A MUSIC COMPILER

1. INTRODUCTION.

The idea of playing music on the loudspeaker of EDSAC 2 is by no means a new one. The engineers have played tunes with the loudspeaker connected to a digit of the accumulator, and there have been proposals to use the manual register as a keyboard (see J.D. Roberts, Diploma Dissertation, 1958). However the full potentialities of the instrument do not appear to have been exploited, mainly because the programming of virtuoso pieces has been too tedious.

The purpose of the Music Compiler is to enable the music to be specified on a data tape in a form to which a musical score can easily be converted. Indeed it should be possible to punch the data tape direct from the score provided that the music is not too complicated. Most of the usual musical facilities are provided, for example: staccato notes, repeats, setting of key and tempo.

The notes are produced by sending a regular succession of pulses to the loudspeaker. The loudspeaker is connected to digits in registers p , w and D . When the output register D is not in use, the contribution from register p predominates. A pulse is produced whenever the 32-digit of p changes from 0 to 1 or vice versa. Register p holds the obeyed address of an order. Further details are given in section 3.

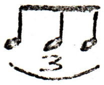
2. THE DATA TAPE.

Pitch and duration of notes. The muscial scale is divided into octaves, each one starting at a C . Octave 0 is the octave from middle C upwards, and the octaves above and below this are numbered 1, 2, 3, ... and -1, -2, -3, ... respectively. Having specified the octave containing a particular note, the pitch of the note is then completely defined by giving its name, i.e. A, B, C, D, E, F or G , possibly with an accidental (sharp, flat or natural). Sharpened or flattened notes are regarded as belonging to the same key as their 'parent' notes. Thus the note C flat one semitone below middle C is in octave 0 , whereas the same note called B is in octave -1 . Each note is played at the correct pitch, using the equal-tempered scale, irrespective of the speed of the machine.

The duration of a note is specified by the tempo of the music and the time-value (i.e. duration relative to the tempo) of the note. For the time-value of a note, we adopt the convention used in time signatures in musical scores of representing a note by the number of them required to make up a whole note, or semibreve.

Thus,	1	=	semibreve	𝅝
	2	=	minim	𝅗𝅥
	4	=	crotchet	♩
	8	=	quaver	♪
	16	=	semiquaver	𝅘𝅥𝅯
			and so on.	

A dotted note is represented by a decimal point after this integer, followed by the number of dots. For example 𝅗𝅥. is punched as 4.1 . In such a combination the integer 4 will be referred to as the basic time-value.

Triplets and similar combinations of notes are easily dealt with. For instance, each of the notes  (3 quavers in the time of 2) has a time-value of 12 .

Rules for punching the data tape. The data tape consists of a sequence of items, each of which must be terminated by either two spaces or cr, ~~lf~~ lf . Single spaces are ignored everywhere.

Items may be of the following kinds:

1. Key setting. The key of the music is set by an item of

the form KEY ns for sharp keys, or
KEY nf for flat keys, where n is an integer between 0 and 7 , indicating the number of sharps or flats in the key signature.

If the music is to be transposed, that is, played at a different pitch to that at which it is written, the form
KEY 3s/k is used, indicating that the music is written with a key signature of 3 sharps (say), but is to be played k semitones higher $(-12 \leqslant k \leqslant 12)$. This facility will be found useful when playing music written for a transposing instrument such as the clarinet.

If no key is set before the first note, C major (or A minor) is assumed. This key can be represented by either KEY 0s or KEY 0f .

2. Tempo setting. The tempo of the music is set by an item of the form TEMPO m=n . This is equivalent to a metronome indication (e.g. ♩ = 96) in a musical score. Here, m represents the time-value of the note, punched according to the rules already described, and n (a positive integer) is the metronome rate, that is the number of notes of time-value m in one minute. A tempo must be set before the first note is read.

3. Octave setting. An item of the form

OCTAVE n , where n is a positive or negative integer, causes succeeding notes (up to the next 'Octave' item) to lie in this octave. Octave 0 is assumed to be set initially. Note that in the case of transposed music, the octave number rofers to the written notes, and not to the pitch of the music when played.

4. Notes and rests. An item of the form

X k m represents a note to be played,

where X is A , B , C , D , E , F or G

k is s (meaning sharp) or f (flat) or n (natural) or may be omitted

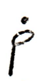
m is a time-value punched according to the rules described already.

