BBC RD 1984/19

RESEARCH DEPARTMENT REPORT

L.F. RADIO-DATA: Specification of BBC phase-modulated transmissions on long-wave

D.T. Wright, C.Eng., M.I.E.E.

Research Department, Engineering Division
THE BRITISH BROADCASTING CORPORATION
December 1984

BBC RD 1984/ 19
UDC 621.396.029.4

L.F. RADIO-DATA: SPECIFICATION OF BBC PHASE-MODULATED TRANSMISSIONS ON LONG-WAVE

D.T. Wright, C. Eng., M.I. E. E.

Summary

The BBC long-wave a.m. transmitter network carries a low bit-rate data signal, in addition to the normal programme signal modulation. The data signal is conveyed by phase-modulation of the carrier, the signal parameters being chosen so that there is no disturbance to reception on existing domestic receivers. This Report describes the data signal and includes details of both the data signal message structure and the modulation parameters.

Issued under the authority of

M. E. B. Moffat

Head of Research Department

Research Department, Engineering Division,
BRITISH BROADCASTING CORPORATION
December, 1984
(EL-173)

L.F. RADIO-DATA: SPECIFICATION OF BBC PHASE-MODULATED TRANSMISSIONS ON LONG-WAVE

L.F. RADIO-DATA: SPECIFICATION OF BBC PHASE-MODULATED TRANSMISSIONS ON LONG-WAVE

D.T. Wright, C.Eng., M.I.E.E.

1. Carrier frequency

This document* specifies the data signals carried by the BBC long-wave transmissions (Radio 4 UK) which are at present on 200 kHz. It is important to note, however, that as from 1st February 1988 the carrier frequency used for this service is scheduled to change to 198 kHz. [1]

2. Method of modulation by the data signal

The data is conveyed using linear phase-modulation of the carrier by the shaped and biphase coded data waveform (see Section 4). The phase-modulation is symmetrical about the nominal rest position of the carrier and the peak phase-deviation is ± (22.5 ± 1) degrees either side of this nominal rest position. (See Fig. 1)

The sense (advance or retard) of this phase-modulation may be deduced from Fig. 2.

