# TELETEXT TELESOFTWARE TRANSMISSIONS

TELETEXT TELESOFTWARE is an extension of the TELETEXT system. In teletext, pages of text and graphics are received by a special decoder built into a domestic television. These pages are sent in the television waveform just before the picture starts, in lines that were previously unused. The data which represents the teletext page is sent in digital form as a series of pulses. This is ideal for telesoftware as it exactly the type of data which is decoded stored and used by a computer.

A telesoftware decoder is very similar to a teletext decoder, except that it has an interface to a computer. The computer programs are broadcast with a special format on teletext pages, so that software in the home-computer can download them.

# TRANSMISSION RATE

The proportion of the teletext magazine devoted to software is an editorial decision which depends on a balance of the number of text pages, the number of programs and the access times. The specification allows full flexibility of choice. Teletext can transmit up to 32000 bytes per second, but presently we send around 8000 bytes per second. Of this, around 300 bytes per second are used for software and there are about 5 programs. The average data rate for a program is then 60 bytes per second, which is faster than a cassette tape at 300 baud. Once the programs have been received they can be stored for future use.

# HOW IT WORKS

There are two parts to describing how we transmit telesoftware. Firstly, teletext pages are joined up using pointers. Such a ‘Chain’ of pages can be any length and so any size of program can be accommodated. Secondly, the program is put on the page in a redefined format. This is called the ‘Redefinable Telesoftware Format’ or RTF for short. It describes the data on the pages to allow the decoder to interpret it.

The Telesoftware decoder is given a teletext page number containing the software to be downloaded. It searches for this page and checks it to ensure that there are no errors. The page contains a pointer, the page number, of the next page in the chain, which it then searches for. This continues until all pages in the chain has been acquired, and the program can be run. Before running the program, the data on the page has to read into the program area of the computer. To do this the page is passed through a Redefinable Telesoftware Format (RTF) decoder, a piece of software which converts the coded form of the program is broadcast, into the form required by the computer. RTF also conveys to the computer the properties of the program, such as its title and how big it is.

# A DESCRIPTION OF THE TELESOFTWARE TRANSMISSIONS

All the transmissions used for telesoftware are downwards compatible with the news and information teletext service. Some extra data has been added to enable software to be downloaded efficiently.

The functions provided by the telesoftware transmission system are:

1. Any computer language or data in binary form may be sent.
2. There is no restriction on the length of the data.
3. The data can be divided into 8-bit values of parity-checked 7-bit values.
4. The data can be compressed by replacing strings of bytes with single bytes, so saving transmission time.
5. The data is checked for errors with a cyclic redundancy check which gives a high degree of confidence that the data is correct.
6. Source code programs can be neatly displayed on a conventional decoder if desired, and so they are available to users without a telesoftware decoder.
7. The attributes of the data, such as programming language and title, are conveyed to the decoder.
8. The data may be structured as a byte-stream or as records.
9. Transmissions containing programming languages whose statements can be loaded in any order, such as BASIC, can be decoded in any order.
10. Teletext pages containing binary data, which would be confusing to users of the service, can be concealed from existing text-only decoders.

# TELESOFTWARE PROTOCOL EXAMPLE

Consider a two-page program on page number 703, subcode 1 and page 703, subcode 2. A pointer on page 703s1 points to 703s2 and a pointer on page 703s2 points to page 703s1. Both pages have a bit set to indicate that the page is for computer interpretation.

Page 703s1 starts with:

```
‖B21112‖a26EXAMPL10‖i13BBC‖p‖s÷#1÷e‖s÷Z0
```

Which is interpreted as:

| | |
|---|---|
| ‖ | Set to escape status |
| B | Command for disordered 7-bit Teletext |
| 2 | Two arguments follow |
| 1 | The first argument is one byte long |
| 1 | This is the block number |
| 1 | The second argument is one byte long |
| 2 | This is the number of blocks in the program |
| ‖ | Set to escape status |
| a | Command for title, version and date |
| 2 | Two arguments follow |
| 6 | The first argument is six bytes long |
| EXAMPL | The title of the program |
| 1 | The length of the second argument |
| 0 | The version is 0 |
| ‖ | Set to escape status |
| i | The command to send the datatype (language) and the hardware required |
| 1 | There is one argument |
| 3 | The argument is 3 bytes long |
| BBC | The datatype is BBC Basic |
| ‖ | Set to escape status |
| p | Command to inhibit automatic running of the program when loading |
| ‖ | Set to escape status |
| s | Command to redefine a byte in the lone table |
| ÷ | Operator to reduce the value of the following byte by 88 decimal |

