AES18-1996

AES recommended practice for digital audio engineering — Format for the user data channel of the AES digital audio interface

Published by
Audio Engineering Society, Inc.

Abstract

This document describes a method of formatting the user data channels provided within the digital audio serial interface format (AES3). The transmission format is an adaptation of the packet-based high-level data link control (HDLC) communications protocol and provides for the transmission of ancillary data that may or may not be time related to the audio signal. The data rate is constant within a range of ± 12.5 percent of a sampling frequency of 48 kHz. The standard also provides a data priority and management strategy to ensure that adequate capacity is available for downstream data insertion.

Contents

Foreword

[This foreword is not a part of *AES Recommended practice for digital audio engineering — Format for the user data channel of the AES digital audio interface*, AES18-1996.]

This document is a revision of AES18-1992. It is a companion document to the AES digital audio interface specification AES3, *AES Recommended practice for digital audio engineering — Serial transmission format for two-channel linearly represented digital audio data* and its revisions. It is the result of a desire by users of the interface to have a recommended format for the user data channel provided by the user bit.

In this edition, an amendment has been added to provide recommendations for coding to be used in complying with AES18 and to complete the subclauses therein. The coding is presented in an added clause 7 and an added annex B. An added annex C provides informative references for the coding. The coding has been developed in cooperation with the European Broadcasting Union.

The requirement was for a system that is flexible and independent of the user application, and that can carry message data related in time to the audio data, as well as information, such as text, that may be unrelated to the audio data. A further requirement was for a format that treats the user data channel as a transparent carrier at sampling frequencies in the range of 48 kHz ± 12.5 percent such that a constant data rate is achieved. The system specified is based on a widely used packet communication protocol, high-level data link control (HDLC) (ISO Publication 3309), which is standardized in the information technology industry. The HDLC protocol has been adapted for unidirectional transmission and to permit the accurate transmission of time-dependent information, but the HDLC integrated circuits, which are readily available from several manufacturers, are still able to be used. It is therefore expected that this transmission format will be easy to implement without recourse to special hardware and will, as a result, be included as a matter of routine in commercial interface equipment.

This document results from the work of the Working Group SC-02-04 Signal Labeling and Ancillary Data, established by SC-02 Subcommittee on Digital Audio of the AES Standards Committee, in close collaboration with the European Broadcasting Union (EBU).

Experimental and prototype equipment was designed and built by A. Komly and A. Viallevieille to prove and test the system during development of the specification. Input documents were provided by R. Lagadec, G. McNally, J. Wilkinson, A. Komly, A. Weisser, A. Viallevieille, and J. P. Nunn.

The following additional individuals participated in the preparation of the document: B. Blüthgen, R. Caine, R. Finger, W. Foster, N. Gilchrist, A. Griffiths, R. Hankinson, T. Holman, Y. Ishida, A. Komly, R. Lagadec, P. Lidbetter, S. Lyman, G. McNally, L. Moller, D. Queen, C. Sanchez, W. T. Shelton, A. Viallevieille, A. Weisser, and J. Wilkinson. In addition, R. Chalmers participated in the preparation of the revision.

J. P. NUNN, *Chairman*
SC-02-04 Working Group Signal Labeling and Ancillary Data
1996-07

1998-07-18 printing

AES recommended practice for digital audio engineering — Format for the user data channel of the AES digital audio interface

1 Scope

This document specifies a recommended method of formatting the user bit of each channel of the AES digital audio interface (AES3, *AES Recommended practice for digital audio engineering — Serial transmission format for two-channel linearly represented digital audio data*). Each user data channel so formed is independent of the user application and is primarily intended for the transmission of data associated with the audio signal, although data unrelated to the audio signal may also be transmitted. There is no restriction on the length of messages that may be transmitted in the user data channel. The format treats the user data channel as a transparent carrier at sampling frequencies in the range of 42 kHz to 54 kHz, that is, 48 kHz ± 12.5 percent. There is no theoretical upper limit, but if sample rate conversion to a sampling frequency below this range is performed, data management is to be used to avoid loss of vital data.

The document describes the method of formatting user information into packets, together with the rules for data insertion into the multiplex and for data management. The purpose and the content of the user information for particular applications are outside the scope of this document.

This document describes a practice for professional applications and is not intended to encompass applications related to consumer versions of the digital audio interface.

2 Normative references

The following standards contain provisions that, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this document are encouraged to investigate the possibility of applying the most recent editions of the indicated standards.

1) AES3-1992, *AES Recommended practice for digital audio engineering — Serial transmission format for two-channel linearly represented digital audio data.* New York: Audio Engineering Society, 1992.

2) ISO 3309, *Information processing systems–Data communications–High-level data link control procedures– Frame structure.* Geneva, Switzerland: International Organization for Standardization, 1984.

3 Definitions

3.1 transport system: Method by which messages are carried between source and destination of the application.

3.2 address: Identification of either the destination hardware or the application. It can be combined with an address extension.

3.3 address extension: Part of the address that extends the range of destination hardware or applications that may be addressed.

3.4 block: Repetitive structure chosen for the transport system by the user.

3.5 continuity index: Count of the messages or packets with a given address. In this system the count is modulo 8, allowing up to seven missing messages or packets to be detected. This system uses a message continuity index and a packet continuity index.

3.6 control byte: Information enabling receiving equipment to interpret and decode, successfully and reliably, the data that follow. The control byte has four fields for a priority index, a packet continuity index, address extension enable, and link bits.

3.7 cyclic redundancy check code (CRCC): See 3.8.

3.8 frame check sequence (FCS): Two bytes formed mathematically from the packet information that are added to the packet to provide forward error detection.

3.9 high-level data link control (HDLC): Internationally standardized protocol for the transmission of messages based on packets.

3.10 frame level: Level of the transport system where the packets are formed into HDLC frames.

3.11 HDLC frame: HDLC packet after it has been channel coded, with the addition of the frame check sequence and the limit flags (ISO Publication 3309 [3]).

3.12 HDLC frame flag: Unique code at the physical level, which identifies the start and end of an HDLC frame.

3.13 justification bits: Bits left unused in the user data channel to allow the system to operate between audio sampling frequencies 42 kHz and 54 kHz without losing information. The number of justification bits varies dynamically with the audio sampling frequency.

3.14 A0: Least significant bit (LSB) of a byte. It is transmitted first on this serial transmission system.

3.15 message level: Level of the transport system where the messages are received from the application and processed so that they can be formed into packets. At the receiver packets are processed, reformed into messages, and passed to the application.

3.16 packet: Sequence of bytes comprising address and optional address extension bytes, control byte, and message segment.

3.17 A7: Most significant bit (MSB) of a byte. It is transmitted last on this serial system.

3.18 packet level: Level of the transport system wherein the self-contained packets that comprise the message are formed.

3.19 physical level: Level of the transport system wherein the frames are channel coded and inserted into the user data channel of the digital audio interface.

3.20 priority index: Number chosen by the user by which a priority is assigned to a packet or message. Four priority levels have been provided.

3.21 repetition index: Number chosen by the user which controls the number of times each packet of a message is repeated.

3.22 segment: Part of the message after the message has been split up. The segment will form the information field of the packet.

4 General description

4.1 Messages

The digital audio interface specified in AES3 is primarily intended to allow the transmission of audio signals between digital audio equipment. The present document specifies a system by which the user data channel of the interface can be used to transport a wide variety of messages along with the audio data. The messages can come from many different applications, scripts, subtitles, editing information, copyright, performer credits, downstream switching instructions, and so on. The application can be freely chosen and defined by the user, and the messages that are sent can be of many different types.

NOTE — The interface specified in AES3 is also specified in European Broadcasting Union (EBU) Tech 3250, IEC 958, and ITU-C 647. See annex C.

4.2 Transport system

The transport system described is based on a widely used protocol, high-level data link control (HDLC), complying with ISO 3309, which is a standard in the information technology industry. In general the HDLC system is capable of handling messages in two directions, but in this system it is only used in one, the same direction as the audio signals. Integrated circuits, available from several manufacturers, support the HDLC protocol. These often take the form of the HDLC interface combined with a microprocessor and a local interface to the message receiving or sending system.

4.3 Packets

Digital audio signals are likely to be repeatedly processed and passed between different equipment. The messages that the user would like to accompany the audio signals are likely to grow at successive stages in the program chain. This transport system allows downstream equipment to remove or add messages in the channel, provided there is enough capacity. Each message is sent as one or more packets, depending on its length. Each packet carries the address of its destination. This allows a receiver to read only those messages addressed to it. A large number of addresses is available, and the system puts no restriction on the user as to how or for what purpose the messages are used.

4.4 Features

4.4.1 Messages

The transport system can be used to carry a wide variety of information. Messages can be of any length; they can be time critical or not.

4.4.2 Multiplexing

The transport system can carry simultaneous messages from different applications, subject to the maximum bit rate provided by the interface. The system allows the user to insert messages at any point of the chain. The messages can have different priority levels, which affect the speed and frequency with which the parts of a message can be inserted.