Examples:	A4	A, 𝅗𝅥
	Bf8.2	B flat, ♪..
	Fn1	F natural, 𝅝

If k is omitted, the note may be sharpened or flattened according to the key selected. For instance in the first example above, the note would be played as A flat if the key was E flat major (KEY 3f) . If the note contains an accidental (s, f or n) the key setting is ignored for that note.

A rest (silence) is denoted by an item Rm , where m is the time-value.

5. Staccato notes. A note to be played staccato is denoted by a decimal point preceding the note item; this causes the duration of the note to be halved .

e.g. .Cs2 represents C sharp, 

This is equivalent to Cs4 R4 .

6. Phrasing. Unfortunately, the programmer has no control over the loudness of the music and this makes phrasing and the stressing of individual notes rather difficult. The only solution seems to be to separate phrases by very short pauses.

This is denoted by the character + , which must be punched as a separate item, and causes a slight pause to be inserted after the last note read. This facility will also be found useful for separating two notes at the same pitch which would otherwise tend to be heard as one note.

7. Repeats. A simple repeat of a sequence of notes can be arranged by placing the notes between the two items (and) .

This facility can be extended as follows:

(n   where m is an integer greater than 1 , means "play the following sequence of notes, up to the next) , n times". Thus if n is omitted, it is taken to be 2 .

)m   where m is a positive integer, means that the repeat is to exclude the last m items, where

(i)   'items' here means notes and rests taking account of any intervening repeats

(ii) a staccato note counts as 2 items

(iii) a pause, + , does not count as an item.

Thus if m is omitted, it is taken as 0.

Examples:

(A8 B8)   played as A8 B8 A8 B8

(3 C4)   played as C4 C4 C4

(G2 Fs8 E8)1   played as G2 Fs8 E8 G2 Fs8

Repeats can be nested inside each other at will. For example the repeat of the constituent notes of a trill could occur inside the repeat of a few bars, which itself is part of a repeated movement. There is no practical limit to the depth to which repeats can be nested.

8. End of music. This is denoted by the item

STOP or, if the music is to be played repeatedly without stopping, the item

CYCLE .

9. Wait during input. An asterisk punched as a prefix or as a separate item causes the machine to stop reading the tape and light the 'wait' lamp.

Method of Use.

(i) Read in the Music Compiler using Set Start and Clear. The machine will then measure its own speed (this takes 1 second~~s~~), print this as the number of microseconds per microprogram step (for anyone who is curious) and then wait.

(ii) Read in the data tape using the Run key. On reaching 'STOP' or 'CYCLE', the machine waits with 2047 displayed on the control panel neons.

(iii) On raising and lowering the Run key, the music will be played after an initial pause of about 2 seconds. If the last item on the data tape was STOP, when the end of the music is reached there will be another 2-second pause, then the machine will wait, ready to read in a further data tape.

Detection of Errors. If the machine encounters a fault in the data, it prints an input report and halts with the Optional Stop lamp lit, in the manner of the Assembly Routine. On pressing the Reset button, reading of the data tape is resumed, but when 'CYCLE' or 'STOP' is reached, the machine will stop with the Report lamp on.

A fault in the nth line of data causes a report m/n where m is a report number identifying the fault (see list below). Since a report causes the remainder of an item to be ignored, an error may cause bogus reports elsewhere. For example an error in the first tempo-setting item would result in the tempo appearing to be unset, and would cause reports on succeeding notes.

The error detection facilities are not claimed to be foolproof, but should trap all the likely errors.

List of Report Numbers.

432 Integer omitted or contains illegal character.

479 Staccato mark not followed by note.

495 Incorrect item beginning with C .

503 One or more repeats not completed i.e. (not followed by) .

515 Incorrect item beginning with S .

568 Note not followed by time-value.

590 Note too low or too high.

611 Tempo not set.

624 Note too long.

628 Note too short.

647 Basic time-value wrongly terminated.

649 Number of dots negative or wrongly terminated.

680 Illegal character after + .

687 Incorrect item beginning with T .

690 No metronome rate in Tempo setting.

696 Metronome rate negative or wrongly terminated.

732 Incorrect item beginning with O .

734 Illegal character after Octave number.

745 Incorrect item beginning with K .

748 Integer after KEY negative, or greater than 7 , or not followed by f or s .

768 Illegal character after f or s in Key setting.

772 Attempted transposition by more than 1 octave, or number of semitones wrongly terminated.

781 First character of item illegal.

790 Number following (less than 2 or wrongly terminated.

818 Number following) negative or wrongly terminated

822) not paired with (.

