FRED

SOUND

SYNTHESISER

Users Instructions.

Lowe Electronics Ltd.
August 1981.

FRED Sound Synthesiser.

CONTENTS.

==========

A knowledge of BASIC programming and program editing is assumed in this manual. If a disk system is used then knowledge of the disk operating system is also assumed.

1. Connection to your system.

The FRED sound synthesiser plugs directly onto the expansion connector of the Video Genie. The 50-way plug should be orientated so that the lead exits downwards, and the red coloured edge of the cable is to the right. A bus extender cable should be used to connect FRED when an expansion interface is also present.

The audio output from FRED is via two phono sockets mounted on the box. These outputs are suitable for connection to the AUX, TUNER or TAPE inputs of most Hi-Fi amplifiers, and will produce a stereo effect if suitably programmed. A mono output can be provided by connecting the two outputs together.

2. Synthesiser Operation.

The FRED synthesiser is based around two of the General Instruments AY-3-8910 programmable sound generator chips (PSG's). A brief description of these chips follows but for more complete details consult appendices A, B and C and the AY-3-8910/8912 Data Manual.

The functional blocks of the PSG chip consist of three tone generators which feed via attenuators to three channel outputs A, B and C. These outputs are mixed externally to give one FRED output for each PSG chip. There is also a white noise generator which can be switched into any of the channel outputs. The output attenuators can be set to produce a constant output or controlled by an internal envelope generator. This enables the output level to change with time and the production of different sounds.

The envelope generator can be programmed to give various "envelopes" of sound via the attenuators, providing various combinations of note attack and decay. For example a piano-type sound would have a fast attack and a slow decay whist bowed strings would have a much slower attack.

The functions of the PSG chip are controlled by data registers inside the chip which can be accessed via I/O ports from the Z80 processor. Details of the registers and register access are given in appendix A. To avoid many complicated and repetetive operations setting PSG chip registers, software is supplied with the FRED synthesiser to convert a high-level music program into the necessary low-level functions.

3. Music Compiler Software.

The software supplied with the FRED synthesiser allows the user to control the PSG chips with mnemonic commands which can be entered, edited, executed and saved in much the same way as a BASIC program.

The software consists of two interlinked parts. The compiler section converts the program instructions that are stored in the machine (the source code) into an internal code called M-Code. The compiler also detects and flags any errors present in the source code. The second section is the play-module and this drives the sound synthesiser using instructions contained in the M-Code.

Programs are supplied on a cassette tape with the FRED sysnthesiser. One side, labled TCOMP, is for cassette-based systems and the other side, labled DCOMP, is for disk-based systems. Each program is recorded twice, starting at tape positions of 005 and 035. A sample music program is recorded at the end of each side, starting at a tape position of 070.

LOADING The method of loading is different for disk and cassette systems. If a cassette system is used then the compiler should be loaded from BASIC using the SYSTEM command, there being no need to reserve memory on entering BASIC. After typing SYSTEM the compiler can be loaded by typing TCOMP and linked to BASIC by typing /. Control will return to BASIC with a message on the screen to indicate that the compiler is operational.

For use with a disk system it is first necessary to transfer the compiler system from cassette onto disk. This is best done with the DOS utility supplied for the purpose. This could be TAPEDISK, DUMP or LMOFFSET depending on the DOS. The file is called DCOMP on the tape and should be saved onto disk as DCOMP/CIM. The program starting address is 27264 (6A80 Hex), ending address is 30294 (7656 Hex) and entry point is 30172 (75DC Hex).

Once saved on disk, the compiler should be loaded from DOS by typing LOAD DCOMP/CIM. BASIC should then be loaded in the normal way without reserving memory space. The compiler should then be linked to BASIC by entering the SYSTEM mode then typing /30172. A compiler ready message will be put on the screen and control will return to BASIC.

ENTERING AND EDITING MUSIC COMMANDS Although the commands are different from normal BASIC program statements they are entered and edited in a similar way. Command lines should begin with a line number then one or more commands. Lines are stored in ascending numerical order and line numbers are used to define lines for editing and listing. The following BASIC commands operate normally on the music program :-

LIST	AUTO	EDIT	DELETE	CLOAD	?MEM
LLIST			NEW	CSAVE	
				CLOAD?	