4.4.3 Bit rate

The transport system is based on packets and can be used with any audio sampling frequency. For audio sampling frequencies within the range of 48 kHz ±12.5 percent, which includes 44.1 kHz, it provides a constant data rate. This is achieved by inserting justification bits at sampling frequencies above 42 kHz.

4.4.4 Synchronization

The transport system allows the user data channel to become a self-contained transmission channel that is independent of the audio signal block structure. However, the channel can easily be synchronized to an external signal such as a time code clock or a video frame rate if this is required by the user.

4.4.5 Error detection

The transport system includes error checking to allow detection of any corruption of the data in the messages. The user can safeguard important messages by instructing the transport system to repeat every packet automatically, or to repeat the message itself.

4.4.6 Efficiency

The system has an efficiency that is the ratio of the application data bit rate to the total bit rate of the user data channel. The system needs a certain amount of overhead bit rate to provide addressing, packet identification, error detection, justification, and so on. The percentage amount of this overhead varies predominantly with the length of the message and, to a lesser extent, with the block length. The efficiency can be as high as 60 percent with a block length of 40 ms and a sampling frequency of 48 kHz. A 70 percent efficiency is achievable with the same block length and a sampling frequency of 44.1 kHz because fewer justification bits are required.

5 Data formatting

5.1 Application level

The messages are generated by the application and are then passed to the transport system together with the other parameters that the transport system uses to process the messages. The parameters are:

a) destination address;
b) priority index;
c) repetition index.

5.1.1 Destination address

The application shall provide a destination address. It is either 1 byte long, or 2 bytes if an address extension is used. The use of the address and address extension bytes is described in clause 7

5.1.2 Priority index

The priority index shall be chosen by the user according to the urgency of the message. It is one of four possible levels and determines either the maximum delay before the message is sent, or how quickly the message is sent if it is a long message. The priority system is described in more detail in 5.2.2.

5.1.3 Repetition index

The repetition index shall control how many times each packet of the message shall be repeated by the transport system. The repetition index can have any value, but to minimize the system loading, a repetition index between 0 and 5 is recommended. The whole message can also be repeated by the application.

5.1.4 Message

The message content shall be treated as a binary signal, and it can be of any number of 8-bit bytes.

5.2 Transport system

The transport system shall receive the message data from the application level and processes them so that they can be transmitted in the user data channel. The data formatting shall be carried out on four levels (figure 1):

a) message level;
b) packet level;
c) frame level;
d) physical level.

Each level has its own function, and all the levels together ensure the correct formatting of the data to allow them to be transported.

5.2.1 Message level

The message level (figure 1, step 1) shall receive the message from the application together with the other parameters listed in 5.1. It shall add a header to the message and then divide the message, including the header, into segments, which will be transported separately. Finally the message level shall pass on the segments, together with the parameters, to the packet level.

5.2.1.1 Message header

The message header (figure 2) shall contain information on the message length and a message continuity index. The header is either 1 or 2 bytes long, depending on the length of the message. A message length format bit in the first byte signals whether the second byte, the message length extension, is used.

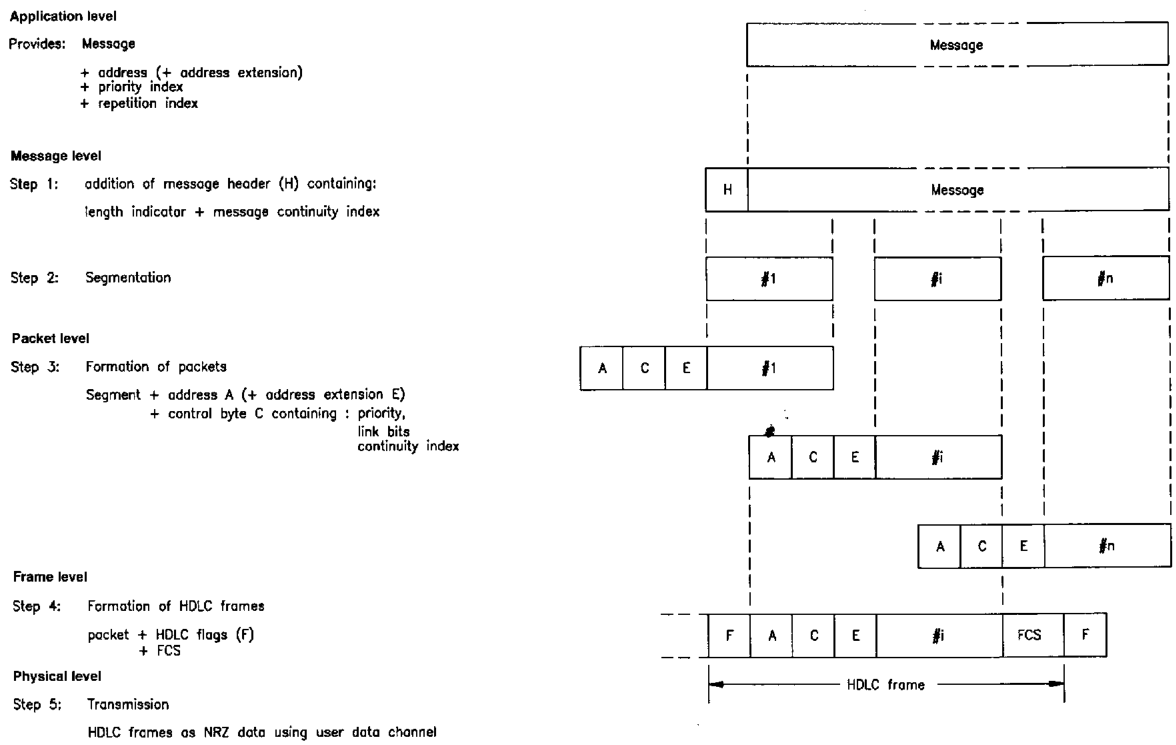


Figure 1 — Main steps in data formatting

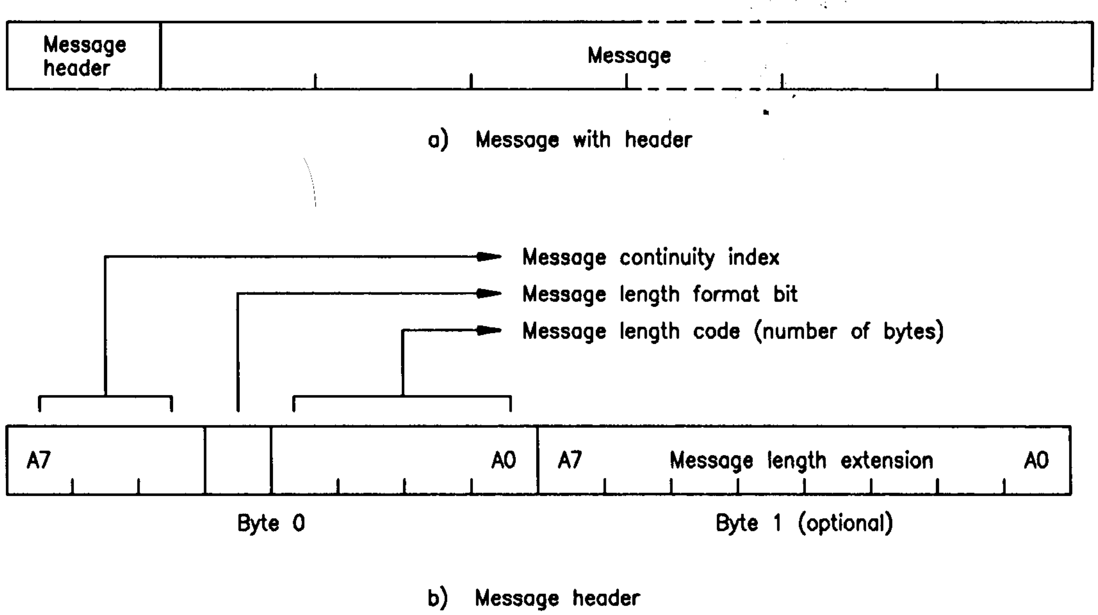


Figure 2 — Message level

5.2.1.1.1 Message header: byte 0 (A7, MSB to A0, LSB)

a) Bits A7, A6, A5: message continuity index

The 3-bit continuity index is a modulo-8 count of the messages sent by an application. A7 is the MSB. The receiver can use this index to detect any loss of messages. Each application shall have its own continuity index. However, when a message is repeated under the control of the repetition index, the message continuity index does not change (see 5.2).

b) Bit A4: message length format

0: The header is 1 byte long
1: The message length extension byte is used and the header is 2 bytes long.

c) Bits A3, A2, A1, A0: message length code, most significant bits

These bits indicate the length of the message. The precise use is described in 5.2.1.2.

5.2.1.1.2 Message header: byte 1 (A7, MSB to A0, LSB) (optional)

Bits A7 to A0: message length extension code

These bits allow the length code of the message to be extended as described in 5.2.1.2.

5.2.1.2 Message length coding

The messages can be any length. The different message lengths shall be coded in one of three ways, according to length. (Note that the message length does *not* include the message header byte or bytes.)