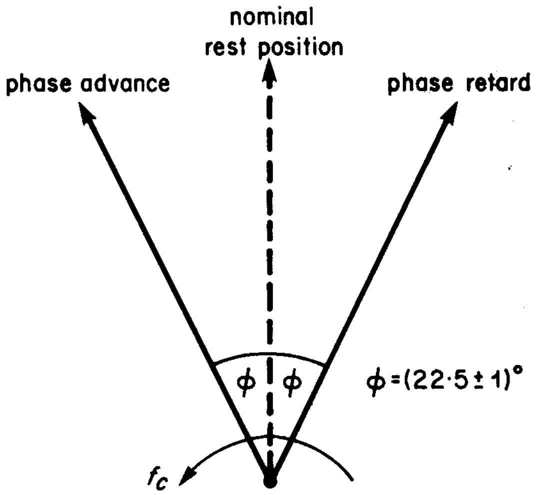


Fig. 1 - Phasor diagram of carrier modulated by data

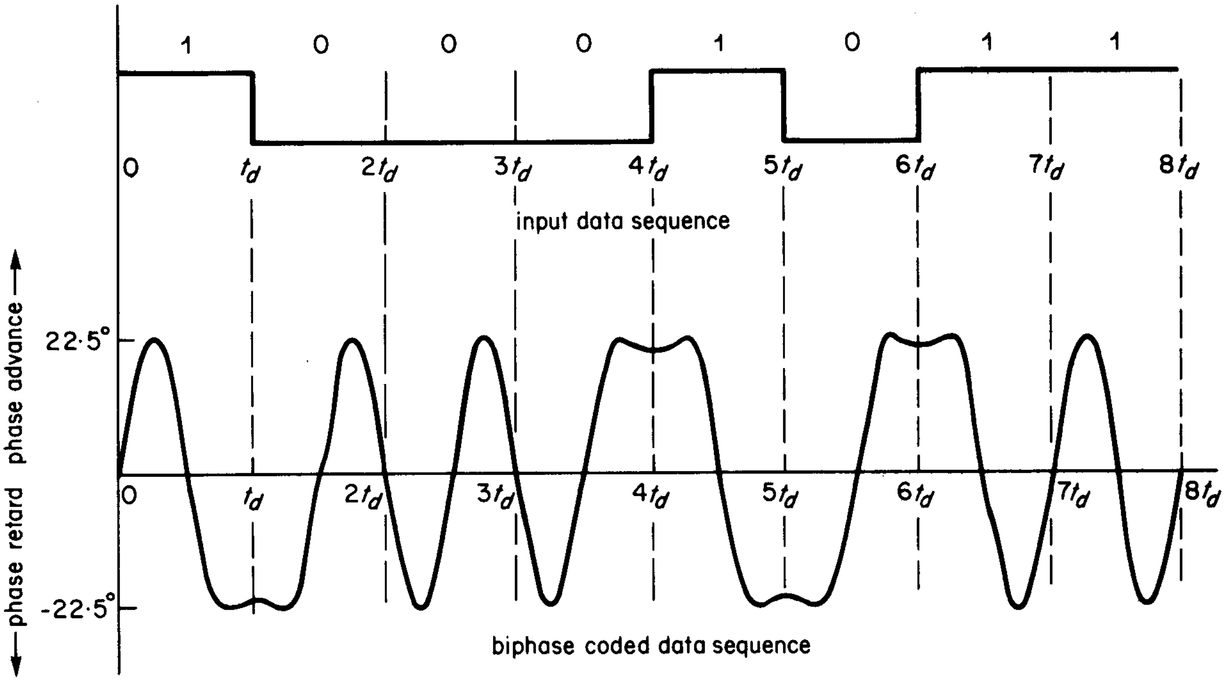


Fig. 2 - Phase of data modulated carrier as a function of time.

---

* A provisional specification was issued as Research Report 1982/2.

3. Data bit-rate

The data bit-rate of the system is nominally 25 bit/s

With two reservations, the bit-rate clock may be obtained by dividing down the frequency of the received carrier by 8000 for 200 kHz and by 7920 for 198 kHz, as shown in Fig. 3.

The first reservation is that the timing of the data waveform relative to that of the divided-down carrier may, occasionally, be deliberately shifted by up to 1 ms at intervals of not less than two seconds (50 bit-periods), in order to maintain or restore the specified timing relationship between clock-time and data blocks*. The second reservation is detailed in Section 6.2.1.

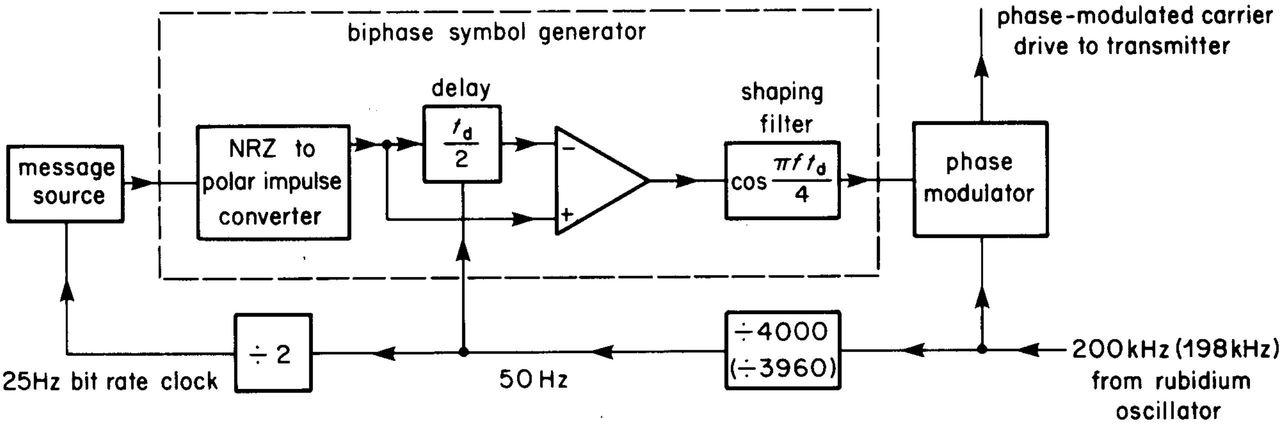


Fig. 3 - Principle of the process of generation of the shaped biphase symbols

4. Data channel spectrum shaping

To preserve the usefulness of the carrier as a frequency standard it is essential to avoid any net shift in carrier phase (averaged over a period of one second or more) caused by the phase-modulation. This requires that the modulating data signal should have small signal power at and close to d.c., and this is met by coding each source data bit as a biphase symbol. The principle of the process of generation of the shaped biphase symbol is shown schematically in Fig. 3. In concept each source bit gives rise to an odd impulse-pair, $e(t)$, such that a logic 1 at source gives:

$$e(t) = \delta(t) - \delta(t + t_d/2) \qquad (1)$$

and a logic 0 at source gives:

$$e(t) = -\delta(t) + \delta(t + t_d/2) \qquad (2)$$

These impulse-pairs are then shaped by a filter**, $H_T(f)$, to give the required band-limited spectrum where:

$$H_T(f) = \begin{cases} \cos\dfrac{(\pi f t_d)}{4} & ; 0 < f \leqslant 2/t_d \\ 0 & ; f > 2/t_d \end{cases} \qquad (3)$$

and here $t_d = 1/25$ second.

The specified transmitter and receiver low-pass filter responses, as defined in Equation (3) are illustrated in Fig. 4(a), and the overall data channel spectrum shaping is shown in Fig. 4(b).