And with DISK BASIC, also :-

SAVE	LOAD	MERGE	KILL

COMMAND STRUCTURE The music compiler requires a strict command format and, with a few exceptions, spaces cannot be used freely. All commands consist of a 3-letter mnemonic op-code followed by a number of operands. The operand(s) should be separated from the op-code by a space and from each other by commas. Spaces can be inserted before the op-code if required for clarity.

Multiple command lines are allowed with commands separated by a colon. No spaces are permitted before a colon, but readability of the program can be improved by inserting spaces after the colon.

Labels are used to mark jump destinations and subroutines in the music program. A label should be preceeded by an asterisk and can contain any characters to a maximum length of 8 characters.

Comments can be inserted in the program and are ignored by the music compiler. They should

begin with an apostrophe after which any characters can follow. Both labels and comments are terminated by a colon or new line.

. COMMANDS These are divided into 5 sections controlling note setting, note duration, output switching and levels, envelope shape and program flow.

i) Note setting is performed by commands NTA NTB and NTC for channels A,B and C respectively. Notes are referenced by a note number in the range 0 to 95, note number 0 being the lowest and pitch increasing in semitone intervals to 95, the highest. A complete table of notes and numbers is contained in appendix E. Command format for NTA only is shown since NTB and NTC are similar.

NTA chip number,note number eg NTA 0,48

This sets channel A on the specified chip to produce the specified note. eg Channel A on chip 0 set to Middle-C.

It is possible to set all three channels on one chip simultaneously with the chord command CHD. Command format is shown below, and further chord information is given in appendix F.

CHD chip number,root note number,first interval,second interval eg CHD 1,48,4,7

Channel A on the specified chip is set to the root note number, channel B is set to the root note plus the first interval and channel C is set to the root note plus the second interval. eg Chip 1 channels A, B and C are set to Middle-C, E and G respectively. This is a Major Chord.

Tones not in the normal equally-tempered scale can be played using a modified form of the NTA, NTB and NTC commands by specifying the tone period value directly to the PSG chip. This value is related to the frequency of the tone produced by;

Frequency = 110880.21/tone period value For 10.6445 MHz crystal in Genie.

Frequency = 109166.67/tone period value For 10.4800 MHz crystal in Genie.

The note commands are modified by inserting an asterisk before the tone period value in place of the note number. Values should be integers between 1 and 4095.

NTA chip number,*tone period value eg NTA 0,*1447

This sets the specified tone period value on channel A of the specified chip. eg A tone period value of 1447 on chip 0.

ii) Note duration is controlled by two commands. The tempo command TEM determines the speed at which the music is played and the note duration command DUR the length of each note. This method allows a fixed relationship between note lengths to be maintained when the playing speed (tempo) is varied. The tempo value is usually set at the beginning of a musical passage. Values can range from 0 to 4095 but a sensible starting value is between 300 and 500. A lower value produces a faster tune.

TEM tempo value eg TEM 500

There are 16 possible note durations, numbered from 0 to 15. These note duration numbers are listed in appendix E and are related to the normal musical intervals based around a crochet, which has a duration number of 10. The DUR command is effectively a delay in the execution of the music program which allows the notes that have been set up to play for the required duration before the next notes are set up.

DUR duration number eg DUR 10

This command waits for the specified duration (eg one crochet) to allow previously set notes to play.

iii) Each channel output can be sourced from its respective tone generator or from a common white noise generator. After initialisation the three channels are all switched to tone generators, but this switching can be changed with the noise command NOS. This command also specifies a noise period which controls the "pitch" of the noise, a maximun value of 31 producing a low rumbling noise and a minimum value of 0 producing a sharp hiss.

