DECUS
PROGRAM LIBRARY

DECUS NO.	8-403
TITLE	STEREO-A 2 CHANNEL MUSIC PROGRAM
AUTHOR	Maurice Retter
COMPANY	University of Oxford Oxford, England
DATE	March 1969
SOURCE LANGUAGE	

STEREO - A 2 CHANNEL MUSIC PROGRAM

DECUS Program Library Write-up

DECUS NO. 8-403

ABSTRACT

A musical program has been written for the PDP8 which can control two loudspeakers independently. A frequency is produced by creating a square wave pulse train, where each pulse is generated by an IOT instruction, and the time delay between pulses is under program control. Two channels are made available, if required, by using two IOP pulses from one IOT instruction to activate independent loudspeakers. The program is divided into a coding section, and a decoding section and play routine. This paper describes how the program works, how it is operated, and its limitations. The trivial changes necessary in the hardware are described and an example is given in appendix II.

1 Introduction

The musical program STEREO is divided into three sections. There is a coding section which enables the operator to type in notes in a familiar notation, and this language is turned into coding which is stored in core. This notation caters for single notes, mixed notes, time intervals and DO-LOOPS etc., within the coding. The internal coding produced can then be decoded by the DECODE routine, which unpacks the bits from a twelve bit word and sets up appropriate addresses and flags which determine what note should be played, for how long, and how it should be played. The play section then executes the various NOTE, MIXED NOTE, and PAUSE states, necessary for the synthesis of musical tunes.

The coding consists of some control words, but mainly of six bit address bytes which point to the lookup table of notes. This table contains constants which when used in the play routine bring about the appropriate delay between IOT instructions, and how many IOT instructions are executed. The scale of notes used is the diatonic scale, and the frequency of 'Middle C' is chosen as 256 c/s.

The program is written in PALIII and all timing control and constants are based on a 1.5 micro second cycle time of the CPU. The interrupt is not used, and the only requirements for the system are a teletype, 4K PDP8, and suitable space for some trivial interface modifications.

The necessary interface modifications involve the introduction of a 1-shot package (R302) and other cards to invert and gate the appropriate IOP pulses into the 1-shot. Existing circuitry used for a 50 c/s clock acting through the interrupt, was used to derive the device and IOP selection logic, but the 1-shot was necessary to turn the IOP derived pulse into 'audio' dimensions.

STEREO was developed on an 8K PDP8 which is integrated into a measuring system in which operators measure parameters from film under ON-LINE control. The film contains records of high energy nuclear particle collisions, and the feedback of errors etc., is by teletype only. Other forms of feedback might include audio and visual guidance, and the audio channel was built partly to investigate this means of control. Its use is limited to the general monitoring of the ON-LINE program, and the modes of use of the floating point package contained in the program. Apart from its limited use as a debugging aid,

its use with STEREO was of value as a distraction to the author, and as entertainment to the scanners. It may have value in the future with this particular installation, as a means of remote monitoring, when the PDP8 becomes part of a multi (5-way) processor complex.

2 General description of the STEREO program

A simple flow diagram for the program is given in figure 1, together with a summary of the layout of the coding. After loading and starting the program, the program branches to one of the two modes, the coding or playing mode, depending on whether switch 1 is 1 or 0 respectively. The program branches when the switch register is negative (i.e. when switch 0 is 1). The starting address is 200(8).

The coding section accepts commands from the teletype, and turns the symbolic code for notes, timing intervals etc., into internal code, and assembles this code sequencially in memory, starting at location 1000(8), or any other location if specified in the keys when the program is started. Coding can be created or edited in this way.

The decoding section retrieves the data in core starting at location 1000(8) and interprets this code as commands or as data, and computes the necessary delay constants, or flags, before passing control to the PLAY routine for execution of a note, notes or pauses. One command, or note, or two notes together is decoded at a time before PLAY is called.

The PLAY routine is called after the appropriate flags and constants have been set up by the decoding section. Several states are possible. For example a note is derived as a square wave pulse train for a specific time interval, or a pause state is entered which lasts for a given time interval. A note may also be followed by a pause state, where the time interval for the note, and the pause, are the same.