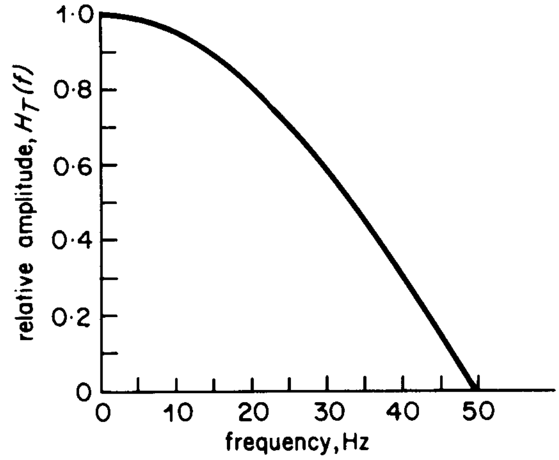


Fig. 4 - (a) Amplitude response of the specified transmitter or receiver data-shaping filter

---

[*] Such a shift would correspond to an eye-height reduction of about 2% in a receiver that was correctly phase-locked to the data signal before the shift occurred.

[**] The data spectrum shaping filtering has been split equally between the transmitter and receiver (to give optimum performance in the presence of random noise) so that, ideally, the data filtering at the receiver should be identical to that of the transmitter (i.e. as given above in Equation (3)). The overall data channel spectrum shaping would then be 100% cosine roll-off[2].

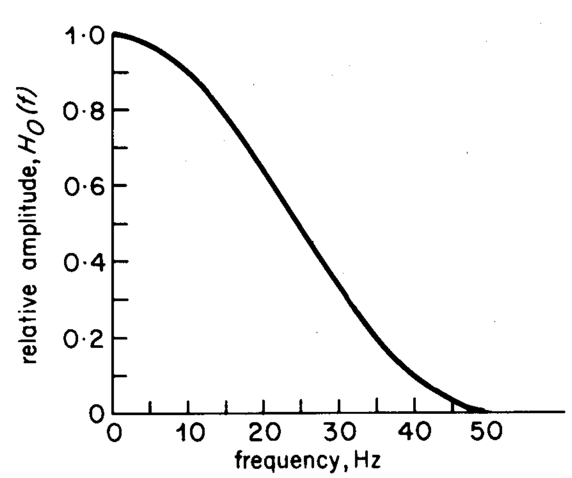


Fig. 4 - (b) Amplitude response of the combined transmitter and receiver data-shaping filters

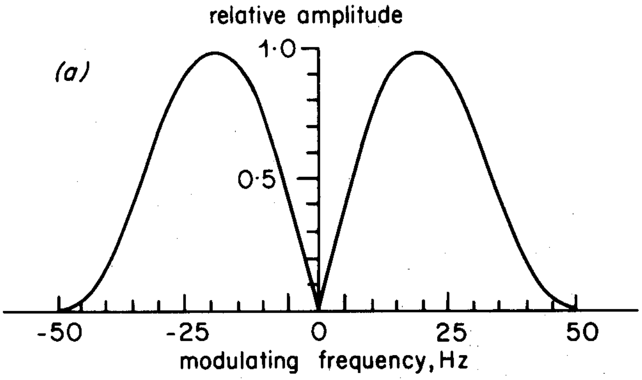


Fig. 5 - (a) Spectrum of biphase coded l.f. radio-data signals

The spectrum of the transmitted biphase-coded radio-data signal is shown in Fig. 5 (a) and the time-function of a single biphase symbol (as transmitted) in Fig. 5 (b).

In the BBC l.f. radio-data source equipment the biphase symbols are generated by direct digital synthesis of the shaped data-waveform. This ensures precise and stable spectrum shaping. The data 'eye-pattern' at the output of the specified receiver filter may be seen in the photograph of Fig. 5(c).

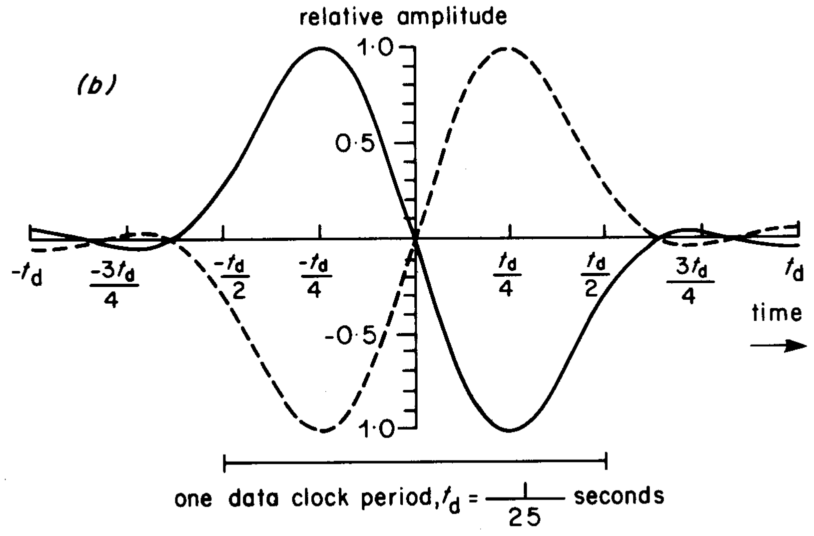
5. Data format

5.1. Block structure

The transmitted data stream is partitioned into blocks of length 50 bits each (see Fig. 6). The first transmitted bit of each block is fixed at logic 1; this prefix is part of the block synchronisation system (see Section 5.4 below).