| | |
|---|---|
| # | This byte has the value 95; so it is reduced by 88 to 7 (alpha white) |
| 1 | The string to be put in the table is one byte long |
| ÷ | The next byte is lowered by 88 decimal |
| e | This byte has the value 101 and so is reduced by 88 to give 13 (return) (This sequence forces every occurrence of alpha white to be replaced by the carriage return code. There is an alpha white at the end of every BASIC program line). |
| \|\| | Set to escape status. |
| s | Command to redefine a byte in the lone table. |
| ÷ | Operator to lower value of next byte by 88 |
| Z | Byte with value 90, is reduced by 88 to give 02 (alpha-green) |
| 0 | The second argument is length zero: (This sequence ensures that all following alpha-green codes are ignored). |

The program on the pages ends with a || c which is the end of block command. The next page in the program, page 703s2, is very similar, except that it starts with || B21212, which means that it is the second of two.

This document describes the BBC telesoftware transmissions. It is an extract from a full specification that is being considered for national and international adoption.

Teletext pages are extended to carry eight-bit data and pages are linked in chains to form larger data structures. The software and its attributes are then formatted, with a redefinable protocol, onto chains of pages.

# 5. Compatibility with the Teletext Text and Mosaic Graphics System

All the extensions are downwards compatible with the Broadcast Teletext Specification of September 1976 which should be read in conjunction with this document. There are some places where the former specification has been extended. The reference to these changes, by paragraph number, is included in brackets.

## 5.1 Packets (2.2.2)

Rows of teletext data may no longer correspond to fixed display positions and are renamed packets. These two terms can be used interchangeably.

## 5.2 Subcodes (2.1.3)

The timecode of page addresses now rarely contains the time and so it has been renamed subcode. Every teletext page should have a unique address, so that pages with the same magazine and page number should have differing subcodes. The operation of subcode addressing remains unchanged.

## 5.3 Non-teletext character bytes (1.2, 1.2.4)

Data bytes in a packet which do not represent teletext character codes are treated as 8-bit values without parity. Data protection in this case is provided by a cyclic redundancy check.

## 5.4 Packet addresses (2.1.1)

The packet addresses for data within a page are extended from 0-23 to 0-25. Packet 27 (called the Page Service Data Packet) must be accepted by a decoder as it contains linking information and error checking data relating to the current page.

## 5.5 Page Numbering (2.1.3)

As well as the decimal range for page numbering, the data pages may have addresses up to the full range available in the bits provided. That is page numbers up to FF hexadecimal, and up to 3F7F hexadecimal as subcodes. Pages with digits outside the decimal range cannot be selected on a keypad, but should be available under computer control.

# 6. Formation of a Byte-Stream

Binary data for transmission is first divided into blocks, each of which fits within an extended teletext page. The pages are then linked together with pointers to form as big a data structure as required.

## 6.1 Page Blocks of Data

One teletext page can hold 1024 bytes of data. A page block of data is taken from the centre 24 bytes of packet 0, and 40 bytes from each packet 1 to 25. Packets, with numbers in the range 1 to 25, that are not transmitted, are assumed to contain 40 bytes of the odd parity space character 2/0.

## 6.2 Error Checking

The one kilobyte block of data is checked with a cyclic redundancy check. This is best defined in terms of model hardware. A 16 bit shift register is set to the all zero state, and the data from the page is shifted into it bit by bit, after adding, modulo 2, the contents of the 7th, 9th, 12th and 16th stages. If the result does not agree with the separately transmitted cyclic redundancy check, on packet 27, then the page may be taken to be in error and it must be re-acquired.

## 6.3 Page Linking

Each page may have associated with it up to 4 page service data packets with packet number 27. Each of these is distinguished by having a different designation byte in the range 0-3. Each has 6 page link groups which point to associated pages. These link addresses are numbered sequentially 0-23. There are two sorts of page linking, branched and chained.

## 6.4 Chained Links

Pages may be joined into a chain by setting the bits in the link control byte to indicate that the page is chained. These bits also indicate the start and end of a chain. The data in the chain forms a variable length data structure. A chain structure is formed when link 0 of successive pages points to the next in the chain. Link 0 of the last page in the chain should point to the first page in the chain.