A frequency lookup table is provided which contains two constants for each frequency provided. The first is used as a delay to provide the correct time interval between two IOT pulses, and the second is the number of pulses that would occur in a two second time interval. This is the maximum time for which a note can run, and so for greater times the note must be sent again, and for shorter times the constant is automatically divided by 2, 4, 8 etc.

The internal coding is stored in location 1000(8) onwards and can extend up to location 4777(8). It is decoded sequencially except for commands called REPEAT commands, such as pseudo DO LOOPS, SUBROUTINES, which bring about the necessary branching to repeat sections of the coding one or more times; whole tunes can be repeated or two notes, or even one note repeated to a maximum of 77(8) times. Coding once produced and debugged can be saved with ODT in the normal way, but it remains absolute and cannot be relocated subsequently.

3 Hardware

The hardware changes needed at the interface were trivial. Any device code is acceptable provided it does not conflict with the other device codes used in the program, or the specific action taken by any device. In this case device 05 was used with IOPS 2 and 4, which correspond to no action, and clear the clock flag respectively.

Two IOT instructions were used to create pulses derived directly from the IOP pulses. These instructions are 6052 and 6054 so that IOP2, IOP4 or IOP2 and IOP4 could be sent with the one IOT instruction.

These pulses are lengthened to 450 micro seconds via a 1-SHOT (one for each IOP) and so two separate channels are catered for with one R302 board. Each output is taken to a simple one stage amplifier which feeds a low impedance three ohm loudspeaker. The loudspeaker is thus driven by a square wave pulse train, the derivation of which, and repetition rate between IOT instructionsbeing under program control. The volume is controlled manually, but depending on the characteristics of each amplifier and loudspeaker the high and low frequency notes may be program assigned to given channels.

4 The coding structure

A summary of the meaning of all the letters used to write a tune in the necessary coding is given in appendix 1. A buffer 20 characters long is allowed for the teletype and is sufficient for one string of music code, which may be followed by a label. Rubbout causes the previous character to be deleted. Carriage return is followed by a line feed and causes the buffer to be examined and turned into internal code. This code is placed in the area 1000(8) onwards, using an index pointer (INDEX). If the coding is correct then the location of the one (or

more) 12 bit words is typed back to the teletype together with its contents in octal, so that the internal coding can be seen.

Examples of some input strings are given below to illustrate how a line of external coding may be written.

NT3,G2,	Single Note G2 for $\frac{1}{4}$ second
NT1,C3#,	Single Note C3# for 1 second

(Note that there is no representation for a flat, so that to get semitones, or to represent the 'black' keys on a piano keyboard, the number sign must be used. F1= E1#)

NT4I,D2,	Single note D2 for $\frac{1}{8}$ second followed by $\frac{1}{8}$ second pause.

(note that the I, which is optional, if it appears must follow the timing argument 4).

MT3I,E2,C2#,	Two notes are played together (E2 and C2#) and a $\frac{1}{4}$ second silence interval follows the mixed note interval (of $\frac{1}{4}$ second).

(the higher note should come first, although in some cases it may be better to invert the two notes, if these are to appear on one loudspeaker, to overcome unpleasant beat frequencies. This effect is due to the quantised nature of the timing loops, and also to the fact that the note duration is determined by the first frequency only. See next sections on frequency and timing).

MT1,B3,G3,	Two notes for 1 second
PT0,	Pause of 2 seconds
R1000,03,	Repeat the coding from location 1000 up to this repeat command three times, then proceed.

(It is always best to have a single line of code at the address used for a repeat argument. If a mixed note, which occupies two locations, is to be the branch point, then the address of the first line of coding should be given. The absolute address must be specified but for convenience a label may be typed after a given string for later reference. The label is ignored by the processor which scans the input string).

R1245←1266,01, — Repeat the coding from location 1245 to 1265 once, then continue from this repeat command.

(The second address should be the first line of the coding not to be executed, at the end of the range).

X1,2 — The higher (first) frequency is to go to loudspeaker 1, and the second (lower) frequency (if present by virtue of a Mixed note) is to go to loudspeaker 2.

X3,0 — Frequency A of a note or mixed note is to go to loudspeakers 1 and 2, but frequency B if present as a mixed note is suppressed.

(This facility allows a block of coding to be used in different ways. A given section can be played as a single note rhythm, then as a mixed tune on one speaker, and then again separating the frequencies playing each on separate speakers).