The next four bits in each block are the application code (16 possible codes, numbered 0 to 15) and define the type of message to be expected within the block. The next 32 bits are the user message information (See Section 6).

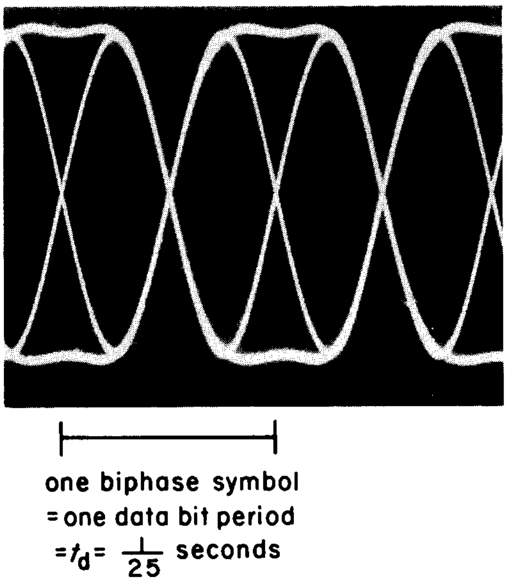
The last thirteen bits of each block are allocated to a cyclic redundancy checkword (CRC) which is used to enable the receiver/decoder to



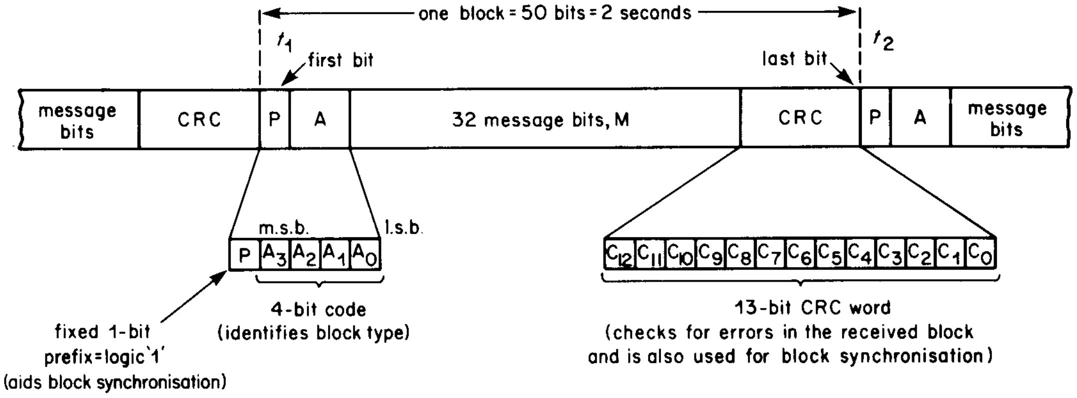
(b) Time-function of a single biphase symbol

——— symbol generated when the data input bit is a logic 1

– – – – symbol generated when the data input bit is a logic 0



(c) Eye pattern at the output of the specified receiver data-shaping filter



1) P = Fixed 1-bit prefix (= logic '1') = 1 bit (see Sections 5.3. and 5.4.)
2) A = Block application Code = 4 bits (see Sections 5.1. and 6.2.)
3) M = Message = 32 bits (see Section 6)
4) CRC = Cyclic redundancy word = 13 bits (see Sections 5.3. and 5.4.)
Total block length = 50 bits
5) $t_1 < t_2$, i.e. the prefix for a particular block is transmitted first and the CRC word last.

Fig. 6 - Structure of l.f. radio-data blocks

test for corruption of the received data by errors during transmission. This CRC will be described in detail in Section 5.3.

Each block is a self-contained 'packet' and can be decoded independently of blocks adjacent to it. Thus maximum flexibility for interleaving blocks of different types is obtained, permitting a wide range of applications to be serviced simultaneously.

Independence of the blocks also ensures maximum immunity to errors because errors in one block cannot affect the data in another.

The block length of 50 bits was chosen for two reasons:-

1) Theory and experiment have shown that this gives optimum data thoroughput efficiency in the presence of random noise. With shorter blocks the percentage overhead for check-bits and header codes increases, whilst with longer blocks the proportion of blocks received error-free decreases.