The channels affected by this command are specified in the 3-digit channel code. An A in the first position affects channel A, a - (minus) leaves the channel unaffected. Similarly B or - in the second position controls channel B and a C or - in the third position, channel C. The

command format is shown below:

NOS chip number,noise period,channel code eg NOS 0,12,AB-

The noise generator period on the specififed PSG chip is set to the required value and the noise generator output switched to the channels affected by the channel code. Unaffected channels are switched to tone generators. eg Channels A and B of chip 0 produce noise with a noise period of 12. Channel C produces tone.

The amplitude of output from each channel is controlled by two commands, VOL (volume) and ENV (envelope). The VOL command produces a continuous sound from any channel specified in the channel code at an amplitude determined by the volume level; 0 producing no sound and 15 maximum level. Note that channels not specified in the channel code remain unaffected.

VOL chip number,volume level,channel code eg VOL 1,8,-BC

The example sets channels B and C on PSG chip number 1 to continuous output at volume level 8. Channel A remains set as before the command was executed. Initially all channels are set to a volume level of 0.

Instead of a continuous sound, the amplitude of any channel output can be controlled by the envelope generator in the PSG chip. This mode is set by the ENV command which also determines the envelope shape by means of a shape number. A list of values and shapes is given in appendix C. The volume level of the envelopes ranges from 0 to 15 and it is not possible to control the maximun output amplitude when in envelope mode.

ENV chip number,shape number,channel code eg ENV 1,8,A-C

The amplitude of the channels specified in the channel code and chip number is controlled by the envelope generator which produces an envelope determined by the shape number. The envelope is triggered as soon as the command is executed. eg A fast attack / slow decay envelope is triggered for channels A and C on PSG chip number 1. If channel B has been previously set with a VOL command it will be unaffected by this command but if previously set with an ENV command then it will be triggered with the current envelope, as channels A and C.

iv) There are two further commands that control the envelope generator, EVP (envelope period value) and EUR (envelope update register). The duration of an envelope is determined by the envelope period value which lies between 0 and 65535. A lower value results in a faster envelope, typical values being in the range 10000 to 30000.

EVP chip number,envelope period value eg EVP 0,10000

In the example the period of the envelope generator on PSG chip number 0 is set to 10000.

The envelope update register allows envelopes to be re-triggered every note, when a DUR command is executed. This avoids the need for repeating ENV commands whenever a note is to sound. The command repeats the last envelope set up on the selected PSG chip.

EUR update register value eg EUR 2

A value of 1 causes update on PSG chip number 0, a value of 2 on PSG chip number 1 and a value of 3 on both PSG chips. A value of 0 clears the re-trigger facility.

v) The remaining five commands control music program flow and are likened to similar BASIC statements where possible. The PRT (print message) command displays a message on the screen when executed.

PRT message

The message can contain any characters other than a colon which terminates the message as does new line. Note that a line feed is not sent at the end of a message, and if a new line on the screen is required then a down arrow (or cntrl) should be inserted in the message text.

Subroutines of commonly used sections within a piece of music can be written within a program and called using the JSR (jump to subroutine) command. A subroutine is terminated with an RET (return) command, whence execution continues from the command following the JSR that called the subroutine. The subroutine should be indicated with a label which is specified in the JSR command, the line number should not be used. With this exception the JSR and RET commands are the same as the BASIC commands GOSUB and RETURN.

JSR label

RET

Repetition of parts of the music program is enabled by the RPT (repeat) command. The section

of program between a specified label and the RPT command can be repeated upto 15 times. The label specified should occur before the RPT command.

RPT number of repeats,label eg RPT 2,loop

The example causes the section of program between the label "loop" and the RPT command to be executed a total of three times.

A special case of this command occurs when the number of repeats is set to zero. The resultant action is a jump to the specified label which can be before or after the RPT command. This can be used to jump around subroutines in the music program, and is similar to a BASIC GOTO statement.

RPT 0,label

The last executed statement of a music program should be the EXT (exit) command, which turns off the PSG chips and transfers control from the play module back to command mode.