5 Frequency generation

A frequency is generated by executing an IOT instruction with the appropriate periodicity. The pulse separation or period varies from 1/64 second (C0), up to 1/2048 second (C5). Considering the single channel, single note mode, the pulse train is generated by the following simple program loop P:

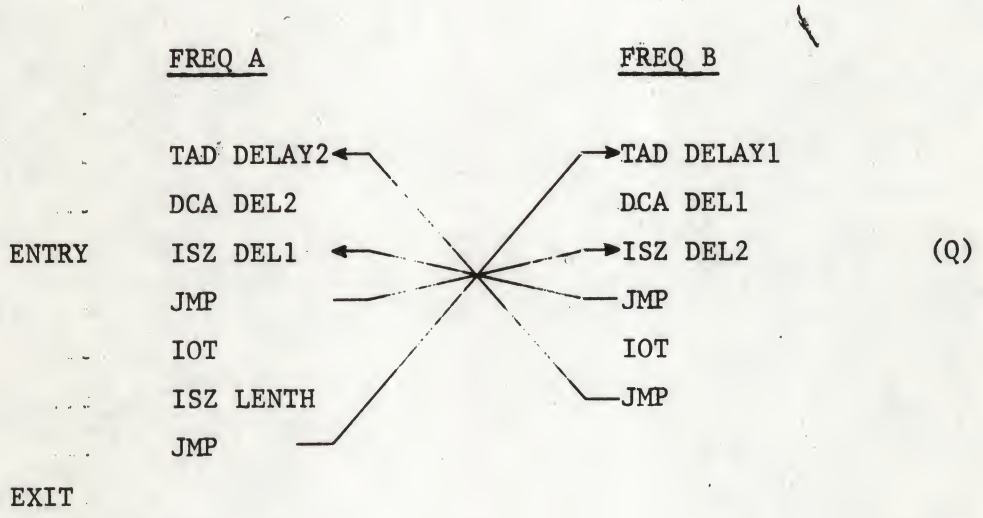
FREQ A ONLY

```
TAD TIME
DCA LENTH
TAD DELAY  <──────┐
DCA WAIT          │
ISZ WAIT  <─┐     │
JMP .-1  ───┘     │        (P)
IOT               │
ISZ LENTH         │
JMP .-6  ─────────┘
RETURN
```

Here TIME and DELAY, two parameters which define the length and frequency of a note, have been previously computed in the routine DECODE. The coded data is fetched, decoded, and then the play routine called as follows:

```
        CALL  DECODE ←┐
        CALL  PLAY    │
        JMP   .-2     ┘
```

To enable two channels to be activated or two frequencies to appear on one channel, the coding becomes a little more subtle. In principle the following scheme is adopted: Q:



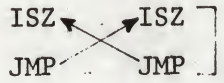
The IOT instructions may be 6052, 6054 or 6056, and depending on which IOP pulses are present or absent one obtains a 'pause' state (or silence state) or frequencies A only, or A or B, or both on either or both channels. The IOT instructions can be replaced by special reassignment pseudo-ops decoded in DECODE, so enabling either channel to be selected for frequencies A, B, A and B or neither, independently, and independently of whether a note B exists. (i.e. whether it is a single (N) or Mixed Note (M)). A and B may be present or absent or masked by the appropriate IOT exchange.

Three separate loops occur in subroutine PLAY, and these correspond to one of the following:

a) If (-) then this is a pause state
b) If (A) then this is a note state
c) If (A&B) then this is a mixed note state.

The first is handled by a P loop, but with a dummy IOT instruction (e.g. KCC) which must not bring about any SKIP action etc. The second is handled by a loop equivalent to P but is in fact part of the Q loop, and so is subject to a previous channel assignment, command X. Frequency A can therefore be channelled to loudspeaker 1 or 2 or 1 & 2 (Not 1 or 2 (6050) is equivalent to a pause state). (C) is handled by a loop of the Q type, and within each interlaced subloop, the IOT's may be changed to determine on which channels that frequency is to appear. This loop has extra coding to try to offset inaccuracy of timing due to added cycles (see below), and due to phase changes of 2π between different frequencies.