5.2.1.2.1 If the length of the message is less than or equal to 15 bytes, the message header shall be 1 byte long because the message length can be given by the 4-bit message length code (header byte 0, bits A3 to A0). The message length format bit (header byte 0, bit A4) is set to logic 0.

5.2.1.2.2 If the length of the message is greater than 15 bytes but less than or equal to 4094 bytes, the address extension byte shall be used and the message length shall be coded by 12 bits (header byte 0, bits A3 to A0 plus the extension byte, header byte 1, bits A7 to A0). The message length format bit of the byte 0 shall be set to logic 1.

5.2.1.2.3 If the message length is greater than 4094 bytes or unknown, the message length shall be coded as 4095 (FFF hex).

5.2.1.3 Segmentation

The message, complete with its header, shall be divided into segments of 16 bytes or less if it is the last or only segment (figure 1, step 2). The header shall always be part of the first segment.

5.2.2 Packet level

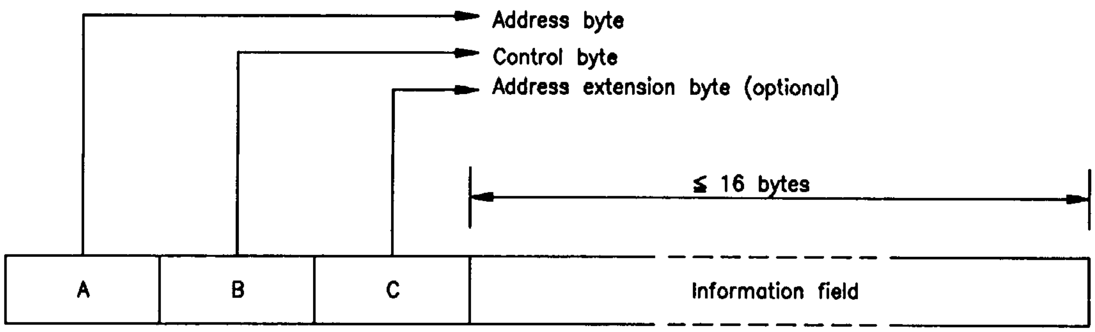
The packet level shall assemble the segments into packets (figure 1, step 3). From the message level it shall receive:

a) the segments of the message;
b) the parameters that were passed on from the application:

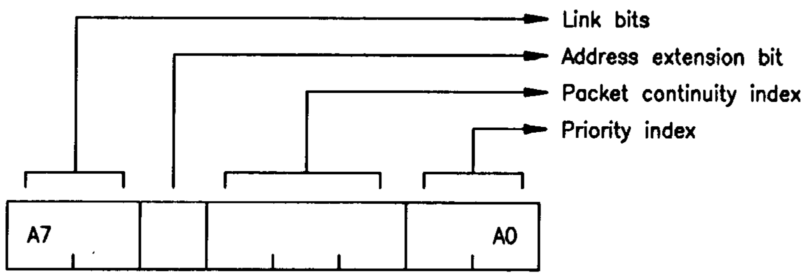
1) the address (and the address extension if there is one);
2) the priority index;
3) the repetition index.

Each packet (figure 3a) shall be formed from one segment of the message with the addition of an address byte, a control byte, and an address extension byte if present. The segment shall become the information field of the packet.

The segment, the address byte, and the address extension byte shall be inserted into the packet exactly as they are received from the message level.



a) Packet



b) Control byte

Figure 3 — Packet level

5.2.2.1 Control byte

The control byte shall be generated at the packet level and shall be 8 bits long (A7, MSB to A0, LSB), divided into 4 fields [figure 3(b)].

a) Bits A7, A6: link bits

1 0: The packet is the first or only packet of a message.
(This packet always contains the message header.)
0 1: The packet is the last packet of a message that is two or more packets long.
0 0: The packet is an intermediate packet of a message that is three or more packets long.
1 1: Reserved for system packet identification (see 5.2.1).

b) Bit A5: address extension bit

0: There is no address extension byte.
1: There is an address extension byte.

c) Bits A4, A3, A2: packet continuity index

The packet continuity index, a modulo-8 count of all the packets, shall be sent with the same address defined by an application. A4 shall be the MSB. This allows the receiver to detect if a packet is missing from a message.

When a packet is successively repeated, under the control of the repetition index, the packet continuity index shall not change.

d) Bits A1, A0: priority index

0 0: Priority index 0 (lowest);
0 1: Priority index 1;
1 0: Priority index 2;
1 1: Priority index 3 (highest).

The priority index shall be set by an application (see 5.1.2).

5.2.2.2 Packet level repetition

After the packets are formed, they shall be repeated as many times as defined by the repetition index and passed on to the frame level.

5.2.3 Frame level

The frame level shall receive the packets and construct the frames (figure 1, step 4), which are inserted into the user data channel. The frames use the high-level data link control (HDLC) structure. A frame (figure 4) comprises:

a) beginning flag: (01111110) (7E hex);
b) packet;
c) frame check sequence (a cyclic redundancy check);
d) ending flag: (01111110) (7E hex).

When a series of packets is transmitted, the end flag of one frame may be the start flag of the next frame. The HDLC frame is defined in ISO 3309.

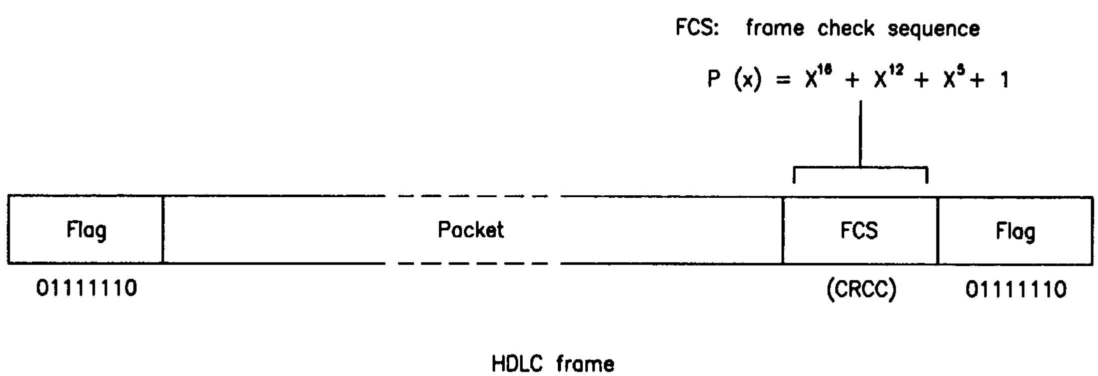


Figure 4 — Frame level

5.2.4 Physical level

The physical level (figure 1, step 5) shall be the serial transmission of the encoded frames at a bit rate equal to the audio sampling frequency.

5.2.4.1 Data coding

The data in the packet part of each frame, including the FCS, shall be coded to insert one extra logic 0 after every sequence of five logic 1's (figure 5). Sequences of six logic 1's are therefore available to flag unambiguously the beginning and the end of a new frame, because this pattern cannot occur in data. The flag is the byte (01111110) (7E hex).

When there are at least seven consecutive logic 1's, the link shall be in the idle mode. This procedure is fully described in ISO 3309.

5.2.4.2 Data transmission

The frames shall be transmitted in the user data channel (U) of the digital audio interface (AES3) as non-return-to-zero (NRZ) signals.

The use of this transport system shall be signaled in the channel status channel (C) of the interface (channel status byte 1, bits 4 to 7).

The least significant bit of each byte shall be transmitted first.

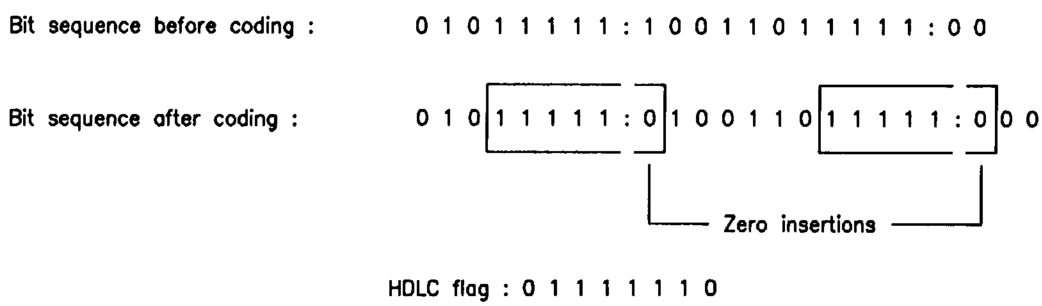


Figure 5 — HDLC channel coding

6 Channel management

The system described so far is able to transmit messages, but no special provision is made for inserting messages in downstream equipment or for resolving any conflicts of priority. The channel management system is provided to control these functions. It defines

a) channel formatting and block structure;
b) channel description system;
c) rules for inserting the data packets.

6.1 Channel formatting

The data flow in the user data channel shall be divided into repetitive structures called blocks. The block structure is shown schematically in figure 6.

