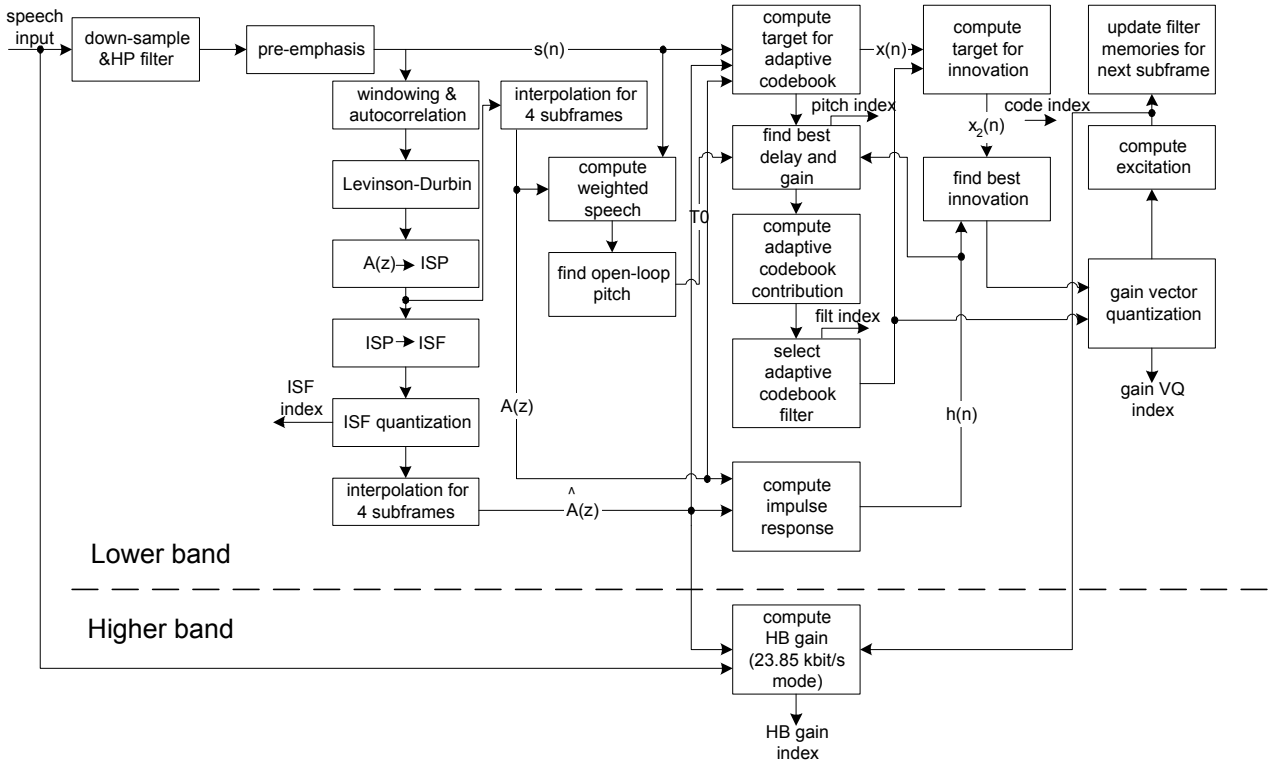

Figure 2 Detailed block diagram of the ACELP encoder

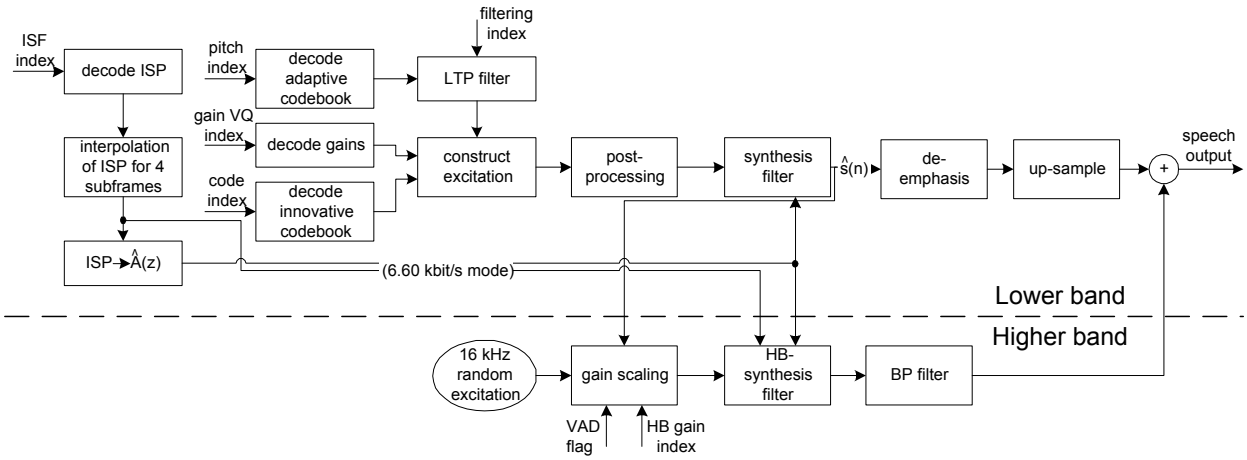

Figure 3 Detailed block diagram of the ACELP decoder

[End Correction]

6.1.10 G.722.2 Section 6.1

Description:	Editorial correction

[Begin Correction]

8. Post-processing of excitation elements (6.60 and 8.85 kbit/s mode): **A post-processing of excitation**

[End Correction]

6.1.11 G.722.2 Appendix I changes

Description:	Editorial correction (Add section I.5.2.5)

[Begin Correction]

I.5.2.5 High-band gain (for 23.85 kbit/s mode)

When RX_FRAMETYPE = SPEECH_BAD or RX_FRAMETYPE = SPEECH_LOST the received high-band energy parameter of the frame is not used and the estimation for the high-band gain is used instead. This means that in case of bad/lost speech frames, the high-band reconstruction operates in the same way for all the modes.

[End Correction]

6.1.12 G.722.2 Section 1 and 10

Description:	3GPP has reduced the number of mandatory AMR-WB codec modes for the speech telephony service in 3GPP (see COM 16 – D 287). Three active codec sets (Configuration A, B and C) have been defined. This should be observed when interoperability with 3GPP standards-based wireless networks is required. Add sentence to Section 1 with reference to Section 10 and insert new Section 10 (old Section 10 renumbered to Section 11).