ISZ
JMP — 3 cycles



6 cycles

6 Timing the IOT instructions.

The notes for the scale C1 to C2 were taken as follows:

C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C (octave)
1	$\frac{135}{128}$	9/8	6/5	5/4	4/3	$\frac{45}{32}$	3/2	8/5	5/3	16/9	15/8	2

The timing for the various loops described in 5 is based upon the loops as they are written

$[(11\frac{1}{2} + (N-1)3)1.5 = T]$ NORMAL

$[(11\frac{1}{2} + (N-1)6)1.5 = T_H]$ MIX LOOP A

$[(36 + (N-1)6)1.5 = T_L]$ MIX LOOP B

Where T is the periodic time

So $N = \frac{T}{1.5x3} - 3$ NORMAL

$N = \frac{1}{2}\frac{T}{1.5x3} - 1$ High frequency Mixed Loop $N = \frac{1}{2}\frac{T}{1.5x3} - 5$ Low frequency Mixed Loop

The frequency table contains the value $\frac{T}{1.5x3}$

and depending in which loop the frequency is to be derived the appropriate N is computed.

As an example of these equations the count for middle C (f = 256) is 1544(8). That is, the periodic time is: 3906μsecs., so $\frac{T}{(1.5x3)}$ = 868 = 1544(8). Also the number of IOT instructions needed for a 2 second period is 512 = 1000(8).

7 Limitations

There are several improvements, and limitations to the system. The basic limitation to the system is the restriction of the coding to 12 bits. In order to compress one note into one location, or two mixed tones into two locations the timing has been restricted to units of 2, 1, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, $\frac{1}{64}$ seconds. This could be expanded by introducing some pseudo-ops which would change the unit of one second to $\frac{3}{5}$ of a second, or $\frac{1}{3}$ of a second. However, the timing has proved adequate. The ability to slide between frequencies A and B in a given time interval has not been included, although it could be included fairly easily provided $\Delta f/f$ was small, and another type of loop was incorporated in the PLAY routine.

If an error occurs in the coding of an input string, then this is indicated in most cases by absence of typeout showing the inserted code. Some improvements could be made here to indicate the error more precisely. Note that if errors occur where more than one line of internal coding is produced, the index pointer may not be left in the correct position, so that a X1215 (for example) may be necessary.

The logic to offset small changes in the timing due to mixing widely different frequencies was not fully explored and some improvements could be made. The combinations of frequencies A and B on 1 speaker can lead to discord, if the two frequencies A and B are too far apart, or/and they lead to a low beat level (e.g. C2, C3). This effect was found to be less severe if the frequencies were channelled to different loudspeakers.

8 Conclusions

The musical program STEREO was conceived, written and hardware modifications made, tunes written etc., within a very short interval of time, and so the system may lack the professional finish. Improvements in error correction, or flexibility of coding could be made, and the program warrants a closer inspection of the timing perturbations and associated discords, brought about by mixing two very different frequencies. An investigation to this end would prove interesting and rewarding but there is no justification for such work with this installation.

STEREO proved very entertaining as a source of Christmas carols etc., and is presented as a working system. One tune is given in Appendix 2. It also represents an entirely empirical approach to the generation of sounds by a computer, and may be of use as a comparison with other efforts in this field.

Appendix 1

The use of the letters A-Z

A-G These letters stand for notes. The number following one of these letters specifies which octave is used. There are two octaves below and three above middle C which is C2 i.e. the scale goes:
C0, C0#, D0, D0#, E0, F0, F0#--------
---A1#, B1, C2, C2#, ------A4#, B4, C5
(the number sign (#) may be used with any note to raise the note one semitone).

H When typed in is turned into code 0000 which brings about a pseudo HALT, i.e. the tune or tunes will be played until a zero appears, whereupon the program halts. This is useful as a terminator when debugging sections of the coding. Depressing the CONTINUE key allows playing to continue.

I When I appears in the specification of a note or mixed note, then an equal time interval or pause follows the note or mixed note time interval. It can be used to separate notes, in time, or to emphasize a staccato rhythm.

J J followed by a four octal digit number (e.g. J1026), causes a command word 0077 to be inserted in the coding, and the number 1026 to be inserted in the list of addresses starting at 6000. When this command is executed, the decoding and playing routines branch to the coding starting at the next address retrieved from this address list at 6000. This has not proved a great advantage. A means is provided, for entering subsequent addresses into the list without entering further commands into the coding. This is carried out by the following: JXXXX←,. This adds XXXX to the address list only.