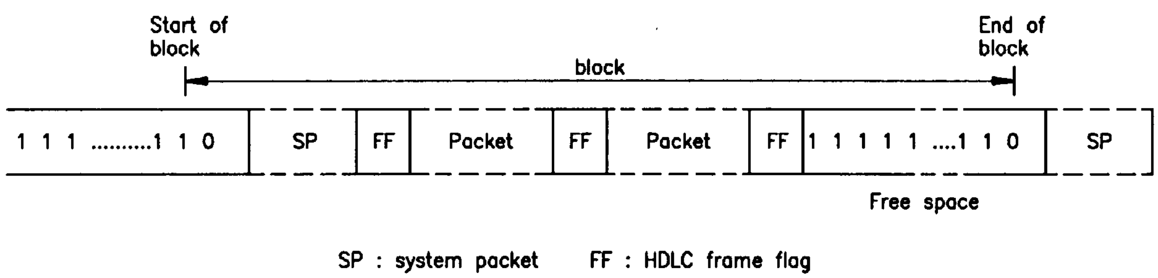


Figure 6 — Block structure

6.1.1 Block rate and length

The block structure can be synchronized to an external event and the user can set the repetition rate of the blocks to suit the application. Recommended repetition rates are shown in table 1. Any other repetition rate can, however, be defined by the user (see 6.2.1.3). These block rates are independent of the audio sampling frequency and will contain different numbers of bits at different audio sampling frequencies.

Normally the number of bits in a block will be constant. However, in some cases it will be impossible to provide a constant number of bits in the blocks. For example, using the National Television Systems Committee (NTSC) frame rate with an audio sampling frequency of 48 kHz, there are 8008 audio samples every 5 video frames. Because the number 8008 is not a multiple of 5, it would be necessary to have variable block lengths.

Table 1 — Recommended repetition rates

Blocks per second	Duration ms	Remarks
2	500	
5	200	
24	41.67	Film frame rate
25	40	Phase alteration line-rate (PAL), sequential color with memory (SECAM), or 1250/50 high-density television (HDTV) frame rate
29.97	33.37	NTSC frame rate
30	33.33	1125/60 HDTV frame rate
33.33	30	Digital audio tape (R-DAT) frame rate
100	10	Shortest practical block

6.1.2 Block start

A block start shall be identified by a logic 0 which follows at least 7 logic 1's and is terminated by at least 7 logic 1's which are the logic 1's before the first logic 0 of the following block. This logic 0 is defined as the first bit of the block. The HIGH to LOW transition that follows the sequence of logic 1's is considered the beginning of the block (figure 6).

6.2 Channel description

The channel shall be divided into blocks. The start of each block is indicated by the block start sequence described in 6.1.2.

Following the block start sequence, an optional system packet may be added which describes the block length and controls the data insertion. The system packet can have an information field that can be used to carry system data.

6.2.1 System packet

The system packet shall contain an address byte, a control byte, an optional address extension, a descriptor byte, and an information field, which can contain up to 15 bytes [figure 7(a)].

The address FF (hex) shall always be used for the system packet and shall be restricted to the system packet to permit easy identification.

6.2.1.1 Control byte of system packets

The control byte of a system packet (A7 to A0) [figure 7(b)] shall have a structure different from the control bytes described in 5.2.2.1 for normal packets.

a) Bits A7, A6: system packet identification

1 1: This code is always used and identifies the packet as a system packet.

b) Bit A5: address extension bit

0: There is no address extension.
1: There is an address extension (see 6.2.1.2). Reserved and shall not be used.

c) Bit A4: reserved and shall not be used at present; set to logic 0.

d) Bits A3, A2, A1, A0: priority enable bits:

A3: priority enable for priority 3 (highest);
A2: priority enable for priority 2;
A1: priority enable for priority 1;
A0: priority enable for priority 0 (lowest).

When a priority enable bit is set to logic 1, messages that have the corresponding priority can be inserted; when the bit is set to logic 0, the insertion is forbidden. There is no restriction on the combination of enable bits that can be set. For general rules of insertion see 6.3.1.

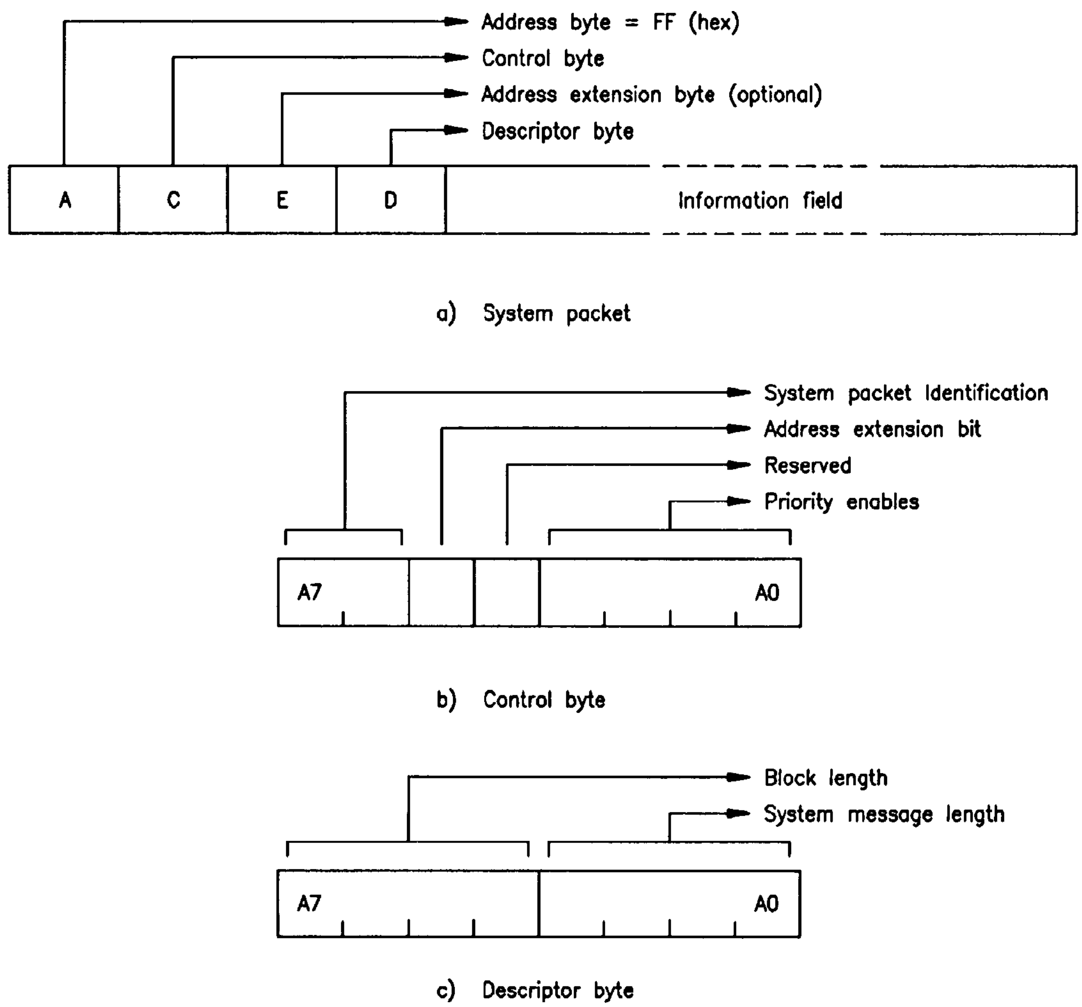


Figure 7 — System packet

6.2.1.2 Address extension byte

The address extension byte shall identify the contents of the information field in the system packet and shall only be transmitted when the address extension bit is set to logic 1.

Bits A7 to A0: description of information field of system packet. See clause 7.

6.2.1.3 Descriptor byte

The 8-bit descriptor byte [figure 7(c)] shall have two fields that define the length of the block and the length of the system message.

a) Bits A7, A6, A5, A4: coded length of block

0	0	0	0: 24 frames/s
0	0	0	1: 25 frames/s
0	0	1	0: 30 frames/s
0	0	1	1: 29.97 frames/s
0	1	0	0: 10 ms
0	1	0	1: 200 ms
0	1	1	0: 500 ms
0	1	1	1: 30 ms
1	0	0	0: User-defined length

All other codes are reserved at present and shall not be used.

b) Bits A3, A2, A1, A0: system message length (A3, MSB A0, LSB)

The number of bytes in the information field of the system packet shall be given by bits A3 to A0.

6.2.1.4 Information field in system packet

The information field that follows the descriptor byte can contain up to 15 bytes of data (figure 7). As shown in clause 7, table 7, byte 31, it can carry time codes, indicate the block length when user defined, or carry other information. The content of the information field, if present, shall be identified by the address extension byte of the system packet (see 6.2.1.2).

6.3 Data insertion (see annex A)

Packets shall be inserted into the blocks of the user channel by packet multiplexers. When packets are due to be inserted, a block may already contain some packets. Existing packets shall be followed by successive logic 1's which indicate an idle channel up to the end of the block. The ending flag of a packet and the beginning flag of the following packet can be the same, or can be separated by up to three bytes.