2) When transmitting clock-time information it is convenient to have an integral number of blocks per minute, so that the minute marker is always in the same position in the block. With a data rate of 25 bit/s there are exactly 30 x 50-bit blocks per minute.

5.2. Order of bit transmission

All codewords, checkwords, binary numbers or binary address values have their most significant bit (m.s.b.) transmitted first (see Fig. 6). Thus the last bit transmitted in a binary number or address has weight 2^0.

The data transmission is fully synchronous and there are no gaps between the blocks.

5.3. Error protection

Each transmitted block contains a 13-bit CRC word. This checkword is:

> The remainder after multiplication by x^{13} of the 36-bit message string (including the four-bit application code but excluding the fixed prefix) and then division (modulo 2) by the generator polynomial, $g(x)$.

Where the generator polynomial, $g(x)$, is given by:

$$g(x) = x^{13} + x^{12} + x^{11} + x^{10} + x^{7} + x^{6} + x^{5} + x^{4} + x^{2} + 1$$

It should be noted that the 1-bit prefix is not included in the formation of the CRC word. The purpose of this prefix is to provide reliable block synchronisation in the receiver (see the next Section). It also avoids 'all-zeroes' strings (no data signal present) from being falsely accepted as valid data by the receiver.

The CRC word thus generated is transmitted m.s.b. (i.e. the coefficient of x^{12} in the remainder) first, and is transmitted at the end of the block which it protects.

The above description of the CRC may be regarded as definitive but further explanatory notes on the theory of the CRC encoder and decoder are given in Appendix 1.

In the receiver/decoder, the received blocks are checked by using an identical division circuit to that at the transmitter. The decoder division register is preset to binary 1000000000000 (which is equivalent to inverting the first bit of the block, i.e. the prefix) before reading-in the first bit of a received block. All 50 bits of the received data block (including the received CRC word) are then read serially into the division register when, in the absence of transmission errors, the remainder in the division register will be 'all-zeroes'.

The CRC has the following error-checking properties[3]:

1) Detects all cases of one, two or three errors per block and any odd number of errors per block.
2) Detects any single error burst spanning 13 bits or less.
3) Detects two bursts each spanning 2 bits or less.
4) Detects 99.988% of bursts spanning 14 bits and 99.994% of all longer bursts.

The code is also capable of correcting any single burst of span 6 bits or less, but the use of the error-correcting properties of the code in a receiver/decoder greatly increases the undetected error-rate. This is because many uncorrectable error patterns are deemed correctable and thus pass undetected.

5.4. Synchronisation of blocks

In the receiver/decoder it is of course necessary to be able to recognise the beginnings and ends of the data blocks, i.e. to provide block synchronisation. Conventionally this is achieved by adding a header code to each transmitted block and ensuring that, with sufficiently high probability, the message does not mimic the header code. The penalty for using this simple system is the overhead in transmitting the header code plus any packing bits needed to avoid mimicking of the header code by data.

A more efficient scheme is to code the tranmitted blocks in such a way that the error-check in the receiver/decoder will detect block-synchronisation slip as well as additive errors. Cyclic or shortened cyclic codes are not suitable for this purpose (unless they are modified - see below) because of the fundamental weakness that cyclic shifts of codewords give rise to other code-words; thus the reliability of detection of block-synchronisation slip is poor.

Random coset codes [4,5] (in which a randomly chosen binary sequence is added to each codeword in a cyclic or shortened cyclic code), however, have good capability for detecting block-synchronisation slip; it can be shown [2,4,5] that with n-k check bits, it is possible to make the probability that synchronisation loss is undetected as small as 1 in $2^{(n-k)}$.

The operations on the cyclic code-words are all reversible at the receiver and so the normal (additive) error-detecting ability of the code is unaffected. Furthermore, with a suitably chosen random coset code, synchronisation loss can be reliably detected even in the presence of errors.

Here the random coset codewords are formed from the shortened cyclic codewords by adding the fixed 1-bit prefix (x^{49} = 1) to every codeword. In the decoder this prefix may be removed, either by complementing the first bit of each 50-bit block or (equivalently) presetting the first bit (x^{12}) of the division register to logic '1' (as noted in Section 5.3) before performing the CRC calculation. It is found that for random data the probability of synchronisation slip remaining undetected is 1 in 2^{13} per bit.

The principle, therefore, whereby block synchronisation is obtained and maintained is that, with a high level of confidence, a zero remainder in the decoder division-register will occur only once for each block received, and will occur precisely at the end of each complete block.

6. Message content

6.1. Structure

The message structure used in the present transmissions is intended to provide a high degree of freedom in assembling the information to suit the needs of the users at any given time and to allow for future developments. Apart from the fixed positions of the clock-time blocks (see below), the various blocks for different applications may be assembled in any order and in any proportion.