M This letter implies that a mixed note is required. It indicates to the coding compiler that two frequencies are to be expected.

(Examples for M, N and P show the coding that is produced:)

NT5[I],G2#, → (00[1])(5)(-)(-)

Address of G2# note

MT3[I],G2#,C3, → (10[L])(3)(-)(-)

G2# Address

(0)(0)(-)(-)

C3 address

PT5, → (0)(5)(0)(0)

The numbers in parentheses are octal numbers, and the letter I and the appropriate switch bit shown in square brackets is optional. That is the I can be present or absent and if present causes a pause interval to follow the note for the same duration as the note.

Options are:

INTERVAL (no note)
TONE (no interval)
TONE + INTERVAL
MIXED TONES (no interval)
MIXED TONES + INTERVAL

The note address is computed as follows:
noteGN#

1) G is turned into a number V
2) then (V + N x 12) is computed. (12 notes/octave)
3) If # present add one
4) Multiply by 2 since there are two constants/note.

P This means a pause when neither loudspeaker is activated, and the only argument is a time interval.

R Indicates a repeat command. There are three types to allow for resting, and these have proved adequate for the tunes so far written. No one repeat may be rested but each repeat may appear within another. The three types are written:

RXXXX,03, which codes into 6003
XXXX
RXXXX,02←, codes into 7002
XXXX

RXXXX←YYYY,01, codes into 7402
XXXX
YYYY

The last two octal numbers are stored in the right hand six bits of the first word and represents the number of times the coding is to be repeated, and in the first two cases the coding between XXXX and the repeat command is executed. In the third case the coding between XXXX and YYYY is executed, and this must exclude the command itself, i.e. it is equivalent to a 'JMS' called a number of times.

s S is not permissable. The intention was to slide between frequency A and B, but this is not programmed. It causes the program to halt.

T This stands for timing and must precede a number between 0 and 7. The timing units are binary divisions of the two second interval. So 0 means 2 seconds, 1 = 1 second, 2 = ½ second, 3 = ¼ second, 4 = ⅛ second etc. This restriction on the timing is a limitation, but has proved acceptable for those tunes written so far. It was adopted to save coding space. In this way a note and its associated time interval can be specified in 12 bits. Any other time interval can be achieved by repeating the note, so that the sum of the time intervals is the required one.

X This letter precedes certain specific commands as follows:

XX means proceed to start, and play the coding. The index pointer which points to the coding is automatically set to 1000, and the music is played until a Halt or 0000 is found in the coding.
X1234 will reset the index pointer to 1234 so that the next line typed in will overwrite line 1234, and subsequent lines if more is typed.
X0, 3, X3, 3 etc. This specifies to what channels the frequencies are sent. For example, X1,2 means that frequency A will to go loudspeaker 1, and frequency B to loudspeaker 2; X3,0 means that frequency A of a mixed note will appear on both speakers, frequency B if present is suppressed. A or B or speaker 1 or 2 can be masked off in this way. The arguments can be 0, 1, 2, 3 and are directly equivalent to IOP 0, 2, 4, 2 & 4, generation.

Z Z is coded as 7777, and is a reset command, so that branching takes place to 1000, but playing continues. The index pointer is reset to 1000 and the tune or tunes are replayed over and over again. It should not appear within any kind of repeat, of course, but can be used as a terminator to repeat the tunes.

, A comma is an important delimiter in the combination of various notes, and should always occur at the end of an input string of coding.

(All other letters K, L, O, Q, U, V, W, Y are not used, and will cause an input string to be ignored).

Appendix 2

This appendix gives the coding, in full, of a tune written for the stereo program, and plays on both loudspeakers most of the time. In PART B is shown the additional program logic to produce random music on one loudspeaker.

PART A

CODING for the tune 'ONE, TWO, THREE O'LEARY' by BARRY MASON and MICHAEL CARR.