6.3.1 Rules of insertion

Multiplexing of new packets involves two steps. The first step is the detection of a block start (see 6.1.2). The second step is the detection of seven consecutive logic 1's which indicate that the channel is in the idle mode. This allows the multiplexer to insert a new packet if there is enough free space before the end of the block. If there is space for the new packet, the seventh existing logic 1 is set to logic 0, forming a new HDLC flag. The new packet (possibly including a further HDLC beginning flag) then follows (figure 8).

Table 2 — Examples of numbers of justification bits for various block lengths

Sampling Frequency	Free Space	Total bits per block			Justification bits per block		
kHz	%	10 ms	40 ms	200 ms	10 ms	40 ms	200 ms
42	0	420	1680	8400	0	0	0
44.1	5	441	1764	8820	21	84	420
48	12.5	480	1920	9600	60	240	1200
54	25	540	2160	10800	120	480	2400

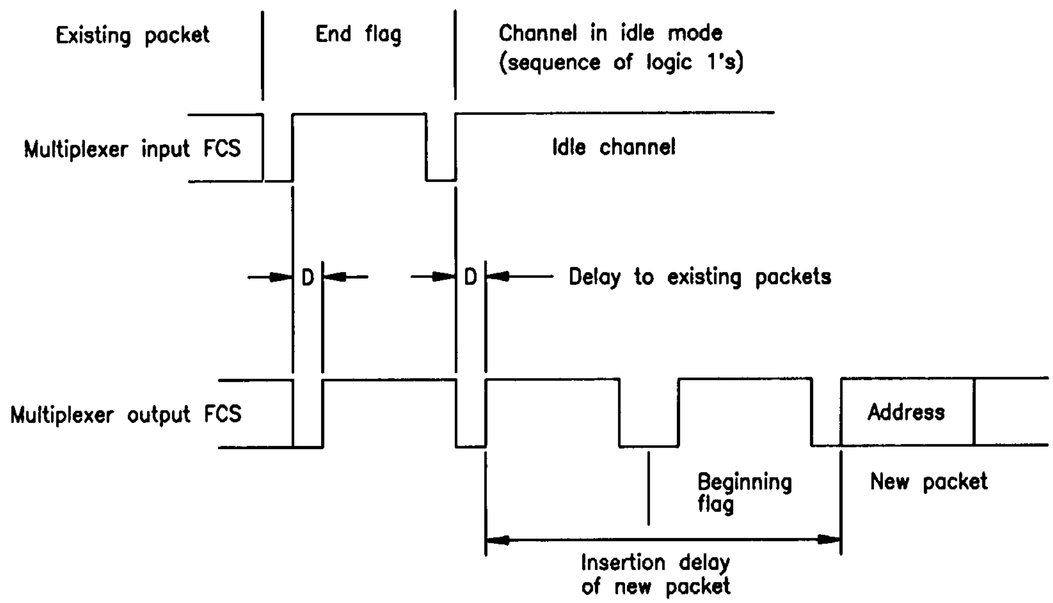


Figure 8 — Example of packet insertion

The amount of available free space depends on the block length and the length of previously inserted messages. However, free space (justification bits) shall be reserved at the end of each block to avoid losses of packets during sampling frequency conversion or variable-speed operation. The amount of free space that shall be reserved should allow the audio sampling frequency to be reduced to 12.5 percent below 48 kHz. Examples are given in table 2 for various audio sampling frequencies and various block lengths.

The use of justification shall be mandatory over this audio sampling frequency range. The interface and the transport system can be used at other audio sampling frequencies, such as 32 kHz, but will need appropriate data management.

6.3.2 Priority management

Priority management shall be carried out at the multiplexing level and, in the first instance, the insertion of messages shall be allowed or forbidden according to the state of the priority bits of the system packet (see 6.2.1.1). Only packets of the messages whose priority bits are set at logic 1 can be inserted.

The priority management rules further ensure that the multiplexers are able to average the loading of the blocks to prevent upstream multiplexers from monopolizing the channel capacity.

When the application is designed, the user shall assign a priority to each message. The choice of this priority may be either

a) to ensure a minimum delay in the insertion of short urgent messages carrying real-time data, or

b) to provide a sufficient bit rate, compatible with the application, if the messages are long.

6.3.2.1 Priority insertion rules

The priority management rules shall be according to table 3. The table defines the number of packets per message that can be inserted per block as a function of the block length and the priority.

When the number of packets that can be inserted into a block for a given priority is greater than or equal to 1, packets can be multiplexed according to the rules defined in table 3.

However, for long messages of lower priorities, where the rules will only allow one packet to be inserted into one of several blocks, extra rules are needed to prevent overloading of the earlier blocks. In these circumstances the following rules shall apply to spread the loading over all the available blocks.

If the insertion rules allow one packet to be introduced into every *n* blocks (*n* = 2, 4, 5, 10, 20, 40, defined in table 3), then:

1) The packet shall only be inserted into the first *n*/2 blocks if there is at least one of the blocks with more than half of its length still available for downstream multiplexing.

2) If the packet is not inserted in the first *n*/2 blocks, it shall be inserted into the earliest available block.

Table 3 — Maximum number of packets per message inserted per block as a function of block length and priority

	Block length			
Priority	10 ms	1 frame	200 ms	500 ms
3 (highest)	1	4	20	50
2	1/4	1	5	12
1	1/20	1/5	1	2
0 (lowest)	1/40	1/10	1/2	1

6.3.2.2 Real-time applications

For urgent or critical messages, which are usually one packet long, the priority index can be used to control the insertion delay of the message packets. The block length represents the maximum delay needed to multiplex an urgent packet into the network (a single packet message can be inserted into a single block). Table 4 shows how the maximum delay varies with the block lengths and priority. This is subject to the multiplex not being overloaded at the time the urgent or critical message originates. The block length shall be chosen to produce a delay that is appropriate to the application.

Table 4 — Maximum delay of packet insertion as a function of block length and priority

	Block length			
Priority	10 ms	1 frame	200 ms	500 ms
3	10 ms	1 frame	200 ms	500 ms
2	1 frame	1 frame	200 ms	500 ms
1	200 ms	200 ms	200 ms	500 ms
0	500 ms	500 ms	500 ms	500 ms

6.3.2.3 Bit-rate adaptation

The priority management rules and the choice of priority shown in table 5 can be used to determine the number of packets that can be inserted into one block, and hence the effective bit rate. For example, for a 10-ms block length and a message with priority 3 (the highest) one packet will be inserted into every block during the transmission. However, with the same block length but a message with priority 2, one packet shall be inserted into only one block out of every four blocks. The mean value is one packet per 40 ms.

Table 5 — Priority management rules

Priority	Approximate packets per second	Approximate bits per second
3 (highest)	100	12 800
2	25	3 200
1	5	640
0 (lowest)	2.4	320

7 Addressing of applications

7.1 Introduction

Previous sections of this document have defined the transport and packet structures and the rules for inserting packets into the user bit data stream. However, a receiver needs more information about an application if it is to interpret a message correctly. This clause describes a message addressing strategy that enables a receiver to identify applications and select messages.

Because the user channel is unidirectional and no return channel between the receiver and the source exists, messages remain in the channel until specific action is taken to remove them. By using a hard address and a soft address (address extension) the protocol provides a simple means of filtering out messages that are no longer required, without the need to decode the messages.

The choice of address can depend upon the application that is currently defined and is given in the tables in this clause. System designers may extend the number of applications and reserve choices of related addresses by application to the AES Standards Secretariat (see 7.6). Provision has also been made for addresses that are specific to a particular user environment.

The rules for choosing addresses are given in this clause.

7.2 Classes of application

Applications should be divided into four classes according to the general area in which they are to be used. These classes broadly relate to the audio signal path and should be defined as

a) local applications;
b) production applications;
c) distribution applications;
d) common applications.

This classification method enables downstream multiplexers to detect and delete messages when they are outside their area of use and no longer useful.

7.3 Applications

The address shall be chosen according to the application. The combination of the two available address bytes, the hard address and the soft address (address extension), shall be used to define

a) the purpose of the application;
b) the type of the terminal.

7.3.1 Purpose of application

An application can be generally designed to fulfill one of the requirements of audio processing. Messages may be provided for different purposes: program description, engineering notebook, editing files, switching information, and so on. The purpose of an application can determine the areas where the transmitted messages are to be used along the audio chain. Each receiver shall be able to decide whether or not to accept a message.

7.3.2 Type of terminal

Messages should be generated by terminals that provide different types of data 7-bit characters (7-bit ASCII) conforming to ISO 646, 8-bit characters (8-bit ASCII) conforming to ISO/IEC 4873, binary, and so on) to the transport system. The messages should be organized to be processed efficiently. Their format depends on the processor that is used to generate and to receive messages. An application may use more than one format if the data is available and used in different formats. Such sub-messages of an application shall be easily identified as belonging to the same application, but each of them shall also be easily identified by a downstream multiplexer so that it can be dispatched to the correct equipment according to the format.

For example, an application may need to describe an audio program for use in a D2-MAC service or radio data system (RDS) (see annex B) and may be divided into sub-applications, each with its own message formats:

a) an ASCII message that carries the title of the program;
b) a binary message that carries the "music-speech" information;
c) and so on.