Section 1 (add following sentence after last sentence in section 1):

[Begin Correction]

Section 10 provides information on minimum requirements for support of AMR-WB modes in 3GPP speech telephony service.

[End Correction]

Section 3.3 (add the following acronims):

[Begin Correction]

GERAN	GSM EDGE Radio Access Network
GMSK	Gaussian Modified Shift Keying
O-TCH/F	Octal TCH/Full rate, a 68.4 kbit/s gross bit-rate radio channel of GERAN-8PSK
O-TCH/H	Octal-TCH/Half rate a 32.4 kbit/s gross bit-rate radio channel of GERAN-8PSK
8PSK	8 Phase Shift Keying
TCH	Traffic CHannel (dedicated radio channel for speech or data.)
TCH/F	TCH/Full rate, a 22.8 kbit/s gross bit-rate radio channel of GERAN-GMSK
TCH/H	TCH/Half rate, an 11.4 kbit/s gross bit-rate radio channel of GERAN-GMSK
TFO	Tandem Free Operation (Tandem Free is achieved using in-band signalling after call set-up)
TrFO	Transcoder Free Operation (Tandem Free is achieved using out-of-band signalling before call set-up, the transcoders are in principle not in the communication path)
UTRAN	UMTS Terrestrial Radio Access Network

[End Correction]

Section 10 (renumber old section 10 to section 11).

New Section 10 text:

[Begin Correction]

10 Mandatory AMR-WB codec modes for the speech telephony service in 3GPP

This section should be observed when interoperability with 3GPP standards-based wireless networks is required.