## 6.5 Branched Links

Links may contain pointers to pages containing related data, or pages that are judged likely to be selected next. These pages can be pre-grabbed in a multi-page decoder to reduce the waiting time.

# 7. The Structure of the Additional Data Packets

## 7.1 The Page Service Data Packet

There may be up to 4 page service data packets, associated with each page. Each has packet number 27 and they are distinguished by different designation bytes in the range 0-3. These packets contain pointers to associated pages, the link control byte, and the cyclic redundancy check.

### 7.1.1 The Page Service Packet Designation Byte

The page service data packet designation byte takes values between 0 and 3. Packets with values outside this range should be ignored. The byte is Hamming coded.

### 7.1.2 The Page Link Group

The page link group contains six Hamming coded bytes, giving the address of any teletext page, including magazine, page number and subcode, grouped as in the page header. The page number is in the range 00 to FF hexadecimal, and the subcode is in the range 0000 to 3F7F hexadecimal. The magazine is determined by taking the exclusive-or of the current magazine of the packet, with the three bits corresponding to the control bits C4, C5 and C6. C4 is the least significant and C6 the most significant. Note that with this scheme linking within the same magazine gives C4=C5=C6=0. Page address FF hexadecimal indicates that the page address is unspecified and subcode 3F7F hexadecimal indicates that the subcode is not specified. Links set to page FF hexadecimal and subcode 3F7F hexadecimal are assumed to be null (do not point to a page number). The page link groups on packet 27 are numbered sequentially from 0-23.

### 7.1.3 The Link Control Byte

The link control byte is a Hamming coded byte used to indicate how pages are joined together to form larger data structures. One bit is used to indicate if the page is for computer interpretation. The byte is only transmitted on the page service data packet with designation byte of zero.

### 7.1.4. The Cyclic Redundancy Check

This two byte group is the result of the 16 bit cyclic redundancy check for the page. Bits b9 to b16 are sent first, followed by bits b1 to b8. The group is only transmitted on the page service data packet with designation byte of zero.

# 9. Interpretation of the Byte Stream

The data within a byte-stream, contains the program attributes as well as program statements. These are formatted according to a Redefinable Telesoftware Protocol, although other protocols can be used if desired, by using a different escape sequence.

# 10. The Redefinable Telesoftware Format

The byte-stream provided by the chain of teletext pages is coded according to a redefinable telesoftware format (RTF). The byte-stream is passed through an RTF decoder to give the attributes

of the program, and one of more output data structures. Some of the facilities provided in RTF help the decoding, but do not give output data. RTF is table driven and operates by looking up incoming bytes in RAM tables. These tables can be reconfigured by commands within the protocol, so giving flexibility and data compression.

## 10.1 The Decoding Algorithm

Decoding is performed on all bytes between a start of block command and a corresponding end of block command. Each incoming byte is decoded by looking up one of two tables in RAM, called the lone table and the escape table. There are two possible actions, to decode to a string of bytes, or to call a command subroutine. If the string is more than one byte long, then the additional bytes are queued as part of the input stream, and so subsequently decoded with the table. This is repeated until the incoming byte-stream is exhausted, an end of file command is encountered, or until an error is found.

### 10.1.1 Lone and Escaped Status

The status may be lone or escaped. When in lone status, incoming bytes are decoded with the lone table, and when in escaped status, with the escape table. Lone status is selected when starting to decode a byte-stream. The status can only be changed by executing the ‘Set to Escaped Status’ command subroutine. (This command is equivalent to the escape code in a non-redefinable protocol).

### 10.1.2. The Lone Decoding Table

The lone decoding table is 256 entries long, each entry corresponding to one value of the input byte when in lone status. Each entry can be a string of bytes, of length 0 to 255, or can be a call to a command subroutine.

### 10.1.3 The Escape Decoding Table

The escape decoding table is 256 entries long, each entry corresponding to one value of input byte when in escaped status. Each entry can be a string of bytes, of length 0 to 255, or can be a call to a command subroutine. The status is reset to lone after reading the entry in the escape table.

### 10.1.4 Table Structure

The lone and escape tables are data structures which can vary in size, and so are best implemented with pointers and a heap to store the strings. The size of the heap should be large enough to allow at least 1024 bytes to be stored. After decoding, all the table space can be freed for use by the telesoftware programs.

### 10.1.5 Command Subroutines

Each possible function in the RTF decoder is implemented as a command subroutine. These subroutines are called when their addresses are found in the lone or escape decoding table.