When different parts of downstream equipment need to access these sub-messages, they shall be closely associated.

7.4 Message exchange

The principles in 7.3 suggest that a system should be available to help a user in the choice of message addresses. But some messages inserted into the user channel can be exchanged between different users, perhaps as tapes, along the audio chain. To make use of these messages, receivers need to know beforehand information on the application and format of messages. This information may be given to the receiver in two ways, either as a predefined format, available as a standard, or in a message multiplexed with the other messages in the user channel.

7.4.1 Predefined messages

Common messages that can be received anywhere on a network shall be given special addresses. In this case, the addresses identify the content of messages that should not be deleted. To simplify the design of these applications, the formats of common messages should be pre-defined and standardized. For example, program reference data should have a predefined address and the items such as program title, program number, copyright, and so on should have predefined sub-addresses. The addressing shall identify the format of the data so that any receiver can be programmed to use the message information.

7.4.2 Messages

To exchange messages having downstream users, the main features of each application shall be specified, including the address, the purpose, the type of terminal, and the related sub-applications.

The transmission of these specifications itself forms an application of the user data channel, referred to as a description application. The addresses and format of these description applications shall be pre-defined. These addresses are chosen according to the general rules. Description messages should be repeated to allow a fast access to the content of the user channel.

However, it is necessary to specify very carefully the predefined addresses to permit downstream users to choose their own addresses compatible with existing addresses to avoid confusion of messages.

7.5 Addressing

The HDLC system specification provides an address byte (hard address) and an address extension byte (soft address). These should be used as defined in 7.5.1 and 7.5.2 to provide the various layers of definition of the applications to satisfy the requirements given in 7.4.

7.5.1 Address byte

The two MSB of the hard address byte, bits 7 and 6, should be used to give the scope of the applications as outlined in 7.2. This is defined in table 6.

The rest of this byte, bits 5–0, and the extension address byte should be used to

a) identify the purpose of an application;
b) give the type of the terminal;
c) link sub-applications related to the same application.

To keep the system of addressing as simple as possible, it is convenient to use the ability of the receiver circuits to decode automatically a chosen hard address. Each sub-message of an application shall therefore have the same hard address to identify the purpose of the application. Each of them can then be dispatched into the correct equipment according to its extension address byte.

Codes of the six remaining bits of the hard address should be allocated to common classes of application. Details of description application addresses are given in table 6.1. Details of other common message applications are given in tables 6.2 and 6.3.

7.5.2 Extension address byte

The extension address byte indicates the exact content and format of each sub-message and the type of terminal that is used. A demultiplexer can feed several terminals linked to the same application according to the extension address.

The use of the extension address byte also permits

a) messages meant for the same destination to be inserted in different points of the chain, provided the extension addresses are different;

b) messages related to the same application to be broken down so as to optimize the parameters of the message system (use of the priority index, repetition index, etc.) and to permit future extension.

The extension addresses allocated to the various applications defined by the bytes allocated in table 6.2 are given in table 7.

7.6 Assignment of reserved bits

Clause 7.5 contains the coding in use as of 1995-09-25. Application may be made by any directly and materially affected party for assignment of reserved bits by mail to the publisher, Standards Secretariat, Audio Engineering Society, 60 East 42nd St., New York, NY 10165. Application for codes may also be made via data modem calls to the AESSC document exchange. Access phone numbers and log-in instructions are available from the AES office in your region. All applications shall be reviewed by the AESSC and the EBU according to their respective rules and procedures.

Table 6 — Message address byte

Message address byte bits 7–6	
7 6	**Application of message**
0 0	Local
0 1	Production
1 0	Distribution
1 1	Common

Table 6.1 — Message address byte

Message address byte bit 5
Reserved and should be set to logic 0 except with system packets

Table 6.2 — Message address byte

Message address byte bits 4–0			
4 3 2 1 0	**Hex**	**Dec**	**Type of information**
0 0 0 0 0	00	0	Default. Format not specified.
0 0 0 0 1	01	1	Audio data (low grade)
0 0 0 1 0	02	2	Time code
0 0 0 1 1	03	3	Reserved
0 0 1 0 0	04	4	|
0 0 1 0 1	05	5	|
0 0 1 1 0	06	6	↓
0 0 1 1 1	07	7	Reserved
0 1 0 0 0	08	8	User defined
0 1 0 0 1	09	9	|
0 1 0 1 0	0A	10	|
0 1 0 1 1	0B	11	|
0 1 1 0 0	0C	12	|
0 1 1 0 1	0D	13	|
0 1 1 1 0	0E	14	↓
0 1 1 1 1	0F	15	User defined
1 0 0 0 0	10	16	Data channel (transparent data)
1 0 0 0 1	11	17	Switching and routing
1 0 0 1 0	12	18	Control
1 0 0 1 1	13	19	Signal processing
1 0 1 0 0	14	20	Post-production (edit lists)
1 0 1 0 1	15	21	Subcodes
1 0 1 1 0	16	22	Data systems–television
1 0 1 1 1	17	23	Data systems–radio
1 1 0 0 0	18	24	Subtitles
1 1 0 0 1	19	25	Text
1 1 0 1 0	1A	26	Engineering notebook
1 1 0 1 1	1B	27	Production notebook
1 1 1 0 0	1C	28	Messages or Mail
1 1 1 0 1	1D	29	Reference data
1 1 1 1 0	1E	30	Channel setup
1 1 1 1 1	1F	31	Application description

Table 6.3 — Message address bytes used for application descriptions

Address byte		
Bits 7 6 5 4 3 2 1 0	**Hex**	**Application**
1 1 1 1 1 1 1 1	FF	Reserved for system packets
1 1 0 1 1 1 1 1	DF	Description of common applications
1 0 0 1 1 1 1 1	9F	Description of distribution applications
0 1 0 1 1 1 1 1	5F	Description of production applications
0 0 0 1 1 1 1 1	1F	Description of local applications

Table 7 — Message extension byte information

Message address extension byte 0

Message address: X X X 0 0 0 0 0	
Type of message: reserved	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Reserved

Message address extension byte 1

Message address: X X X 0 0 0 0 1			
Type of message: audio data (low grade)			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Reserved	
8	0 0 0 0 1 0 x x	|	
12	0 0 0 0 1 1 x x	|	
16	0 0 0 1 0 0 x x	|	
20	0 0 0 1 0 1 x x	|	
24	0 0 0 1 1 0 x x	|	
28	0 0 0 1 1 1 x x	|	
32	0 0 1 0 0 0 x x	|	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 1 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 1 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
- - -	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

Message address extension byte 2

Message address: X X X 0 0 0 1 0			
Type of message: time code			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	EBU time code 625/50 to IEC 461	1, 2
8	0 0 0 0 1 0 x x	SMPTE time code 525/59.96 to IEC 461	1, 2
12	0 0 0 0 1 1 x x	SMPTE (drop frame) 525/59.96 to IEC 461	1, 2
16	0 0 0 1 0 0 x x	Film time code 24 ft/s	1, 2
20	0 0 0 1 0 1 x x	Film time code 25 ft/s	1, 2
24	0 0 0 1 1 0 x x	Film time code 30 ft/s	1, 2
28	0 0 0 1 1 1 x x	Reserved	
32	0 0 1 0 0 0 x x	Film keycode 35 mm	1, 2
36	0 0 1 0 0 1 x x	Film keycode 16 mm	1, 2
40	0 0 1 0 1 0 x x	Reserved	
44	0 0 1 0 1 1 x x	Reserved	
48	0 0 1 1 0 0 x x	Sample address code	2, 3
52	0 0 1 1 0 1 x x	Reserved	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1	Data update	

1) 64 information bits of time codes, lowest numbered bit transmitted first
2) The codes of bit 1 and 0 are used as follows:
Code 0 0: the time code is a recording index
Code 0 1: the time code is time of day
3) 32-bit sample count, LSB transmitted first

Message address extension byte 3

Message address: X X X 0 0 0 1 1	
Type of message: reserved	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Reserved

Message address extension byte 4

Message address: X X X 0 0 1 0 0	
Type of message: reserved	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Reserved

Message address extension byte 5

Message address: X X X 0 0 1 0 1	
Type of message: reserved	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Reserved

Message address extension byte 6

Message address: X X X 0 0 1 1 0	
Type of message: reserved	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Reserved

Message address extension byte 7

Message address: X X X 0 0 1 1 1	
Type of message: reserved	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Reserved

Message address extension byte 8

Message address: X X X 0 1 0 0 0	
Type of message: user defined	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Any except those defined as reserved

Message address extension byte 9

Message address: X X X 0 1 0 0 1	
Type of message: user defined	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Any except those defined as reserved

Message address extension byte 10

Message address: X X X 0 1 0 1 0	
Type of message: user defined	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Any except those defined as reserved

Message address extension byte 11

Message address: X X X 0 1 0 1 1	
Type of message: user defined	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Any except those defined as reserved

Message address extension byte 12

Message address: X X X 0 1 1 0 0	
Type of message: user defined	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Any except those defined as reserved