To facilitate the implementation of AMR-WB for the circuit switched speech telephony service in 3GPP systems, the number of mandatory AMR-WB codec modes have been reduced to five. This allows for less complexity for the channel coding in terminals and networks. Following modes are used: 23.85, 15.85, 12.65, 8.85 and 6.60 kbit/s. Based on listening test results for the speech telephony service channels, these five modes are considered sufficient for high quality speech telephony service. For other services and applications in 3GPP all the 9 modes remain in use. All 9 source codecs are kept in the 3GPP AMR-WB codec, but the number of modes used for the speech telephony service is just limited to five.

In order to improve interoperability, the allowed AMR-WB codec mode configurations within active codec sets (ACS), i.e., which modes can be configured to be used within the mode adaptation at the same time, were further limited. The following three configurations are allowed: Configuration A (6.60, 8.85, 12.65), Configuration B (6.60, 8.85, 12.65, 15.85) and Configuration C (6.60, 8.85, 12.65, 23.85). This improves Tandem Free Operation / Transcoder Free Operation (TFO/TrFO) interoperability in 3GPP in the various speech service transmission channels because all these configurations are compatible. They contain the same three lower modes. This enables immediate establishment of TFO/TrFO (where double transcoding for calls between mobile terminals is avoided). This restriction for allowed mode configurations also simplifies signalling (e.g. in call set-up and handovers) and also drastically simplifies testing.

Table 14 summarises the requirements for support of AMR-WB mode configurations in 3GPP speech telephony service.

Table 14: Minimum requirements for AMR-WB modes in 3GPP speech telephony service

Speech telephony service channel	**Terminal**	**Network**
TCH/F	Configuration A (6.60, 8.85 and 12.65)	Configuration A (6.60, 8.85 and 12.65)
O-TCH/H	Configuration A (6.60, 8.85 and 12.65)	Configuration A (6.60, 8.85 and 12.65)
O-TCH/F	Configurations A (6.60, 8.85, 12.65), B (6.60, 8.85, 12.65, 15.85) and C (6.60, 8.85, 12.65, 23.85)	Configuration A (6.60, 8.85 and 12.65)*
UTRAN	Configurations A (6.60, 8.85, 12.65), B (6.60, 8.85, 12.65, 15.85) and C (6.60, 8.85, 12.65, 23.85)	Configuration A (6.60, 8.85 and 12.65) *

*) Support for configurations B and C is optional

~~10~~ 11 Bibliography (informative)

[End Correction]

6.1.13 G.722.2 Annex C C-code changes

Description:	Following changes to the G.722.2 C-code (v. 5.3.0) have been approved by Q7/16 to produce version 5.5.0. These changes will synchronize the G.722.2 C-code with v. 5.5.0 of the 3GPP AMR-WB C-code. The implementation of the C-code changes contained in this section of the Implementers' Guide on the current C-code available from the TSB can be found at the following URL (available after the 3GPP TSG-SA approval end of December, 2002): http://www.3gpp.org/ftp/Specs/archive/26_series/26.173/26173-550.zip, the test vectors are available at http://www.3gpp.org/ftp/Specs/archive/26_series/26.174/26174-530.zip

File: ***p_med_ol.c****, line 43 (Equal sign removed for correct operation of pitch search*

[Begin Correction]

```
for (i = L_max; i >= L_min; i--)
```

[End Correction]

File: dec_main.c, Input parameter *bfi* added into synthesis function and Bad frame substitution added for the mode 23.85 kbit/s

[Begin Correction]

Lines 54-64:

```
static void synthesis(
     Word16 Aq[],                          /* A(z)   : quantized Az               */
     Word16 exc[],                         /* (i)    : excitation at 12kHz        */
     Word16 Q_new,                         /* (i)    : scaling performed on exc   */
     Word16 synth16k[],                    /* (o)    : 16kHz synthesis signal     */
     Word16 prms,                          /* (i)    : parameter                  */
     Word16 HfIsf[],
     Word16 nb_bits,
     Word16 newDTXState,
     Decoder_State * st                    /* (i/o)  : State structure            */
     Word16 bfi                            /* (i)    : bad frame indicator        */
);

...
```