838 Number of notes to be repeated negative or zero.

875 Too many notes on data tape. (There is room for 16381 notes and rests.)

Integers are terminated by double space, cr , f , s , • , or = according to context. "Wrongly terminated" means that the terminator used is meaningless in the context.

Further Notes.

(i) If the interrupt button is pressed (with the switch set to Manual) during the playing of the music, the machine returns immediately to the Wait which occurred after reading in the data tape, thus enabling the music to be restarted.

(ii) The lowest note than can be played depends on the speed of machine. At a speed of 2.65 microsecs./step the lowest note is the F sharp in Octave -2 , at about 93 cycles per second . This can be sustained for 12 seconds . There is no practical upper limit to the pitch obtainable, but the higher notes cannot be sustained for as long as the lower ones. The top note of a piano (C in Octave 4 , or 4186 cycles per second) can be held for only 1/4 second .

(iii) If no output device is connected to Channel 1, and an error is detected, the machine will wait and will not proceed until one is connected.

(iv) The letters s , f and n when representing sharp, flat and natural may appear in either figure-shift or letter-shift; otherwise characters must be punched in the correct shift. Subject to this requirement any number of letter-shifts and figure-shifts may be punched. Thus letter-shift is allowed inside an integer provided it is followed by a figure-shift before the next digit.

3. DESCRIPTION OF THE PROGRAM.

The basic cycle which sounds a note is

```
66tm
75s-1pl
```

, where m is chosen according to the frequency of the note required, and s is set initially to twice the number of cycles required. Each cycle takes 35 + 2m microsteps. If the 32-digit of m is different from the 32-digit of the addresses where these orders are stored, a pulse will be sent to the loudspeaker when entering the 75-order , and another when entering the 66-order . These two pulses can be thought of as one pulse since the interval between them is less than 60 microseconds . Thus the loudspeaker receives a pulse for each cycle and hence the note is generated.

The details of each note occupy one register in the main

Correction

page 77, line 15: Replace from "A count..." to end of paragraph by:

> A cycle of instructions is obeyed which increments a count in b for 50 complete mains cycles, i.e. 1 second. Knowing the number of microsteps for each of the orders involved, we can then calculate the machine speed.

store, in the form 0km , 127td , where k is s if the 32-digit of m is a 1 , and is f otherwise, and d is the initial value for s (twice the number of cycles). If d is odd, the cycles end with s = -1 , causing a jump to another 66-order giving the short pause required for the + facility. The sign digit k determines which of two cycles shall be obeyed, so that the 32-digit of m is different from that in the addresses of the orders of the cycle. A rest is stored in exactly the same way, except that k is set 'wrongly' so that no sound is heard.

When the program starts, the machine speed is measured, using the fact that digit 23 of M following a 120f16 order contains 1 or 0 depending on the phase of the mains voltage. Thus this digit changes every 1/100 th of a second. A count is carried out during alternate half-cycles of the mains; this is continued over 150 half-cycles for the sake of accuracy, and the speed is then calculated using the fact that the counting cycle takes 64 microsteps.

Characters from the data tape are read by a subroutine (entry at p6) which exits with the character in t and ȧ(7) having ignored single space, and passed over figure-shift and letter-shift, while keeping a marker in register 30 to denote the current shift. Characters of the standard words like KEY and OCTAVE are read by a subroutine (entry at p69) which exits with the characters packed into the accumulator after reading a specified number of them, or if double space or c/r is encountered. A character in figure-shift causes an exit with the accumulator equal to zero.

Register 32 holds an indication of the state of the current item: whether it is incomplete or terminated, and if the latter is the case, whether by double space or c/r . This is used to determine the course of action when an item has been dealt with, and is also useful to the Report Routine in deciding whether there are any characters to ignore.

Numbers are read by an Integer Input Routine which exits on 60f0 after the terminators = , · , f or s , and on 60f1 after double space or c/r . An input report occurs if an illegal character is read or if a terminator is read before a digit.

When the first character of an item has been read, a jump is made to the section of the program which deals with the type of item that begins with this character. This is accomplished by reference to a table of jump addresses (at p62). For the characters A , B , D , E , F and G the address in the table is not a jump address but an integer representing the relative pitch of the corresponding note.

If the first character read is an * or C , tests have to be made to determine whether the item is CYCLE or the note C or an asterisk prefix.

The procedure for compiling a note is as follows:

(i) Read next character to see if it is s , f or n .