Message address extension byte 13

Message address: X X X 0 1 1 0 1	
Type of message: user defined	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Any except those defined as reserved

Message address extension byte 14

Message address: X X X 0 1 1 1 0	
Type of message: user defined	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Any except those defined as reserved

Message address extension byte 15

Message address: X X X 0 1 1 1 1	
Type of message: user defined	
Bits 7 6 5 4 3 2 1 0	**Information**
x x x x x x x x	Any except those defined as reserved

Message address extension byte 16

Message address: X X X 1 0 0 0 0			
Type of message: data channel (transparent data)			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 0 0	Transparent data channels	
5	0 0 0 0 0 1 0 1	|	
6	0 0 0 0 0 1 1 0	|	
7	0 0 0 0 0 1 1 1	|	
8	0 0 0 0 1 0 0 0	|	
9	0 0 0 0 1 0 0 1	|	
10	0 0 0 0 1 0 1 0	|	
11	0 0 0 0 1 0 1 1	|	
--	- - - - - - - -	|	
58	0 0 1 1 1 0 1 0	|	
59	0 0 1 1 1 0 1 1	|	
60	0 0 1 1 1 1 0 0	|	
61	0 1 0 0 0 0 x x	|	
62	0 0 1 1 1 1 1 0	|	
63	0 0 1 1 1 1 1 1	↓	
64	0 1 0 0 0 0 x x	Reserved	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
116	0 1 1 1 0 1 x x	|	
120	0 1 1 1 1 0 x x	|	
124	0 1 1 1 1 1 x x	|	
128	1 0 0 0 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

Message address extension byte 17

Message address: X X X 1 0 0 0 1			
Type of message: switching / routing			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Signal source identification	A
8	0 0 0 0 1 0 x x	Signal destination identification	A
12	0 0 0 0 1 1 x x	Input routing control (user defined)	
16	0 0 0 1 0 0 x x	Output routing control (user defined)	
20	0 0 0 1 0 1 x x	Reserved	
24	0 0 0 1 1 0 x x	Routing control dialect ESbus	E
28	0 0 0 1 1 1 x x	Reserved	
32	0 0 1 0 0 0 x x	|	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	↓	
60	0 0 1 1 1 1 x x	Reserved	
64	0 1 0 0 0 0 x x	User defined	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
		↓	
92	0 1 0 1 1 1 x x	User defined	
96	0 1 1 0 0 0 x x	Reserved	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
- - -	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

A) Free form ASCII
E) EBU Tech 3245, Supplement 5: Routing switcher type specific messages

Message address extension byte 18

Message address: X X X 1 0 0 1 0			
Type of message: control			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	|	
3	0 0 0 0 0 0 1 1	|	
4	0 0 0 0 0 1 x x	|	
8	0 0 0 0 1 0 x x	|	
12	0 0 0 0 1 1 x x	|	
16	0 0 0 1 0 0 x x	|	
20	0 0 0 1 0 1 x x	↓	
24	0 0 0 1 1 0 x x	Reserved	
28	0 0 0 1 1 1 x x	ATR control dialect ESbus	E
32	0 0 1 0 0 0 x x	Logic or d.c. signaling (user defined)	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	↓	
44	0 0 1 0 1 1 x x	Logic or d.c. signaling (user defined)	
48	0 0 1 1 0 0 x x	Reserved	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	↓	
60	0 0 1 1 1 1 x x	Reserved	
64	0 1 0 0 0 0 x x	User defined	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	↓	
92	0 1 0 1 1 1 x x	User defined	
96	0 1 1 0 0 0 x x	Reserved	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

E) EBU Tech 3245, Supplement 3: Audio tape recorder type specific messages

Message address extension byte 19

Message address: X X X 1 0 0 1 1			
Type of message: signal processing			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Reserved	
8	0 0 0 0 1 0 x x	|	
12	0 0 0 0 1 1 x x	|	
16	0 0 0 1 0 0 x x	|	
20	0 0 0 1 0 1 x x	|	
24	0 0 0 1 1 0 x x	|	
28	0 0 0 1 1 1 x x	|	
32	0 0 1 0 0 0 x x	|	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

Message address extension byte 20

Message address: X X X 1 0 1 0 0			
Type of message: post-production (edit lists)			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Reserved	
8	0 0 0 0 1 0 x x	|	
12	0 0 0 0 1 1 x x	|	
16	0 0 0 1 0 0 x x	|	
20	0 0 0 1 0 1 x x	|	
24	0 0 0 1 1 0 x x	|	
28	0 0 0 1 1 1 x x	|	
32	0 0 1 0 0 0 x x	|	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

Message address extension byte 21

Message address: X X X 1 0 1 0 1			
Type of message: subcodes			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Reserved	
8	0 0 0 0 1 0 x x	Reserved for CD subcodes	
12	0 0 0 0 1 1 x x	|	
16	0 0 0 1 0 0 x x	|	
20	0 0 0 1 0 1 x x	↓	
24	0 0 0 1 1 0 x x	Reserved for DAT subcodes	
28	0 0 0 1 1 1 x x	|	
32	0 0 1 0 0 0 x x	|	
36	0 0 1 0 0 1 x x	↓	
40	0 0 1 0 1 0 x x	Reserved for future use (DCC)	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	↓	
56	0 0 1 1 1 0 x x	Reserved for future use (MD)	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	↓	
72	0 1 0 0 1 0 x x	Reserved	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

Message address extension byte 22

Message address: X X X 1 0 1 1 0			
Type of message: data systems — television			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	NICAM data	E
8	0 0 0 0 1 0 x x	Reserved	
12	0 0 0 0 1 1 x x	|	
16	0 0 0 1 0 0 x x	|	
20	0 0 0 1 0 1 x x	|	
24	0 0 0 1 1 0 x x	|	
28	0 0 0 1 1 1 x x	|	
32	0 0 1 0 0 0 x x	|	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

E) The NICAM system is defined in EBU Tech 3266: Transmission of two channel digital sound with terrestrial television systems B, G, H, and I

Message address extension byte 23

Message address: X X X 1 0 1 1 1			
Type of message: data systems — radio			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	RDS protocol	E
8	0 0 0 0 1 0 x x	Reserved	
12	0 0 0 0 1 1 x x	|	
16	0 0 0 1 0 0 x x	|	
20	0 0 0 1 0 1 x x	|	
24	0 0 0 1 1 0 x x	|	
28	0 0 0 1 1 1 x x	|	
32	0 0 1 0 0 0 x x	|	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

E) EBU SPB 490, RDS universal encoder communication protocol

Message address extension byte 24

Message address: X X X 1 1 0 0 0			
Type of message: subtitles			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Reserved	
8	0 0 0 0 1 0 x x	|	
12	0 0 0 0 1 1 x x	|	
16	0 0 0 1 0 0 x x	|	
20	0 0 0 1 0 1 x x	|	
24	0 0 0 1 1 0 x x	|	
28	0 0 0 1 1 1 x x	|	
32	0 0 1 0 0 0 x x	|	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

Message address extension byte 25

Message address: X X X 1 1 0 0 0			
Type of message: text			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Reserved	
8	0 0 0 0 1 0 x x	Reserved	
12	0 0 0 0 1 1 x x	ITTS packet	I
16	0 0 0 1 0 0 x x	Reserved	
20	0 0 0 1 0 1 x x	|	
24	0 0 0 1 1 0 x x	|	
28	0 0 0 1 1 1 x x	|	
32	0 0 1 0 0 0 x x	|	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

I) ITTS specification (see annex C).

Message address extension byte 26

Message address: X X X 1 1 0 1 0			
Type of message: engineering note book			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Permanent information not to be removed	A
8	0 0 0 0 1 0 x x	User defined	A
12	0 0 0 0 1 1 x x	|	A
16	0 0 0 1 0 0 x x	|	A
20	0 0 0 1 0 1 x x	|	A
24	0 0 0 1 1 0 x x	↓	A
28	0 0 0 1 1 1 x x	User defined	A
32	0 0 1 0 0 0 x x	Reserved	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
- - -	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

A) Free form ASCII

Message address extension byte 27

Message address: X X X 1 1 0 1 1			
Type of message: production note book			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	User defined	A
8	0 0 0 0 1 0 x x	|	A
12	0 0 0 0 1 1 x x	|	A
16	0 0 0 1 0 0 x x	|	A
20	0 0 0 1 0 1 x x	|	A
24	0 0 0 1 1 0 x x	↓	A
28	0 0 0 1 1 1 x x	User defined	A
32	0 0 1 0 0 0 x x	Reserved	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

A) Free form ASCII

Message address extension byte 28

Message address: X X X 1 1 1 0 0			
Type of message: messages or mail			
# **Bits** 7 6 5 4 3 2 1 0		**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Reserved	
8	0 0 0 0 1 0 x x	|	
12	0 0 0 0 1 1 x x	|	
16	0 0 0 1 0 0 x x	|	
20	0 0 0 1 0 1 x x	|	
24	0 0 0 1 1 0 x x	|	
28	0 0 0 1 1 1 x x	|	
32	0 0 1 0 0 0 x x	|	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