### 10.1.6 Nesting of Command Subroutines

The command subroutines may be nested. In particular the ‘Raising’ and the ‘Lowering’ routines can be called within the arguments of other command subroutines. (These commands allow 8-bit values to be obtained from 7-bit data).

### 10.1.7 The Default Tables

Before beginning to decode a byte-stream the tables are set to the default values.

## 10.2 Command Structures

Command subroutines may read following bytes as arguments. Each of these must be decoded through the decoding tables.

### 10.2.1 Command Arguments

If a command has arguments, then it is followed by a number, called the field count, giving the number of argument fields which follow. Each argument field is preceded by a number giving the field length, in bytes. Some commands have a fixed number of fields, or fixed field lengths, which can be implied by context and so are omitted from the byte-stream.

### 10.2.2 Binary Numbers

If the decoder is in an 8-bit mode then the field counts and field lengths are one byte unsigned binary coded numbers, taking values between 0 and 255. Numbers with a given field length are unsigned binary coded values with the most significant bytes first.

### 10.2.3 Hexadecimal Numbers

If the decoder is in a 7-bit mode then the numbers are in character coded hexadecimal format. Field counts and field lengths in the range 0 to 15 are represented with a single hexadecimal digit in the range 0 to F. Field counts and field widths in the range 16 to 255 are represented with two hexadecimal digits, preceded by the code 5/8 or 13/8 (character X). Numbers with a given field length are represented by the specified number of hexadecimal digits. The most significant digits are sent first.

### 10.2.4 Table Overlays

Some commands in the escape table (4/0 to 5/15 and 12/0 to 13/15) are used to configure the tables to suit particular transmissions. To do this they overlay the current tables with predefined values. Only a few of these codes are currently used in Teletext.

### 10.2.5 Load and Execute Addresses

There are commands to set the load and execute address of a program. These need not be sent, in which case default values appropriate to the current datatype will be used. (For example if the datatype is Basic, then the statements are passed directly to the interpreter, which will allocate storage addresses by itself). These commands should only be used for special cases like machine code programs for a known processor.

### 10.2.6 Command Names

Each command subroutine has a three byte name. This name can be used to load it into the decoding tables.

## 10.3 The Command Subroutines

The command subroutines that are built into the decoder are listed here. For specialised applications, where the details of the receiver decoder is known, it is possible to download additional commands using telesoftware.

### 10.3.1 7-bit Teletext Overlay

If the program data is coded with 7-bits, odd parity, then the 7-bit teletext overlay will configure the tables accordingly. The decoder is set to the 7-bit mode. All the table entries for even parity codes are filled to call the error decoding subroutine. All other codes in the lone table decode to the 7-bit versions of themselves. All overlay commands, except 4/2 and 12/2, are removed, and the raising and lowering command subroutines are added. A start of block decoding subroutine is then called. This command cannot be redefined.

### 10.2.3 8-bit Teletext Overlay

If 8-bit data is sent then the tables are modified by removing the overlay commands. The decoder is set to 8-bit mode. A start of block decoding command is then called. This command cannot be redefined.

### 10.3.4 Start Block

The start block command can have zero, one to two arguments. These are the block number, starting from one, and the total number of blocks. This command cannot be redefined so that the arguments can be used to allocate space in a decoder.

### 10.3.5 End Block

The end block command always has zero arguments and the field count is implied. All data after this command and before the next start of blocks should be ignored.

### 10.3.6 End of File

The end of file command is used to terminate a stream of data. All bytes after this command should be ignored. An end of file does an implied end of logical record and end of block. It has no arguments and the field count is implied.

### 10.3.7 Start Logical Record

The data may be divided into logical records, each of any length. The start logical record command starts the record, and finishes any previous record. The command has one argument which is the record number, which should begin at one. The records may be transmitted in any order, according to the application. Records are accessed by record number, through calls from application programs. This gives some random access. The field count is implied.

### 10.3.8 Ignore Data

The ignore data command has one or more fields. Each field contains data that is ignored, but it should be decoded by the decoding tables as for all other input bytes.

### 10.3.9 Comment

The comment command has one or more fields. Each is a number of bytes coded with the teletext character set. The data in each field may be passed to the display device of the decoder as it is received, with one filed on each line of the display. It is intended that this facility is used for comments during decoding, for example to give the user confidence that something is happening during the loading of a long program.