(ii) If it is not, adjust relative pitch according to key selected.

If n , do nothing.

If s or f , adjust relative pitch by 1 semitone.

(iii) Transpose if necessary.

(iv) Form μsecs/cycle as if in Octave 0 , using look-up table.

(v) Form correct μsecs/cycle according to current octave number.

(vi) Convert to address for 66-order , and form modifier according to 32-digit .

(vii) Read time-value.

(viii) Convert to number of cycles, using current tempo.

(ix) Halve length of note if staccato, and plant details of note in main store.

(x) If staccato, change modifier to form rest, and plant in main store.

Repeats. A list is kept in the free store from 1900 onwards containing one word for each outstanding repeat. This word contains the location in the main store of the first note of the repeat and the number of times the sequence of notes is to be copied (i.e. one less than the number of times it is to be played). When the) corresponding to this repeat is read the copying cycle is entered. The location of the entry in the list corresponding to the most recently read (, that is the innermost repeat, is held in a(20) . The list can hold up to 70 entries at a time, that is, repeats nested to a depth of 70 !

Use of store.

Free Store:	0-33	Working space (see below)
	34-46 64-70	Playing routine
	200-223	Constants
	300-1035	Compiler program
	1900 onwards	Repeat list
	2041	Jump for interrupt.
Main Store:	0	Indirect address (during compiling) Initial pause (during playing)
	1 to n	Data for playing routine (n notes and rests)
	n+1	Final pause (omitted if music is to cycle)
	n+2	Zero, to mark end of music.

Working Space (free store). All numbers are held in fixed-point form.

0	Subroutine links	
2	Count of lines of data	
4	Octave number	
6	a(7) = character just read	
8	Key word -- indicates notes to be sharpened or flattened for current key	
10	Tempo (µsecs/semibreve)	
12,14	General working space	
16	µsecs/cycle of current note	
18	Word for main store	
20	a(20) = pointer for repeat list	
22	35u	where u is the machine speed in µsecs/step
24	1/2u	
26	Transposing constant	
28	General working space	
30	Current shift : 0 if figure-shift 27 if letter-shift	
32	State of current item : 0 if item not complete 2 after c/r 30 after double space	
464 (= p8)	Storage for subroutine links.	

4. ANNOTATED PROGRAM

```
25/82
t
tMUSIC COMPILER
t
tMACHINE SPEED
t(MICROSECS. PER MICROSTEP)

F1=34
                         Playing Routine
6682016        Pause for + facility
11Lf*1(90
10f*0          Fetch next note
52f0(53        Jump if end of music
15f2
0033           Set duration
00+2
55f57          Jump according to 32-digit
75+f50          Set pitch
66+2(56      } Play note
755-1f1      }
280
50f50

F1=04

75+f55(57       Set pitch
00+0(55       } Play note
758-1f1       }
280
50f50

11Lff1(22
50f1f50         Restart music (for CYCLE facility)
```

Measure Machine Speed.

```
p1=301

70s102   <== START
70tp82
112ff0(82
88tp91
112f*1(90   }
120f16      } Count in b during 1st half-cycle
62p52       }
52p90       }
112f*1(89   }
120f16      } Count in b during 2nd half-cycle
62p52       }
53p89       }
75sp90(91     Count no. of cycles (jumps to p82 first time)

108f2
8f2
34f218       52 b
32f224       52 b + 970
8f2
47f226        1_10 8
35f2
19f2         Store machine speed, u, in 2
59f22        Print u
107f2
107f8
107f8

34f220
39f22         35u
10f222
35f2
19f24        1/ 2u

2/.67502005

102f0(94       WAIT for data tape
46f0     }
49f2     }
49f6     }
9f10     }   Clear working space
9f18     }
9f20     }
9f26     }
70s1     }
70sp50   }   Set program to initial state
70sp19   }          (in case it is restarted after input reports
70s1p46  }
112ff0       Set b = 0
50p44
```

Report Routine

```
50R1058       On first entry:  Set report at end of input
70†14
75†R-2
70†f32     }
75†1f46    }
120f16     }
0f2        }
66†4       }   Check output device on Channel 1
77†1       }   Wait if none connected
50R3       }
102f1365   }
50R-6      }
107f31
107f2
107f6
70†-1      }  Suppress page layout
55f5       }
05†1       }
46†-1      }  Report number
55f21      }
107f11
4f5        }
46f0       }  No. of line of printing
55f21      }
107f2
107f6
120f0          OPTIONAL STOP
5f16           Clear 18 ready for next item
46†-1m102  }   Jump if report occurred after CYCLE or STOP
52f32      }
60†33      }
77†1       }
50R2       }
50f6       }   Ignore ~~[illegible]~~ rest of item if terminator not already read
46†-30     }           then jump to p47, or p44 if end of line reached
52f47      }
46†-2      }
52f44      }
50R-5      }

17·6404311505
```

