INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.722.1

Corrigendum 1
(11/2000)

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

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

Coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss

Corrigendum 1

ITU-T Recommendation G.722.1 – Corrigendum 1

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TESTING EQUIPMENTS	G.500–G.599
TRANSMISSION MEDIA CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
General	G.700–G.709
Coding of analogue signals by pulse code modulation	G.710–G.719
Coding of analogue signals by methods other than PCM	**G.720–G.729**
Principal characteristics of primary multiplex equipment	G.730–G.739
Principal characteristics of second order multiplex equipment	G.740–G.749
Principal characteristics of higher order multiplex equipment	G.750–G.759
Principal characteristics of transcoder and digital multiplication equipment	G.760–G.769
Operations, administration and maintenance features of transmission equipment	G.770–G.779
Principal characteristics of multiplexing equipment for the synchronous digital hierarchy	G.780–G.789
Other terminal equipment	G.790–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
QUALITY OF SERVICE AND PERFORMANCE	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DIGITAL TERMINAL EQUIPMENTS	G.7000–G.7999
DIGITAL NETWORKS	G.8000–G.8999

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation G.722.1

Coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss

Corrigendum 1

Summary

This corrigendum corrects three changes necessary to the existing C code (Release 1.1) that is supplied with ITU-T Rec. G.722.1. In each case, it corrects an error that was introduced when the original C code (known as release *code3.003* at the time of Determination) was converted to use basic operators. The corrected code will be labelled as *Release 1.2*.

The complete release 1.2 code is available in a zipped form with ITU-T Rec. G.722.1 (09/99) on ITU-T website.

Source

Corrigendum 1 to ITU-T Recommendation G.722.1 was prepared by ITU-T Study Group 16 (2001-2004) and approved under the WTSA Resolution 1 procedure on 17 November 2000.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

ITU-T Recommendation G.722.1

Coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss

Corrigendum 1

Introduction

This corrigendum corrects three changes necessary to the existing C code (Release 1.1) that is supplied with ITU-T Rec. G.722.1. In each case, it corrects an error that was introduced when the original C code (know in as release *code3.003* at the time of Determination) was converted to use basic operators. The corrected code will be labelled as *Release 1.2*.

The complete release 1.2 code is available in a zipped form with ITU-T Rec. G.722.1 (09/99) on the ITU-T website.

Correction #1

The code in question concerned the decoder error checking to test if the regions had been properly received and decoded. The code in question is only used when the decoder is unaware that it is processing an errored frame, i.e. that the receiving system has not informed the G.722.1 decoder the frame is errored.

The erroneous code is located in file *decoder.c*, in the function *test_4_frame_errors()*. The relevant code from *Release 1.1* is shown in Figure 1, and the proposed corrected code is also shown in Figure 1.

```
     /*  the next two lines of comments were modified in release 1.2
        *  to correct the description of the range of
        *  absolute_region_power_index[] to be tested in the next
        *  9 lines of code.
        */
       /*  if ((absolute_region_power_index[region] > 33 31) ||
             (absolute_region_power_index[region] < -6 -8) */

  acca = L_add(absolute_region_power_index[region],
                     ESF_ADJUSTMENT_TO_RMS_INDEX);
  accb = L_sub(acca,31);
  acca = L_add(acca,8);
  test();

   /* the next line was modifed in release 1.2 to
        * correct miss typed code and error checking.
        */

       if ((accb > 0) || (acca < 0))
    if ((accb > 0) || (accb > 0))
    {
       frame_error_flag |= 4;
       logic16();
    }
```

Figure 1. Code extracted from Release 1.1. The incorrect comments and code which are to be deleted from Release 1.1 are shown with a strike-through, and the new text is shown in bold.