### 10.3.10 Title, Version and Date

Of the three fields in the title, version and date command the last two are optional. The title is a string of any 7-bit codes. The version is a string of any 7-bit codes, although it is recommended that the teletext character codes for numbers should be used. The date is a string of five teletext character codes in the form of the Modified Julian Date. It should be a date associated with the program rather than the date of the current transmission.

### 10.3.11 Subtitle

There is one field which contains a string of 7-bit codes. All data following the command, up to the next subtitle has the name given by these codes. An advanced decoder should be able to select some data from within a byte-stream, by referring to it by subtitle. The field count is implied.

### 10.3.12 Datatype and Hardware

The datatype, or computer language, is a string of 7-bit codes. The second field, which is optional, is a description of the hardware required for the program. The detailed coding will depend on the datatype. All data following this command, and up to the next datatype command has the datatype specified. If the datatype is not recognised by a particular decoder then, unless it is data for a program, the following bytes should be discarded.

### 10.3.13 Set String in Escape Table

This command is used to change an entry in the escape table to decode to a string. The first field, which must be one byte long is the code to be changed, and the second field, of length 0 to 255 is the string that it should decode to. The number of fields, and the first field length, are implied.

### 10.3.14 Set Command in Escape Table

This command is used to change an entry in the escape table to call a command subroutine. The first field, which must be one byte long, is the code to be changed, and the second field, is the three byte name of the command. The number of fields, and both field lengths, are implied.

### 10.3.15 Set String in Lone Table

This command is used to change an entry in the long table send a string. The first field, which must be one byte long, is the code to change, and the second field, of length 0 to 255, the string. The number of fields, and the first field length are implied.

### 10.3.16 Set Command in Lone Table

This command is used to change an entry in the lone table to call a command subroutine. The first field, which must be one byte long, is the code to change, and the second field, of length 0 to 255, the three byte name of the command to be called. The number of fields, and both the field lengths are all implied.

### 10.3.17 Load at Absolute Address

The output byte-stream following, is loaded at the address given in the first field. The field count is implied.

### 10.3.18 Load at Relative Address

The output byte-stream of the decoder is loaded at the address given, which is relative to the beginning of the program area in the processor. If there are no arguments, then loading is in any order and the data is passed straight to a language interpreter as for many implementations of the BASIC programming language.

### 10.3.19 Execute from Absolute Address

The program is executed from the absolute address given in the first field. The field count is implied.

### 10.3.20 Execute from Relative Address

The program is executed from the relative address given in the first field. If neither absolute of relative execute address is given, then the program is to be executed from the first transmitted program byte, or the beginning of the program. Programs in source language must be compiled or interpreted as usual. If the field length is zero, then the program is executed at the default entry point, appropriate to the language of the program.

### 10.3.21 Inhibit Run when Loaded

Normally a program should run immediately, if it has been loaded without error, and if the data type is supported. If this command is executed, then the decoder returns to the computer system after loading. There are no arguments and the field count is implied.

### 10.3.22 Set to Escape Status

This command sets the decoder to escaped status to decode the next byte. It has no arguments and the field count is implied.

### 10.3.23 Error Found

This command indicates that an error has been found. The error routine should report more fully on the source of the error.

### 10.3.2. Next Byte is Converted to an 8-bit Code by Raising

The next byte to be decoded with the table is modified by stripping the parity and adding 88 to it. This value modulus 255 is used in its place. There is one argument of field length one, the byte to be raised. The field count and field length are implied.

### 10.3.2) Next Byte is Converted to an 8-bit Code by Lowering

The next byte to be decoded with the table is modified by stripping the parity bit and subtracting 88 from it. This value, modulus 255, is used in its place. There is one field, of length one, which is the byte to be lowered. The field count and field length are implied.

## 10.4 Decoding Pages out of order

Pages within a chain may be decoded out of order (and so acquired more quickly) if the particular datatype allows this. To do this, each page in the chain must begin with an overlay command which allows disordered loading. Pages of a program which must be loaded in order, should use one of the ordered loading overlay commands at the start of each page.

# TABLE 4 THE DEFAULT DECODING TABLES

## 1) The Lone Table

Every lone table entry converts to itself, except 1/11, 9/11, 7/12 and 15/12, which call the ‘set to escape status’ command subroutine.