Integer Input Routine

```
10f0(30 <==   Entry to read entire number
19p6          Preserve link
56p6        . Read first character
50R3

10f0(25 <==   Entry with first character already in t and a(7)
19p8          Preserve link
70s1        } Set +ve sign indication in 12.
79s12       }
70s0          Set stage 0
9f14
50R4

2f1(31        Set stage 2  (after digit)
70s1          Set stage 1  (after + or -)
58p6          Read next character
50t1p1        Jump on character

50p127        ll
50s1p76       f
50p74         cr
50p76         0
50p127        r
50p76         7
50p76         2
50s1p76       s
50p127        ef
50p76         8
50p76         5
50p127        /
50p76         9
50p12.7       (
50p73         =
50p127        ₁₀
50p127        p
50p76         4
50p76         3
50p127        ²
50p76         6
50p127        t
50p127        n
50p77         —
50p76         1
50p127        *
50p127        )
0f0           ls
50p73         •
50s1p76       +
50p74         sp
50f1f31(70      Enter stage 1
2f0             Report
58f50(127       Report
10f0            Terminate
15f0         Restore link
10f14
4f12        }  Negate if - sign read
50R1        }
31f14       }
77t30
60f1           Exit for cr, sp.
60f0           Exit for ·, =, f, s.

10f30(73    }  Jump if in lettershift      (· or =)
53f127      }
50S1f70        ~~[illegible]~~

75t33(74       Mark end of item
70t30          Set t = 30 for exit fork     (cr or sp)
50S1f70

10f30(77    }  Jump if in letter-shift
53f127      }  Set -ve sign indication in 12     ( − )
9f12
50Sf70

                    Digit Cycle
10f30(70       Jump if in letter-shift
53f127
110t551     }
00t5        }
15f14       }  Multiply existing number by 10
0f1         }      and add new (translated) digit
15f14       }
0f2         }
50f14       }
46t0        }
35f14       }
50f51

0(0            Storage for subroutine links
               ____________________________________
10f10          (After *)
55f100         Jump if · already read
102f0          WAIT on * prefix
5f52(27        Start of item
50f0(40        Read 1st character
2t0
00Sf02
70S200         Jump on A, B, D, E, F, G  "
50f25
70Sf101        Jump on * or C
5050
10f10          Jump if · not already read
5250              otherwise report
58f50(100
                                   * or C
10f30(40
52f-3p47        Jump if *
50p0            Read next character
40+-10
53p20           Jump if not Y or 3
10f30
52p20           Jump if 3
10f10
53p100          Jump if • already read
                                                  "CYCLE"
70S0         )
58p69        |
33p60        |  Read next 4 characters
52R4         |  Report if not  C L E  followed by terminator
33p01        |
52R2         |
58p50(101    )
70Sp24       }  Plant address in 52-order of playing routine
79Sp53(46       Plant end-of-music marker in main store  (=0).
50p19
10p63        }  Plant initial pause in main store
19ff0
60S20        }
77S1900      }  Report if any repeats not completed
58p50(102    }
102f2047        WAIT before playing ~~[illegible]~~ music
112ff0(99        b=0
50f1p58          Start music
                                                  "STOP"
10f30(00
52p01           Jump if in figure-shift
70S0         )
58p69        |
33p02        |  Read next 4 characters; report if not
52R4         |       T O P  followed by terminator
33p01        |
52R2         |
58p50(103    )
10p63        }  Plant final pause in main store
50p19
70Sp92
51p46

70S2(6 <=          Subroutine to read characters, ignoring single space
69+7               and noting current shift.
75+31                     N(30) = 0   if in figure-shift
50R3                              27  if in letter-shift
9f30
50R-4
77+27
77+26
50R3
75+31
50f1p6
70+30
60f0 =>
75SR-1
50f1p6
```

Subroutine to read characters of standard words (e.g. TEMPO)