The left-justified lines beginning with P, X, M etc., are the only lines to be typed in at the OTY teletype.

```
PT1,
  1000/    0100
X1,0
  1001/    2004
MT2,C3,C1,          AA
  1002/    4245
  1003/    0015
MT2,C3,C1,
  1004/    4245
  1005/    0015
MT3,G3,C1,
  1006/    4354
  1007/    0015
NT3,E3,
  1010/    0351
NT3,D3,
  1011/    0347
MT3,C3,GO,
  1012/    4345
  1013/    0010
MT3,B2,G1,
  1014/    4344
  1015/    0024
```

```
MT3,B2,F1,
  1016/    4344
  1017/    0022
MT3,B2,E1,
  1020/    4344
  1021/    0021
MT3,B2,D1,
  1022/    4344
  1023/    0017
MT3,G3,C0,
  1024/    4354
  1025/    0001
MT3,E3,E0,
  1026/    4351
  1027/    0005
MT3,C3,G0,
  1030/    4345
  1031/    0010
NT3,B2,
  1032/    0344
MT2,A2,F0,
  1033/    4242
  1034/    0006
MT2,C3,A0,
  1035/    4245
  1036/    0012
MT3,G3,C1,
  1037/    4354
  1040/    0015
MT3,F3,C1,
  1041/    4352
  1042/    0015
MT3,C3,F1,
  1043/    4345
  1044/    0022
NT3,A2,
  1045/    0342
PT7,
  1046/    0700
```

```
MT2,G2 ,C2,
  1047/    4241
  1050/    0031
MT2,C3,G2 ,
  1051/    4245
  1052/    0041
MT3,G3,E2,
  1053/    4354
  1054/    0035
MT3,F3,D2,
  1055/    4352
  1056/    0033
MT3,E3,E2,
  1057/    4351
  1060/    0035
MT3,D3,F2,
  1061/    4347
  1062/    0036
X1,2
  1063/    2006
R1002,01,  BACK TO AA
  1064/    6001
  1065/    1002
PT7,                DD
  1066/    0700
MT2,E3,C2,
  1067/    4251
  1070/    0031
MT1,E3,C2,
  1071/    4151
  1072/    0031
NT3,F3,
  1073/    0352
NT3,E3,
  1074/    0351
MT2,D3,A2,
  1075/    4247
  1076/    0042
```

```
MT2,F3,C3,
  1077/    4252
  1100/    0045
MT3,A3,F2,
  1101/    4356
  1102/    0036
MT3,F3,C2,
  1103/    4352
  1104/    0031
MT3,D3,A1,
  1105/    4347
  1106/    0026
MT3,B2,D1,
  1107/    4344
  1110/    0017
MT0,C3,C1,
  1111/    4045
  1112/    0015
PT7,                BB
  1113/    0700
MT4,C4,C1,
  1114/    4461
  1115/    0015
NT4,C1,
  1116/    0415
MT4,A3,F1,
  1117/    4456
  1120/    0022
NT4,F1,
  1121/    0422
MT4,F3,A1,
  1122/    4452
  1123/    0026
NT4,A1,
  1124/    0426
MT4,A3,A1,
  1125/    4456
  1126/    0026
```

```
MT4,C4,A1,
  1127/    4461
  1130/    0026
MT3,B3,G1,
  1131/    4360
  1132/    0024
MT3,G3,E1,
  1133/    4354
  1134/    0021
MT3,E3,G1,
  1135/    4351
  1136/    0024
MT4,G3,C2,
  1137/    4454
  1140/    0031
MT4,B3,E2,
  1141/    4460
  1142/    0035
MT4,A3,D2,
  1143/    4456
  1144/    0033
NT4,A3,
  1145/    0456
MT4,F3,D2,
  1146/    4452
  1147/    0033
MT4,F3,E2,
  1150/    4452
  1151/    0035
MT3,D3,F2,
  1152/    4347
  1153/    0036
NT4,F3,
  1154/    0452
NT4,C4,
  1155/    0461
X2,1                CC
1156/      2011
```

```
MT3,B3,F2,
  1157/    4360
  1160/    0036
MT3,G3,C3,
  1161/    4354
  1162/    0045
MT3,B3,F3,
  1163/    4360
  1164/    0052
NT4,G4,
  1165/    0470
MT4,G4,C4,
  1166/    4470
  1167/    0061
MT3,F4,A3,
  1170/    4366
  1171/    0056
X1,2
  1172/    2006
MT4,F4,A3,
  1173/    4466
  1174/    0056
MT4,E4,G2
  1175/    4465
  1176/    0040
MT2,E3,G2,
  1177/    4251
  1200/    0040
MT2,E3,G2,
  1201/    4151
  1202/    0040
PT7,
  1203/    0700
  X1,2
  1204/    2006
R1113←1156,01,        BB TO CC
  1205/    7401
  1206/    1113
  1207/    1156
```

```
MT3,D4,G3,
  1210/    4363
  1211/    0054
MT3,D4,G3,
  1210/    4363
  1211/    0054
MT3,D4,G3,
  1212/    4363
  1213/    0054
MT4,D4,G3,
  1214/    4463
  1215/    0054
X2,1
  1216/    2011
NT4,C4,
  1217/    0461
NT4,E4,
  1220/    0465
NT4,G4,
  1221/    0470
MT3,F4,D3,
  1222/    4366
  1223/    0047
MT4,F4,D3,
  1224/    4466
  1224/    0047
MT4,E4,C3,
  1226/    4465
  1227/    0045
MT1,E4,C1,
  1230/    4165
  1231/    0015
X1,2
  1232/    2006
NT6,G3,
  1233/    0654
NT6,A3,
  1234/    0656
NT6,G3,
  1235/    0654
```

```
PT6,
  1236/    0600
NT6,F3,
  1237/    0652
NT6,G3,
  1240/    0654
NT6,F3,
  1241/    0652
PT6,
  1242/    0600
NT6,E3,
  1243/    0651
NT6,F3,
  1244/    0652
NT6,E3,
  1245/    0651
PT6,
  1246/    0600
NT6,D3,
  1247/    0647
NT6,E3,
  1250/    0651
NT6,D3,
  1251/    0647
PT6,
  1252/    0600
R1002←1066,01,        AA,DD
  1253/    7401
  1254/    1002
  1255/    1066
MT2,E3,C2,
  1256/    4251
  1257/    0031
MT1,E3,C2,
  1260/    4151
  1261/    0031
NT3,F3,
  1262/    0352
```

```
NT3,E3,
  1263/    0351
MT2,D3,A2,
  1264/    4247
  1265/    0042
MT2,F3,D2,
  1266/    4252
  1267/    0033
MT3,A3,C2,
  1270/    4356
  1271/    0031
MT3,F3,D3,
  1272/    4352
  1273/    0047
X3,3
  1274/    2017
MT3,D3,F2,
  1275/    4347
  1776/    0036
MT3,B2,D2,
  1277/    4344
  1300/    0033
MTO,C3,E2,          END
  1301/    4045
  1302/    0035
H,
  1303/    0000
```