## 2) The Escape Table

Every escape table entry call the ‘set to escape status’ command subroutine, except the following bytes:-

| Codes | Character Representation | Command or String |
|---|---|---|
| 1/11,9/11 | <esc> | Convert to themselves |
| 7/12,15/12 | ‖ | Convert to themselves |
| 6/1,14/1 | a | Title, Version and Date |
| 6/2,14/2 | b | Error |
| 6/3,14/3 | c | End Block |
| 6/4,14/4 | d | End of File |
| 6/5,14/5 | e | Start Logical Record |
| 6/6,14/6 | f | Error (in Teletext) |
| 6/7,14/7 | g | Error (in Teletext) |
| 6/8,14/8 | h | Subtitle |
| 6/9,14/9 | i | Datatype and Hardware |
| 6/10,14/10 | j | Comment |
| 6/11,14/11 | k | Ignore Data |
| 6/12,14/12 | l | Load at Absolute Address |
| 6/13,14/13 | m | Load at Relative Address |
| 6/14,14/14 | n | Execute at Absolute Addresses |
| 6/15,14/15 | o | Execute at Relative Addresses |
| 7/0,15/0 | p | Inhibit Run when Loaded |
| 7/1,15/1 | q | Set String in Escape Table |
| 7/2,15/2 | r | Set Command in Escape Table |
| 7/3,15/3 | s | Set String in Lone Table |
| 7/4,15/4 | t | Set Command in Lone Table |
| 7/5,15/5 | u | Error (in Teletext) |
| 7/6,15/6 | v | Error (in Teletext) |

| | | |
|---|---|---|
| 7/7,15/7 | w | Set to Escape Status |
| 7/8,15/8 | x | Error (in Teletext) |
| 7/9,15/9 | y | Error (in Teletext) |
| 4/2,12/2 | B | Set to 7-Bit Teletext (Disordered) |
| 4/3,12/3 | C | Set to 8-Bit Teletext |
| 4/6,12/6 | F | Set to 7-Bit Teletext (Ordered) |

Codes between 4/0 to 5/15 and 12/0 to 13/15 are used to set up different formats. Any of these codes which are unused by a decoder should call the error command subroutine.

# TABLE 5 REDEFINABLE TELESOFTWARE FORMAT COMMANDS

| Name | Number of Arguments | Action |
|---|---|---|
| TXO | 0,1,2 | 7-Bit Teletext Overlay (Disordered) |
| TXT | 0,1,2 | 7-Bit Teletext Overlay (Ordered) |
| TX8 | 0,1,2 | 8-Bit Teletext Overlay |
| DSB | 0,1,2 | Start Block |
| DEB | 0 | End Block |
| DET | 0 | End of File |
| DSL | 1 | Start Logical Record |
| DIG | 1…255 | Ignore Data |
| DCO | 1…255 | Comment |
| DTL | 1,2,3 | Title, Version and Date |
| DST | 1 | Subtitle |
| DDT | 1,2 | Datatype and Hardware |
| DES | 1,2 | Set String in Escape Table |
| DEC | 1,2 | Set Command in Escape Table |
| DLS | 1,2 | Set String in Lone Table |
| DLC | 1,2 | Set Command in Lone Table |
| DLA | 1 | Load at Absolute Address |
| DLR | 0,1 | Load at Relative Address |
| DXA | 1 | Execute from Absolute Addresses |
| DXR | 0,1 | Execute from Relative Addresses |
| DIR | 0 | Inhibit Run when Loaded |
| ESC | 0 | Set to Escaped Status |
| UER | 0 | Error in Transmission |
| URB | 1 | Next Byte is Raised |
| ULB | 1 | Next Byte is Lowered |

# TABLE 6 THE 7-BIT TELETEXT OVERLAY

The following codes are replaced by the entries given:-

| Codes | Character Representation | Command or String |
|---|---|---|
| All even parity codes | | Error in Transmission |
| Lone 14/0 | — | Next Byte is Raised |
| Lone 15/14 | ÷ | Next Byte is Lowered |
| All strings | | Converted to 7-bit strings |

All entries between 4/0 and 5/15 and between 12/0 and 13/15 are set to the ‘set to escape status’ command, except codes 4/2, 4/3, 12/2 and 12/3 which remain unchanged.

# TABLE 7 THE 8-BIT TELETEXT OVERLAY

All entries between 4/0 and 5/15 and between 12/0 and 13/15 are set to the ‘set to escape status’ command, except codes 4/2, 4/3, 12/2 and 12/3 which remain unchanged.