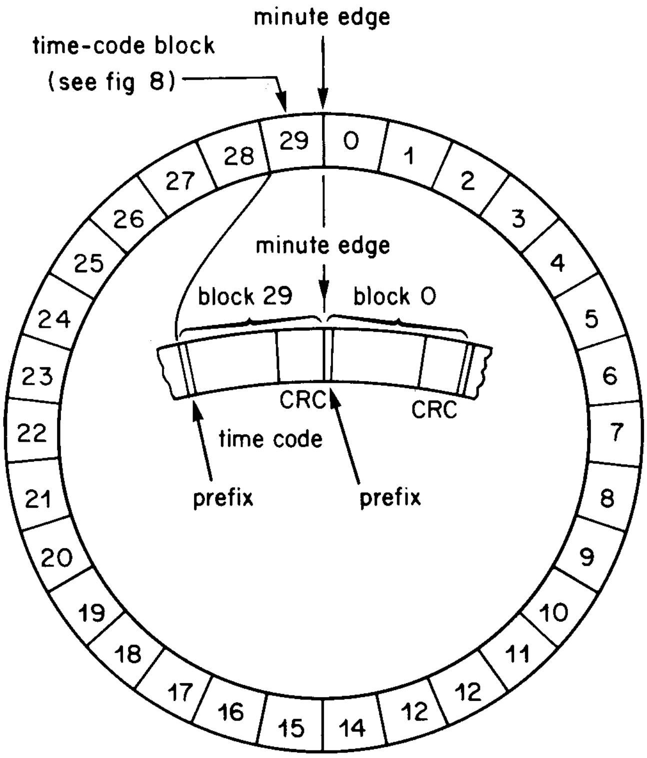
6.2. Block types

As was described in Section 5.1. (see also Fig. 6) the first four bits after the prefix bit in every block are allocated to a four-bit code which specifies the application of the block. Blocks will be referred to a Types 0 to 15 according to the binary weighting $A_3 = 8$, $A_2 = 4$, $A_1 = 2$, $A_0 = 1$ (see Fig. 6).

Type 0 is a non-user block and is reserved for time codes (see Section 6.2.1) and filler codes (see Section 6.2.2.). Other (user) block types will be allocated as the need arises.

6.2.1. Clock-time blocks

To assist in transmitting clock-time information, the block structure is normally fixed relative to clock-time minutes. Thus, in every multiple of 30 blocks (see Fig.7) it will be arranged that the time-code block occurs so that its last bit is transmitted just prior to the minute epoch to which it refers (see Fig. 7). The format of the time-code blocks is shown in Fig. 8, where it can be seen that time-code blocks are Type 0 blocks with the first message-bit set to 0.



(numbering of blocks is for illustrative purposes only)

Fig. 7 – Arrangement of blocks in a multiblock relative to clock-time

As was noted in Section 3, under normal conditions the data waveform may occasionally be deliberately shifted by up to 1 ms at intervals of not less than two seconds, in order to maintain or restore the specified relationship between clock-time and data blocks. Such small corrections should require no special action by the receiver/decoder except to track the resultant changes in phase of the bit-rate clock (see Section 3).

Exceptionally, however, discontinuities may occur in the relationship between clock-time and the data waveform. Such occurrences may be predictable, as in the case of a leap second, or unpredictable, as in the case of an equipment fault.

In order to help receivers to recognise that a discontinuity has occurred and to take the appropriate action (i.e. adjust their free-running clock-time and/or resynchronise with the data blocks), the phase-modulation will be turned off for at least 20 seconds after such a discontinuity. Receivers should take this as a signal to reinitiate searching for carrier-, bit- and block-synchronisation. Receivers should also reinitiate searching if the carrier is absent for 10 or more seconds. To minimise loss of valid data whilst receivers are re-locking, no user information blocks (Types 1 to 15) will be transmitted for at least one minute after recommencing data transmission, after such a gap.

6.2.2. Filler codes

A filler code is required to cover periods when no other suitable code can be transmitted (i.e. no new codes need to be transmitted and no previous codes need to be repeated). A filler code is also needed to fill all blocks in the first minute after a break in transmission, as described in Section 6.2.1. above.

A filler code is carried via a Type 0 block where the first message-bit following the application code is set to 1 (see Fig. 8, note 10). The remaining 31 message bits are undefined but probably will be a . . . . .0101010. . . . . sequence.

7. References

1. ITU 1980 Final Acts of World Administrative Radio Conference, Geneva, 1979, Res no. 500, p.824.

2. BENNETT, W.R. *and* DAVEY, J.R. 1965. Data transmission, New York, McGraw-Hill, 1965.

3. PETERSON, W.W., *and* BROWN, D.T. 1961. 'Cyclic codes for error detection,' *Proc. I.R.E.*, 1961, **49**, 1, pp. 228-235.

4. HELLMAN, M.E. 1975. Error detection in the presence of synchronisation loss, *I.E.E.E. Trans. Commun.*, 1975, **COM-23,** 5, pp. 538-539.