PART B

To produce random music it is necessary to introduce the following logic.

Overwrite one instruction of the STEREO program as follows:

CHANGE TAD I INDEX in SUBROUTINE GET TO JMS I JENRAN 442/1410→4777

Add 577 / → JENRAN, GENRAN

At 6100

```
GENRAN, 0                   0
        CLA                 7200
        TAD RANUM           1340
        RAL CLL             7104
        SZL                 7430
        TAD P3              1341
        DCA RANUM           3340
        TAD RAMUM           1340
        AND TSTZO           0342
        SZA                 7440
        JMP .+4             5316
        TAD FOUR            1343
        TAD RANUM           1340
        JMP DOWN            5320
        CLA                 7200
        TAD RANUM           1340
DOWN,   OSR                 7404
        AND P777            0344
        DCA REMEM           3345
        TAD REMEM           1345
        AND P77             0346
        TAD M70             1347
        SMA CLA             7700
        JMP GENRAN+1        5301
        TAD REMEM           1345
        JMPI GENRAN         5700
```

Add constants

At 6140

```
RANUM,2371          REMEM,0
P3,3                P77,77
TSTZO,177           M70,7710
FOUR,4
P777,777
```

The OSR instruction enables the randomly generated note to be combined with the switch setting. This means that a minimum frequency, or note, can be chosen which is above the lowest permissible note, and the minimum time for which a note can run can be set in this way. That is if (O)(T)(P)(Q) is set in the keys (in octal), then T is the minimum timing, and PQ is the minimum note address. (See Appendix I for further explanation of this format.)