No. of characters to be read in s

Exit after s characters read
or on cr or double space
or on character in figure shift

```
⟹10f0(65  } Preserve link
  15p0
  51R1
  50p6      Read character
  77t30
  50R4
  77t2
  77t3
  50R4
  75t33     If cr or sp, mark end of item
  7082      and prepare to exit
  50R4
  00t31
  70t27
  51R4      Clear M if figure-shift set (so that report occurs after exit)
  6f5       } Pack character into M
  32f6
  7583p65
  80Sp8     }
  4f1p6     } Restore s and exit
  50f0 ⟹   }

                                    data
  106f40(32 } Light REPORT lamps (at end of tape, after input reports)
  0f0

  7080(26   ⟸ C
  50R4

  10f30(45  ⟸ A, B, D, E, F, G
  52p01       Jump if figure-shift set.
  50p6      }
  100tp62   } Jump on next character
  00t3      }
  50t1p1    }

  00t7      }
  75t33     } If cr or sp, mark end of item and report
  50p50(104 }

  7284    If s, raise by 1 semitone
  7382    If f, lower by 1 semitone
  50F75   Skip key test if f, s, or n

  10f8      }
  082       }
  54F75     }
  6f1       } Adjust s according to key
  55R2      }
  7284      }
  7382      }
  02827R75    Transpose
  108p65      Basic μsecs/cycle
  0085      } μsecs/cycle = x     (taking account of current octave no.)
  750       }
  35f10       Store x
```

```
33f24    }   x - 35u
34f24    }   -------  = address for 66-order      (u = machine speed)
               2u
35f12
55R3         Report if -ve   (note too high)
33f200
55R2         Report if > 2047  (note too low)
58p50(105
60S13
75S16        Plant address for main store
46S0
62p66
52R2
2f256
70S0
65S16        Plant modifier according to 32-digit
76t6
50p34        Jump if last character was f, s, or n
60t7             otherwise enter integer input subroutine with
2p25n30          1st character already read.
58p30(34     Read integer (basic time-value, m)
50p45        Jump if terminator  f, s, ·, or =
70t0         t = 0  if not dotted note
6t0(42       }
34f16        }  N(14) =  x m 2^t
46f14        }
30f10         Tempo, T   (μsecs/semibreve)
55R2
58p50(106     Report if tempo not set

6t1       }
35f10     }
15f12     }  N(L) =  T(2^(t+1) - 1)
51R1      }
47f12     }
35f14        No. of cycles
60t19
76t1         Skip if staccato required
6f1          Force l.s. digit to 0
62p67
33f200
55R2
58p50(107     Report if > 2047  (note too long)
35f19         Plant in 19 for storage in main store
32f202
54R2
58p50(108     Report if < 2  (note too short)
10f16
50p19         Store completed note in main store
```

```
76t1
50R6
10f18     }
55R2      }
2f256     }  If staccato, change modifier of 18 to form
76t6      }    rest and plant in main store
65t18     }
10f18     }
50p19     }
5f18         Clear 18 ready for next note
60S33     }
75Sp47    }  Jump according to terminator
50p44     }
_______

15f26(45     Preserve M (= m)      Read no. of dots for note of music
46t-26
52R2
58p50(113    Report if terminator of m was not .
58p30        Read no. of dots
58p50(114    Report if -ve or wrong terminator
55R-1
6f26         Restore m
60t25        t = no. of dots
50p42
_______

9f32(44                            Beginning of new line
46f1      } Increment line count
36f2      }
58p6(5    }
77t30     }
50p5      }
76t1      }
50p9      }  Read characters ignoring bl cr lf fs ls sp
46t-2     }
52p5      }
46t-6     }
52p5      }
50f2k47
_______

10f30(67  <== "+"                   ~~[illegible]~~  Pause facility
53p61        Jump if in letter-shift
46f1      }  Add 1 to l.s. address of last note
36f30     }
50p6      }
46t-2     }
53R3      }
75t33     }
50p44     }  Read terminator and jumps accordingly
46t-30    }
53R3      }
75t33     }
59p47     }
58p50(109    Report if next character not cr or sp sp
```

"TEMPO"

```
10f30(86     Jump if "t" read in figure-shift
52p61 ·
7088       }
58p69      }
33p66      }  Read next 4 characters; report if not E M P O
52R2       }
58p50(110  }
58p30        Read integer (basic time-value, m)
50R2
58p50(111    Report if followed by cr or sp sp
19f16
46t-14
53p41        Jump if terminator not =
9f10
58p30(43     Read integer (metronome rate, n)
58p50(112    Report if -ve or wrongly terminated
55R-1
19f12      }
80t10      }
6t1        }  n(2^(t+1) - 1)        (t = no. of dots)
33f12      }
39f12      }
10f16      }
6t0        }  m2^t
34f204       Convert minutes to μsecs
35f12      }  Calculate T (μsecs/semibreve) and store in 10 as current tempo
19f10      }
60833      }
758p47     }  Jump according to terminator
50p44      }
```