5. HELLMAN, M.E. 1974. Error detection made simple, Internation Conference on Communication 1974, Conference Record, pp. 9A1-9A4.

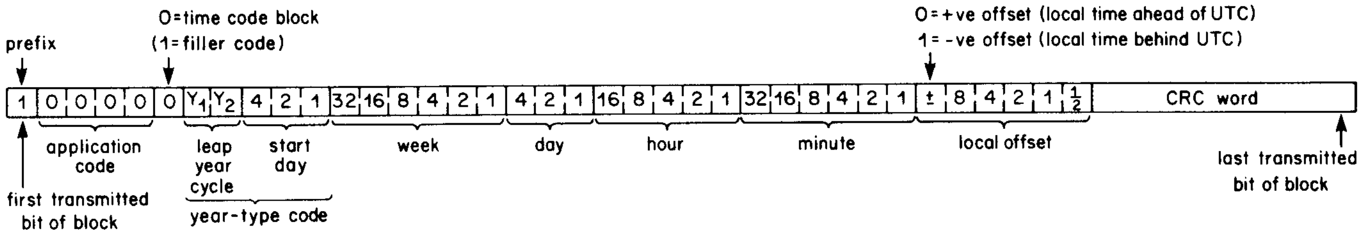


Fig. 8 - Clock-time blocks (type 0)

Notes on Clock-Time Blocks:

1) Time-of-day is expressed in terms of the Co-ordinated Universal Time (UTC).

2) The information relates to the epoch immediately following the start of the next block.
Every block normally starts on a UTC second, and every block following a UTC Clock-Time block starts on a UTC minute (i.e. 'n' minutes 0 seconds).

3) Minutes are coded as a six-bit binary number in the range 0-59. The spare codes are not used.

4) Hours are transmitted as a five-bit binary number in the range 0-23. The spare codes are not used.

5) Day-of-week is coded as a three-bit binary number, 001 = Monday to 111 = Sunday consecutively. Code 000 is not used.

6) Week number is coded as a six-bit binary number in the range 1 to 53. The spare codes are not used.

7) Start-day of Year-Type code indicates the day-of-week code (see note (5) above) of January 8th of the currently coded year, e.g. Wednesday (= 011) for 1986.

8) Leap-year cycle of year-type code indicates the position of the currently coded year in the leap year cycle as follows.

Code	
0 0	**this year is leap**
0 1	**last year was leap**
1 0	**leap year two or more years away**
1 1	**next year is leap**

9) Local Offset is a six-bit binary number which indicates the local offset from UTC in two's complement form.

Examples:	**0 0 0 0 0 0**	**local time has zero offset from UTC**
	0 0 0 0 1 0	**local time 1 hr. ahead of UTC - UK Summer (BST**
	1 1 1 1 1 0	**local time 1 hr. behind UTC**

10) When the bit following the application code is set to 1 the block should be taken as a filler code and not a time code.
In a filler code the remaining 31 message bits are undefined but probably will be a ...0101010... bit sequence.

Appendix

Theory of the Modified Cyclic Code

The data format described in this document uses a shortened cyclic block code, which is given limited self-synchronising ability by the addition of a one-bit prefix. Thus every block consists of one 'synchronisation prefix', thirty six information bits (including a four-bit application code) and thirteen check bits (see Fig. 6).

The basic code has the generator polynomial:

$$g(x) = x^{13} + x^{12} + x^{11} + x^{10} + x^{7} + x^{6} + x^{5} + x^{4} + x^{2} + 1$$

$$= (x^{11} + x^{10} + x^{5} + x^{4} + 1).\ (x^{2} + 1)$$

The thirteen check bits may be formed in the usual way for a cyclic code, i.e. by division of the message vector by the generator polynomial. The remainder of this division forms the 13 bit check word.

Thus if the 36 bit message vector $m(x) = m_{35}x^{35} + m_{34}x^{34} + ... + m_1 x + m_0$ (where the coefficients m_n are 0 or 1), the transmitted code vector $r(x)$ is given by:

$$r(x) = x^{49} + m(x)x^{13} + \left.\frac{m(x)}{g(x)}\right|_{\bmod g(x)}$$

where the term in x^{49} represents the one-bit synchronising prefix. The code vector is transmitted m.s.b. first, i.e. synchronising prefix, followed by information bits $r_{48}x^{48}$ to $r_{13}x^{13}$, followed by check bits $r_{12}x^{12}$ to r_0.

The code vector may also be calculated from the generator matrix $\underset{\sim}{G}$ which is derived from the generator polynomial. The 36 information bits are expressed as a 36 x 1 column matrix and multiplied by the generator matrix to give the information bits and check bits. The complete transmitted code vector is then formed by the addition of the synchronising prefix $r_{49}x^{49}$.

The generator matrix of the code is given in Fig. 9 in octal form (thus 36365_8 = 011, 110, 011, 110, 101).