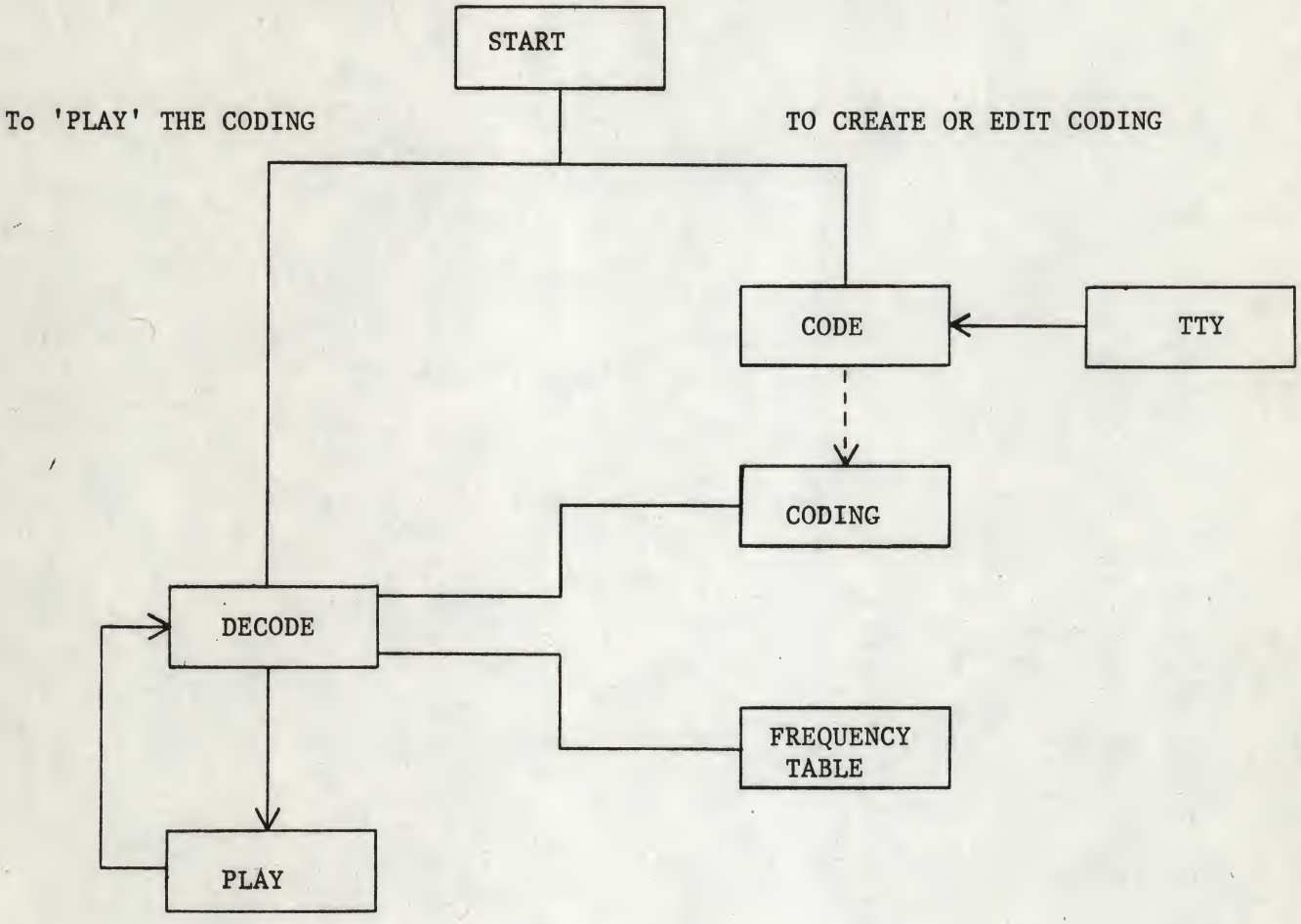


FIGURE 1

The simple flow diagram of the STEREO program. The program runs in 4K and the layout of the core is as follows:

PAGE 0	CONSTANTS ETC. and SPACE FOR COMMAND STRING
PAGE 200	START AND MAIN ROUTINE
PAGE 400	DECODE
PAGE 600	PLAY
1000-4777	SPACE FOR CODING
5000	LOOK UP TABLE
5200 6200	CODE
6000	SPACE FOR JUMP address