Read no. of dots for tempo setting.

```
46t-26(41
52R2
50p113       Report if terminator of m not ·
58p30        Read no. of dots
50R2
50p114       Report if -ve or terminator not =
19f10        N(10) = no. of dots
55R-2
46t-14
53R-4
50p43
```

"Staccato

```
10f30(85  ⟸  "·"
53p61        Jump if letter-shift set
75819        Mark staccato by non-zero a(19)
50p46
```

"OCTAVE"

```
10f30(84
52p61      Jump if "O" read in figure-shift
70810    }
58p69    }
33p71    } Read next 5 characters ; report if not C T A V E
52R2     }
58p50(117 }
58p30      Read octave no.
58p50(116  Report if not terminated by cr or sp sp
19f4       Store in 4 as current octave no.
60S33    }
75Sp47   } Jump according to terminator
50p44    }
```

"KEY"

```
10f30(83
52p61      Jump if figure-shift.
7084     }
58p69    }
33p72    } Read next 2 characters ; report if not E Y
52R2     }
58p50(119 }
58p30      Read integer (no. of flats or sharps)
54R2
58p50(120  Report if wrong terminator or -ve or too large
9f26       Transposing constant = 0 provisionally.
6f1
19f6
33f206
54R-5
77t0     }
50R-7    } Test terminator ; t = 17 if s (sharp key), 1 if f (flat key)
77t2     }
70t17    }
4f9      }
10t-1P70 } Fetch mask for specified key and store in 8
19f8     }
58p6       Read next character
40t-11
52p39      Jump if / or A
40t-30
52p47      Jump if sp sp
40t-2
52p44      Jump if cr
58p50(121  Report otherwise
10f30(35     (After / or A)
53p121       Jumps if in letter-shift
50p30        Read integer
58p50(122    Report if wrong terminator or no. too large
0f1
15f26        Store transposing constant (= 2 × no. of semitones transposed)
40f26      }
33f200     } Check maximum size (± 12 semitones)
54R-5      }
00833      }
758p47     } Jumps according to terminator
50p44      }
_________

58p50(81     Report on illegal gaps character
_________

                                                   Begin repeat
10f30(80  ⟸  "("
53p01        Jumps if in letter-shift
50p6         Read next character
46t-30
52p37        Jumps if sp
46t-2
52p37        Jumps if cr
58p25        Otherwise read integer
58p50(123    Report if wrong terminator or if integer < 2
33f210
55R-2
50R2

79t33(37     Mark end of item on sp or cr
32f212       N(M) = no. of copies required (i.e. 1 less than no. of times to be played)
60t20      }
72t2       } t = next available register in repeat list (1900 if list empty)
76t1900    }
70t1900    }
100t0      }
59t1       } Store in list (i) b+1 = b₀
10f214     }              (ii) t
30t0       }  and reset list pointer in a(20)
79t20      }
00833      }
758p47     } Jumps according to terminator
50p44      }
_________

                                                   End repeat
10f30(25  ⟸  ")"
53p01        Jumps if in letter-shift
50p6         Read next character
46t-30     }
52R3       } ~~Jumps if~~
46t-2      } if cr or sp sp. mark end of item
53R3       }
79t33      }
50p36      }
58p25        Otherwise read integer, k.
58p50(124    Report if wrong terminator or k < 0
55R-1
60t20(36     t = repeat list pointer
76t1900
58p50(125    Report if list empty
7182        } Reset list pointer
75820       }
2t1         } s = no. of copies required
8080        }
0f20
15f12         k 2^20
51R1        } Clear l.s. half to allow arithmetic on main store addresses
55t1        }
100f14 .      ~~[illegible]~~
10t0
33f14
33f214        b0 - b - 1
15ff0         Store in 0 as indirect address
32f12         b0 - b - 1 + k = -(no. of notes in repeat)
55R2
50p50(92      Report if +ve or zero
15f14
80t14(35      t = -(no. of notes in repeat).
                  Copying Cycle
112f*1      }
10R*0       }
58f1p15     }
72t1        }
77t1        }  Copy notes in main store until t=0 ;
50R-5       }  then adjust indirect address, reset t, and repeat ;
10f14       }  continue until s=0
38ff0       }
7381        }
7781        }
50p35       }
80833       } Jump according to terminator
758p47      }
50p44       }
-----
                                              Rest
10f30(20  <== "R"
52p01         Jump if in figure-shift
10f216      }
15f16       }
33f22       }
34f24       }  Plant address in 18 as if for note of 100 c.p.s.
35f12       }
80813       }
75816       }
02p05       }
53R2        }
2f256       }  Plant modifier according to 32-digit (opposite to that
7080        }                                   used for a note)
05816       }
50p34          Jump to read time-value
-----
112f*1(15 <==       Subroutine to write N(M) into next available
15f*0                 main store register.
100f12
10f12
55R2
50p50(120         Report if b=0 i.e. end of store reached
80f0  ==>
-----
```