Thus:

$$(m_{35}x^{35} + m_{34}x^{34} + ... + m_0)\ \underset{\sim}{G} = m_{35}x^{48} + m_{34}x^{47} + ... + m_0 x^{13} + p_{12}x^{12} + p_{11}x^{11} + ... + p_0$$

where

$$p_{12} = (m_{35} \times 1) \oplus (m_{34} \times 1) \oplus ...... \oplus (m_1 \times 0) \oplus (m_0 \times 1),$$

$$p_{11} = (m_{35} \times 0) \oplus (m_{34} \times 0) \oplus ...... \oplus (m_1 \times 0) \oplus (m_0 \times 1), \text{ etc.}$$

The information bits and check bits of the code vector are thus readily calculated by the modulo-two addition (i.e. exclusive OR) of all the rows of the generator matrix for which the corresponding coefficient in the message vector is '1'.

Thus for the message vector:

$m(x) = 000\ 000\ 000\ 001_8$ (where the subscript 8 indicates octal notation).

the corresponding code vector is:

$$c(x) = 00\ 000\ 000\ 000\ 036\ 365_8$$

⊕ indicates modulo-two addition (i.e. exclusive OR).

$$\begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&1&3&4&7&3\\
0&4&0&0&0&0&0&0&0&0&0&0&1&2&7&4&7\\
0&2&0&0&0&0&0&0&0&0&0&0&1&2&2&1&1\\
0&1&0&0&0&0&0&0&0&0&0&0&1&2&0&7&6\\
0&0&4&0&0&0&0&0&0&0&0&0&0&5&0&3&7\\
0&0&2&0&0&0&0&0&0&0&0&0&1&5&5&6&5\\
0&0&1&0&0&0&0&0&0&0&0&0&1&1&7&0&0\\
0&0&0&4&0&0&0&0&0&0&0&0&0&4&7&4&0\\
0&0&0&2&0&0&0&0&0&0&0&0&0&2&3&6&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&1&1&7&0\\
0&0&0&0&4&0&0&0&0&0&0&0&0&0&4&7&4\\
0&0&0&0&2&0&0&0&0&0&0&0&0&0&2&3&6\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&1&7\\
0&0&0&0&0&4&0&0&0&0&0&0&1&7&1&3&5\\
0&0&0&0&0&2&0&0&0&0&0&0&1&0&5&2&4\\
0&0&0&0&0&1&0&0&0&0&0&0&0&4&2&5&2\\
0&0&0&0&0&0&4&0&0&0&0&0&0&2&1&2&5\\
0&0&0&0&0&0&2&0&0&0&0&0&1&6&1&2&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&7&0&5&0\\
0&0&0&0&0&0&0&4&0&0&0&0&0&3&4&2&4\\
0&0&0&0&0&0&0&2&0&0&0&0&0&1&6&1&2\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&7&0&5\\
0&0&0&0&0&0&0&0&4&0&0&0&1&7&2&3&0\\
0&0&0&0&0&0&0&0&2&0&0&0&0&7&5&1&4\\
0&0&0&0&0&0&0&0&1&0&0&0&0&3&6&4&6\\
0&0&0&0&0&0&0&0&0&4&0&0&0&1&7&2&3\\
0&0&0&0&0&0&0&0&0&2&0&0&1&7&6&2&3\\
0&0&0&0&0&0&0&0&0&1&0&0&1&0&6&6&3\\
0&0&0&0&0&0&0&0&0&0&4&0&1&3&2&4&3\\
0&0&0&0&0&0&0&0&0&0&2&0&1&2&4&5&3\\
0&0&0&0&0&0&0&0&0&0&1&0&1&2&3&5&7\\
0&0&0&0&0&0&0&0&0&0&0&4&1&2&0&1&5\\
0&0&0&0&0&0&0&0&0&0&0&2&1&2&1&7&4\\
0&0&0&0&0&0&0&0&0&0&0&1&0&5&0&7&6\\
0&0&0&0&0&0&0&0&0&0&0&0&4&2&4&3&7\\
0&0&0&0&0&0&0&0&0&0&0&0&3&6&3&6&5
\end{bmatrix}$$

Fig. 9 - Generator Matrix for the Basic Cyclic Redundancy Check (in Octal Notation)

$$g(x) = x^{13} + x^{12} + x^{11} + x^{10} + x^{7} + x^{6} + x^{5} + x^{4} + x^{2} + 1$$

which may be seen to be the bottom row of the generator matrix.

After adding the synchronising prefix the transmitted code vector is

$$r(x) = 20\ 000\ 000\ 000\ 036\ 365_8$$

Similarly for all 1s message vector

$$m(x) = 777\ 777\ 777\ 777_8$$

it follows that

$$c(x) = 17\ 777\ 777\ 777\ 762\ 722_8$$

which on adding the prefix becomes

$$r(x) = 37\ 777\ 777\ 777\ 762\ 722_8$$