Lines 339-349:

```
for (i_subfr = 0; i_subfr < L_FRAME; i_subfr += L_SUBFR)
        {
            j = shr(i_subfr, 6);
            for (i = 0; i < M; i++)
            {
```

```
                L_tmp = L_mult(isf_tmp[i], sub(32767, interpol_frac[j]));
                L_tmp = L_mac(L_tmp, isf[i], interpol_frac[j]);
                HfIsf[i] = round(L_tmp);   move16();
            }
            synthesis(Aq, &exc2[i_subfr], 0, &synth16k[i_subfr * 5 / 4], (short) 1, HfIsf,
nb_bits, newDTXState, st, bfi);
}
```

…

Lines 970-975:

```
if (sub(nb_bits, NBBITS_24k) >= 0)
        {
            corr_gain = Serial_parm(4, &prms);
            synthesis(p_Aq, exc2, Q_new, &synth16k[i_subfr * 5 / 4], corr_gain, HfIsf,
nb_bits, newDTXState, st, bfi);
        } else
            synthesis(p_Aq, exc2, Q_new, &synth16k[i_subfr * 5 / 4], 0, HfIsf, nb_bits,
newDTXState, st, bfi);
```

Lines 1009-1019:

```
static void synthesis(
     Word16 Aq[],                          /* A(z)  : quantized Az               */
     Word16 exc[],                         /* (i)   : excitation at 12kHz        */
     Word16 Q_new,                         /* (i)   : scaling performed on exc   */
     Word16 synth16k[],                    /* (o)   : 16kHz synthesis signal     */
     Word16 prms,                          /* (i)   : parameter                  */
     Word16 HfIsf[],
     Word16 nb_bits,
     Word16 newDTXState,
     Decoder_State * st                    /* (i/o) : State structure            */
     Word16 bfi                            /* (i)   : bad frame indicator        */
)
...
```

Lines 1153-1171:

```
    if (sub(nb_bits, NBBITS_24k) >= 0 && (bfi == 0))
    {
        /* HF correction gain */
        HF_gain_ind = prms;
        HF_corr_gain = HP_gain[HF_gain_ind];

        /* HF gain */
        for (i = 0; i < L_SUBFR16k; i++)
        {
            HF[i] = shl(mult(HF[i], HF_corr_gain), 1);  move16();
```

```
        }
    } else
    {
        for (i = 0; i < L_SUBFR16k; i++)
        {
            HF[i] = mult(HF[i], tmp);       move16();
        }
    }
```

[End Correction]

File: bits.c, New function definition added to line 38

[Begin Correction]

```
Word16 Close_write_serial(TX_State *st);
{
   /* allocate memory */
    test();
    if (st != NULL)
    {
        free(st);
        st = NULL;
        return 0;
    }
    return 1;
}
```

[End Correction]

File: bits.h, New function reference added to line 52

[Begin Correction]

```
Word16 Close_write_serial(TX_State *st)
```

[End Correction]

File: cod_main.h, Unused memory allocation removed from lines 58, 60

[Begin Correction]

```
Word16 isfold[M];                       /* old isf (frequency domain) */
    Word32 L_gc_thres;                      /* threshold for noise enhancer */
    Word16 mem_syn_hi[M];                   /* modified synthesis memory (MSB) */
    Word16 mem_syn_lo[M];                   /* modified synthesis memory (LSB) */
    Word16 mem_deemph;                      /* speech deemph filter memory */
    Word16 mem_sig_out[6];                  /* hp50 filter memory for synthesis */
    Word16 mem_hp400[6];                    /* hp400 filter memory for synthesis */
    Word16 mem_oversamp[2 * L_FILT];        /* synthesis oversampled filter memory */
    Word16 mem_syn_hf[M];                   /* HF synthesis memory */
    Word16 mem_hf[2 * L_FILT16k];           /* HF band-pass filter memory */
    Word16 mem_hf2[2 * L_FILT16k];          /* HF band-pass filter memory */
    Word16 mem_hf3[2 * L_FILT16k];          /* HF band-pass filter memory */
    Word16 seed2;                           /* random memory for HF generation */
    Word16 disp_mem[8];                     /* phase dispersion memory */
    Word16 vad_hist;
```