Message address extension byte 29

Message address: X X X 1 1 1 0 1			
Type of message: reference data			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Titles	A
8	0 0 0 0 1 0 x x	Program numbers	A
12	0 0 0 0 1 1 x x	IS RC	A
16	0 0 0 1 0 0 x x	Copyright information	A
20	0 0 0 1 0 1 x x	Source / Origin	A
24	0 0 0 1 1 0 x x	Program type	T
28	0 0 0 1 1 1 x x	Language	L,H
32	0 0 1 0 0 0 x x	Strand (series)	A
36	0 0 1 0 0 1 x x	Date of transmission	D,A
40	0 0 1 0 1 0 x x	Date of resending	D,A
44	0 0 1 0 1 1 x x	Duration	P,A
48	0 0 1 1 0 0 x x	Take Number	A
52	0 0 1 1 0 1 x x	Reserved	
56	0 0 1 1 1 0 x x	Reserved	
60	0 0 1 1 1 1 x x	ESCORT classification	E
64	0 1 0 0 0 0 x x	Reserved	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

A) 8-bit ASCII characters from default character set; characters transmitted in message order; bit 0 of each character transmitted first
D) Data format: yy/mm/dd [hh:mm[:ss]]; transmitted in ASCII (see note A) 8,15, or 18 characters.
L) See ISO 639 for name codes
P) Program duration format: hh:mm:ss; transmitted in ASCII (see note A) 8 characters
T) 1 byte (see annex B) program type code
E) Program classification data in EBU ESCORT format
H) Hexadecimal codes

Message address extension byte 30

Message address: X X X 1 1 1 1 0			
Type of message: channel set up			
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Reserved	
8	0 0 0 0 1 0 x x	|	
12	0 0 0 0 1 1 x x	|	
16	0 0 0 1 0 0 x x	|	
20	0 0 0 1 0 1 x x	|	
24	0 0 0 1 1 0 x x	|	
28	0 0 0 1 1 1 x x	|	
32	0 0 1 0 0 0 x x	|	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	|	
64	0 1 0 0 0 0 x x	|	
68	0 1 0 0 0 1 x x	|	
72	0 1 0 0 1 0 x x	|	
76	0 1 0 0 1 1 x x	|	
80	0 1 0 1 0 0 x x	|	
84	0 1 0 1 0 1 x x	|	
88	0 1 0 1 1 0 x x	|	
92	0 1 0 1 1 1 x x	|	
96	0 1 1 0 0 0 x x	|	
100	0 1 1 0 0 1 x x	|	
104	0 1 1 0 1 0 x x	|	
108	0 1 1 0 1 1 x x	|	
112	0 1 1 1 0 0 x x	|	
---	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

Message address extension byte 31

Message address:	X X X 1 1 1 1 1		
Type of message:	application description		
#	**Bits** 7 6 5 4 3 2 1 0	**Information**	**NOTE**
0	0 0 0 0 0 0 0 0	Default, format not specified	
1	0 0 0 0 0 0 0 1	Reserved	
2	0 0 0 0 0 0 1 0	Reserved	
3	0 0 0 0 0 0 1 1	Reserved	
4	0 0 0 0 0 1 x x	Audio data (low grade)	
8	0 0 0 0 1 0 x x	Time code	
12	0 0 0 0 1 1 x x	Reserved	
16	0 0 0 1 0 0 x x	|	
20	0 0 0 1 0 1 x x	|	
24	0 0 0 1 1 0 x x	|	
28	0 0 0 1 1 1 x x	↓	
32	0 0 1 0 0 0 x x	User defined	
36	0 0 1 0 0 1 x x	|	
40	0 0 1 0 1 0 x x	|	
44	0 0 1 0 1 1 x x	|	
48	0 0 1 1 0 0 x x	|	
52	0 0 1 1 0 1 x x	|	
56	0 0 1 1 1 0 x x	|	
60	0 0 1 1 1 1 x x	↓	
64	0 1 0 0 0 0 x x	Data channel (transparent data)	
68	0 1 0 0 0 1 x x	Switching and Routing	
72	0 1 0 0 1 0 x x	Control	
76	0 1 0 0 1 1 x x	Signal processing	
80	0 1 0 1 0 0 x x	Post-production (edit list)	
84	0 1 0 1 0 1 x x	Subcodes	
88	0 1 0 1 1 0 x x	Data systems–television	
92	0 1 0 1 1 1 x x	Data systems–radio	
96	0 1 1 0 0 0 x x	Subtitles	
100	0 1 1 0 0 1 x x	Text	
104	0 1 1 0 1 0 x x	Engineering notebook	
108	0 1 1 0 1 1 x x	Production notebook	
112	0 1 1 1 0 0 x x	Message or Mail	
116	0 1 1 1 0 1 x x	Reference data	
120	0 1 1 1 1 0 x x	Channel setup	
124	0 1 1 1 1 1 x x	Application description	
128	1 0 0 0 0 0 x x	Reserved	
- - -	- - - - - - - -	↓	
254	1 1 1 1 1 1 1 0	Reserved	
255	1 1 1 1 1 1 1 1	Data update	

Annex A (Informative)

New packets

A.1 Provided sensible data management is performed to remove redundant data, the system capacity is such that it is unlikely that a channel will be fully loaded for a sufficiently long period so as to prevent the insertion of new packets according to the rules described in clause 6. If this situation does arise, two options are available to enable new packets to be inserted.

A.1.1 Existing packets which are no longer required can be deleted and the remaining packets reordered to create sufficient space for new packets to be inserted. The number of new packets that may be inserted will depend upon the amount of free capacity created and the length of the packets to be inserted. The deletion and reordering process requires that packets be removed from one block and then reinserted in the following block, thereby resulting in a one-block delay. All subsequent packets in the channel will experience the same delay until an empty block is found.

A.1.2 Existing packets can be overwritten, but in this case there is the possibility of losing data that may be wanted. Depending upon the length of the packet to be inserted and the lengths of the existing packets in the channel, it is possible that more than one packet will be overwritten. The new packet to be inserted must be preceded by at least one HDLC start flag.

Annex B (Informative)

Code tables

This annex in under consideration. Some formats are to be defined and others are to be drawn from EBU Tech 3258 (See annex C). EBU publications may be obtained from the European Broadcasting Union, Case postale 87, CH-1218 Grande-Saconnex (GE), Switzerland.

Annex C (Informative**)**

Informative References

AR100-RvD-142, *Interactive Text transmission System (ITTS).* Eindhoven, The Netherlands: Philips Consumer Electronics, 1993.

EBU doc. SPG 2353, Geneva, Switzerland: European Broadcasting Union,

EBU doc. Tech. 3097, *EBU Time and control codes for television tape recordings (625 line television systems).* Geneva, Switzerland: European Broadcasting Union, 3rd edition 1982, re-issued 1985.

EBU Tech 3250, *Specification of the digital audio interface*, European Broadcast Union (1985).

EBU doc. Tech. 3232, *Displayable character sets for broadcast teletext.* Geneva, Switzerland: European Broadcasting Union, 2nd edition, 1982.

EBU doc. Tech. 3244, *Radio data system*, *RDS, for VHF/FM broadcasting.* Geneva, Switzerland: European Broadcasting Union, withdrawn (see EN50067).

EBU doc. Tech. 3245, *Remote control systems for broadcasting production equipment Esbus.* Geneva, Switzerland: European Broadcasting Union.

EBU Tech 3258

EBU Tech 3266, *Transmission of two channel digital sound with terrestrial television systems B, G, H, and I.* Geneva, Switzerland: European Broadcasting Union.

EBU doc. Tech. 3258, *Specification of the systems of the MAC/packet family.* Geneva, Switzerland: European Broadcasting Union, 1991.

EBU doc SPB(1992), *RDS universal encoder communication protocol.* Geneva, Switzerland: European Broadcasting Union, 1992.

EBU SPB 490, *RDS universal encoder communication protocol*. Geneva, Switzerland: European Broadcasting Union.

EN50067:1992, *Specification of the radio data system (RDS).* Geneva, Switzerland: European Broadcasting Union, 1992.

World Radio TV Handbook, New York, USA: BPI Communications, 1983.

IEC 461 (1986), *Time and control code for video tape recorders.* Geneva, Switzerland: International Electrotechnical Commission, 1986.

ISO 639:1988, *Code for the representation of the names of languages*. Geneva, Switzerland: International Organization for Standardization, 1988.

ISO/IEC 646:1991, *Information technology — ISO 7-bit coded character set for information interchange.* Geneva, Switzerland: International Organization for Standardization, 1991.

IEC 958, *Digital audio interface*, International Electrotechnical Commission, Geneva, Switzerland (1989).

ISO/IEC 4873:1991, *Information technology — ISO 8-bit code for information interchange — Structure and rules for implementation.* Geneva, Switzerland: International Organization for Standardization, 1991.

ISO/IEC TR 10735:1993, Information technology — Telecommunications and information exchange between systems — Standard Group MAC Addresses. Geneva, Switzerland: International Organization for Standardization, 1993.

ITU-R BS647-2, *A digital audio interface for broadcasting studios.* Geneva: International Telecommunications Union, 1986.