Correction #2

The function *compute_region_powers()* in file encoder.c contains an error when testing the range of the regions powers. Figure 2 shows the relevant section of code and the necessary correction.

```
    /* The MLT is currently scaled too low by the factor
       ENCODER_SCALE_FACTOR(=18318)/32768 * (1./sqrt(160).
       This is the ninth power of 1 over the square root of 2.
       So later we will add ESF_ADJUSTMENT_TO_RMS_INDEX (now 9)
       to drp_code_bits[0]. */

    /* drp_code_bits[0] can range from 1 to 31. 0 will be used only as
       an escape sequence. */
    temp1 = sub(1,ESF_ADJUSTMENT_TO_RMS_INDEX);
    temp2 = sub(absolute_region_power_index[0],temp1);
    test();
    if (temp2 < 0)
    {
        absolute_region_power_index[0] = temp1;
        move16();
    }

    temp1 = sub(31,ESF_ADJUSTMENT_TO_RMS_INDEX);

     /** next line was corrected in Release 1.2 *******/
    temp2 = sub(absolute_region_power_index[0], 31  temp1);
    test();
    if (temp2 > 0)
    {
        absolute_region_power_index[0] = temp1;
        move16();
    }
```

Figure 2. Code extracted from Release 1.1. The incorrect comments and code which are to be deleted from Release 1.1 are shown with a strike-through, and the new text is shown in bold.

Correction #3

In file *encoder.c,* in function *vector_huffman(),* the variable *inv_of_step_size_times_std_dev* is declared as a *word16*. In the function *vector_huffman()* it is set to the product of two 16-bit variables, therefore is susceptible to overflow. In the original release of code it was a 32-bit quantity. In order to correct this using the basic operators, it is necessary to adjust several lines of code to accommodate this change. Figure 3 shows the changes to function *vector_huffman();* it is also necessary change the contents of the table variable *int_dead_zone[]* in the file *tables.c*, add a new table, *int_dead_zone_low_bits[]* – and declare it in file *tables.h* as *extern Word16 int_dead_zone_low_bits[NUM_CATEGORIES]*. Figure 4 shows the new table and the modifications to *int_dead_zone[]*.

```
    Word16 mytemp;
    Word16 myacca;

    /* initialize variables */
    vec_dim = vector_dimension[category];
    move16();

    num_vecs = number_of_vectors[category];
    move16();

    kmax = max_bin[category];
    move16();

    kmax_plus_one = add(kmax,1);
    move16();

    current_word = 0L;
    move16();

    current_word_bits_free = 32;
    move16();

    number_of_region_bits = 0;
    move16();

    /* set up table pointers */
    bitcount_table_ptr = (Word16 *)table_of_bitcount_tables[category];
    code_table_ptr = (UWord16 *) table_of_code_tables[category];

    /* compute inverse of step size * standard deviation */
    acca = L_mult(step_size_inverse_table[category],
                  standard_deviation_inverse_table[power_index]);
    acca = L_shr(acca,1);
    acca = L_add(acca,4096);
    acca = L_shr(acca,13);
    mytemp = acca & 0x3;
    acca = L_shr(acca,2);
    inv_of_step_size_times_std_dev = extract_l(acca);

    for (n=0; n<num_vecs; n++)
    {
        index = 0;
        move16();

        signs_index = 0;
        move16();

        number_of_non_zero = 0;
        move16();

        for (j=0; j<vec_dim; j++)
        {
            k = abs_s(*raw_mlt_ptr);

            acca = L_mult(k,inv_of_step_size_times_std_dev);
            acca = L_shr(acca,1);
myacca = L_mult(k,mytemp);
 myacca = L_shr(myacca,1);
 myacca = L_add(myacca,int_dead_zone_low_bits[category]);
 myacca = L_shr(myacca,2);
 acca = L_add(acca,int_dead_zone[category]);
 acca = L_add(acca,myacca);
 acca = L_shr(acca, 15 13);
             k = extract_l(acca);
```

Figure 3. Code extracted from function *vector_huffman()* of Release 1.1. The incorrect comments and code which are to be deleted from Release 1.1 are shown with a strike-through, and the new text is shown in bold.

```
Word16 int_dead_zone_low_bits[NUM_CATEGORIES]=
{
       2, 1, 0, 0, 3, 2, 0, 0
};

Word16 int_dead_zone[NUM_CATEGORIES]=
{
       9830,10813,11796,12780,13763,14746,16384,16384
       2457, 2703, 2949, 3195, 3440, 3686, 4096, 4096
};
```

Figure 4. New table, *int_dead_zone_low_bits [],* to be entered into the file *tables.c*, and the new values for *int_dead_zone[]*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2002