[End Correction]

File: cod_main.c,

[Begin Correction]

Unused phase dispersion initialisation is removed from line 115

…

```
    Init_gp_clip(cod_state->gp_clip);

    cod_state->L_gc_thres = 0;              move16();
    Init_Phase_dispersion(cod_state->disp_mem);
```

…

Unused filter initialisation is removed from line 175

```
        cod_state->mem_deemph = 0;          move16();

    cod_state->seed2 = 21845;               move16();

    Init_Filt_6k_7k(cod_state->mem_hf2);
    Init_Filt_7k(cod_state->mem_hf3);
    cod_state->gain_alpha = 32767;          move16();

    cod_state->vad_hist = 0;

    wb_vad_reset(cod_state->vadSt);
```

```
        dtx_enc_reset(cod_state->dtx_encSt, isf_init);
```

…

[End Correction]

File: coder.c

[Begin Correction]

Mode vector removed (line 59), coding mode initialised (line 56)

```
    Word16 coding_mode = 0, nb_bits, allow_dtx, mode_file, mode = 0, i;
  Word16 reset_flag;
    long frame;
    char Mode[2] = "0";
```

…

Mode control file read operation changed (line 244)

```
if (mode_file)
        {
            if (fread(Mode, sizeof(char), 1, f_mode) != 1)
            if (fscanf(f_mode, "%hd", &mode) == EOF)
            {
                mode = coding_mode;
                fprintf(stderr, "\nend of mode control file reached\n");
                fprintf(stderr, "From now on using mode: %d kbit/s.\n", nb_of_bits[mode]);
                fprintf(stderr, " From now on using mode: %hd\n", mode);
                mode_file = 0;
                            }

            mode = (Word16) atoi(Mode);
            if ((mode < 0) || (mode > 8))
            {
               fprintf(stderr, " error in bit rate mode %hd: use 0 to 8\n", mode);
               exit(0);
            }

        }
        coding_mode = mode;

        frame++;
        fprintf(stderr, " Frames processed: %ld\r", frame);
        fprintf(stderr, " Frames processed: %hd\r", frame);
```

…

Code lines added to release the resources that were earlier left unreleased (line 282)

```
/* free allocated memory */
    Close_coder(st);
    Close_write_serial(tx_state);
    fclose(f_speech);
    fclose(f_serial);
    if (f_mode != NULL)
    {
                fclose(f_mode);
    }
```

[End Correction]

File: decoder.c, Code lines added to close the files earlier left open (line 267)

[Begin Correction]

```
Close_decoder(st);
    fclose(f_serial);
    fclose(f_synth);
```

[End Correction]

File: agc2.c, Math library header include removed (line 9)

[Begin Correction]

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

[End Correction]

File: dtx.c, Math library header include removed (line 9)

[Begin Correction]

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

[End Correction]

File: laqconc.c, Math and float library header include removed (lines 9, 10)

[Begin Correction]

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <float.h>
```

[End Correction]

File: c4t64fx.c, Ambiguous pointer expression corrected (line 490)

[Begin Correction]

```
for (j = (Word16) ((k + 1) % NB_TRACK); j < L_SUBFR; j += STEP)
                {
                    *p0++ = mult(*p0, psign[j]);    move16();
                   p0=++;
                }
```

[End Correction]

File: c2t64fx.c, Ambiguous pointer expression corrected (line 152-)

[Begin Correction]

```
for (i = 0; i < NB_POS; i++)
    {
        *p0++ = shr(*p0, 1);                move16();
       p0++;
        *p1++ = shr(*p1, 1);                move16();
       p1++;
    }
```

[End Correction]

6.1.14 Other changes to G.722.2 Annex C C-code

In addition to the above reported, mainly algorithmic changes, the ITU output format for v. 5.5.0 has been implemented as a command line option instead of being under a compilation flag. The ITU output format now also supports the DTX mode.