Jump table

address parts : for A, B, D, E, F and G : pointer for the corresponding note in the table of inverse frequencies

for other characters : jump address for gap characters

function parts : (digits 0 – 2) relative jump on character following A, B, C, D, E, F or G

0tp40(02	bl	bl
0s16	F	f
0fp44	cr	cr
0tp64	O	0
0tp26	R	r
0tp63	K	7
0tp61	U	2
04Rf66	S	s
0tp61	lf	lf
0tp61	L	8
0tp61	H	5
0t16	A	/
0tp61	M	9
0tp60	Z	(
0t6	E	=
0t4	D	10
0tp61	P	p
0t14	G	4
0tp61	Y	3
0tp61	W	2
0tp61	J	6
0tp66	T	t
04sp61	N	n
0t22	B	–
0tp61	I	1
0tp40	C	*
0tp25	X	)
0f0	ls	ls
0tp65	V	'
0tp67	Q	+
0fp46	sp	sp

Table of inverse frequencies

Covers 3 octaves : basic octave (Octave 0) and 1 octave each side to allow for transposition

n6055	B	Octave –2
7644	C	Octave –1
7216	C#	
6616	D	
6426	D#	
6067	E	
5727	F	
5406	F#	
5102	G	
4616	G#	
4545	A	
4255	A#	
4050	B	

3822(65	C	Octave 0
3606	C#	
3405	D	
3214	D#	
3034	E	
2863	F	
2703	F#	
2551	G	
2408	G#	
2273	A	
2145	A#	
2025	B	
1911	C	Octave 1
1804	C#	
1703	D	
1607	D#	
1517	E	
1432	F	
1351	F#	
1276	G	
1204	G#	
1136	A	
1073	A#	
1012	B	
956	C	Octave 2

	Character Groups and Masks, etc.
$n25_{2}15(60$ $FF5_{2}10$ $FF14_{2}5$ $FF2$	C L E cr
20(61	Difference between s/p and cr
0f1502(63 0f400	Initial and final pause for main store
$21_{2}15(64$ $FF3_{2}10$ $FF16_{2}5$ $FF2$	T O P cr
0f0(65 0f32	Mask for 32-digit
127+2047(67 127+2046	Mask to force l.s. digit to zero
$14_{2}15(66$ $FF12_{2}10$ $FF16_{2}5$ $FF3$	E M P O
$25_{2}20(71$ $FF21_{2}15$ $FF11_{2}10$ $FF20_{2}5$ $FF14$	C T A V E
$14_{2}5(72$ $FF10$	E Y

Key Masks

Digits 0-27 represent chromatic scale C flat up to B sharp, 2 digits for each note. The digits representing each of the notes C, D, E, F, G, A and B, are:

00	if note unchanged
11	if note flattened
10	if note sharpened.

	Key:
0(70	0f
0f0 24f0	1f
0f768 24f0	2f
0f768 24t0	3f
6f768 24t0	4f
6f780 24t0	5f
102f780 24t0	6f
102f972 24t0	7f
0	0s
0f128 0f0	1s
64f128 0f0	2s
64f136 0f0	3s
68f136 0f0	4s
68f136 0s0	5s
68f648 0s0	6s
68f648 16s0	7s

$n1_2$-23(52 Mask for digit 23 (used when measuring machine-speed)

p1=200

Constants

n2048

2046

$6_{10}7$

15

25

2

1

nn1

$n1_{10}4$

52_2-20

0.35

50

970

$1_{10}8$

p1=2041

50f-1p99 Jump to start of music on interrupt

s301

****