EXT

COMPILING A MUSIC PROGRAM The NAME command is used to compile the music program stored in the computer. A number of options are available duing compilation; these enable you to inhibit listing of the compiler output (NL), inhibit listing of the symbol table (NS), list the compiler output on a printer (LP), and play the music after compilation (PL). Command format is;

NAME,list of options separated by commas

If no options are specified then a compiler listing and a symbol table are printed on the screen, and the music is not played. This allows for music program syntax error checking. Options can be added as required;

NL No listing.
NS No symbol table.
LP List to printer.
PL Play music.

The music can be re-played after compilation by directly executing the play module; Type SYSTEM and then /17152 for cassette systems or /27264 for disk systems. The compiler and the play module can be stopped at any time by pressing the BREAK key.

4. Getting Started with your FRED.

This section is intended as an introduction to the music program language to aid you in transcribing tunes for the FRED synthesiser. Before proceeding with this section you should have read sections 1 and 2 and the introduction to section 3. The detailed command desriptions in section 3 should be referred to as the commands are introduced here.

Firstly connect the synthesiser to the Genie and a suitable audio system then load and link-in the music compiler software as described in section 3, LOADING. Then CLOAD the demonstration music program starting at position 70 on the tape. Use the LIST command to look at the music program and familiarise yourself with the general command structure.

Now try executing the program by typing

```
NAME,PL
```

Note that the compiler prints PASS 1 then PASS 2 and then a listing with a location and a code to the left of the program, followed by a list of labels in the program. The synthesiser should the begin to operate and volume levels can be set as desired.

The compiler need not list all of the program, try instead

```
NAME,PL,NS,NL
```

To clear the demonstration music program, type

```
NEW
```

Now a fresh program can be entered.

Start by printing a heading and setting a tempo value

```
10 PRT FRED SOUND SYNTHESISER. (down arrow or cntrl) (new line)
20 TEM 300
30 #START
```

Then a sound level, note pitch and duration for channel A of chip 0

```
40 VOL 0,12,A--
50 NTA 0,48
60 DUR 10
```

The program must end with an exit command, so

```
999 EXT
```

Try executing this simple program;

```
NAME,PL
```

By listing the program during compilation, any errors that occur in the program are flagged and can be corrected.
Assuming all is well a single note should be played (Middle-C) and a READY message printed.

Now try a scale in C

```
70 NTA 0,50
80 DUR 10
90 NTA 0,52: DUR 10
100 NTA 0,53: DUR 10
110 NTA 0,55: DUR 10
120 NTA 0,57: DUR 10
130 NTA 0,59: DUR 10
140 NTA 0,60: DUR 12
```

And

```
NAME,PL
```

An accompaniment can be added using the chord command operating the other PSG chip.

```
35 VOL 1,8,ABC
45 CHD 1,36,4,7
65 CHD 1,38,4,7
85 CHD 1,40,4,7
95 CHD 1,41,4,7
105 CHD 1,43,4,7
115 CHD 1,45,4,7
125 CHD 1,47,4,7
135 CHD 1,48,4,7

NAME,PL
```

Now the envelope generator can be added to enhance the character of the solo note.

```
25 EVP 0,20000: EUR 1
40 ENV 0,9,A--

NAME,PL
```

And finally the repeat function to keep things going.

```
200 RPT 4,START

NAME,PL
```

Try editing this program, altering the tempo, relative note durations, volumes, and envelopes to your own satisfaction. Then you can try entering proper tunes from sheet music.

Appendix A. PSG Registers.

Chip number 0 gives output on the left hand channel, chip number 1 gives output on the right hand channel. For both chips :-

Register 0 = 8-bit fine tune channel A. } Channel A tone period.
Register 1 = 4-bit coarse tune channel A. }

Register 2 = 8-bit fine tune channel B. } Channel B tone period.
Register 3 = 4-bit coarse tune channel B. }

Register 4 = 8-bit fine tune channel C. } Channel C tone period.
Register 5 = 4-bit coarse tune channel C. }

Register 6 = 5-bit noise pitch control.

Register 7 = Channel output control :-

Bit 0 = Tone to channel A. }
Bit 1 = Tone to channel B. } These control bits are active
Bit 2 = Tone to channel C. } low, ie a zero in any bit position
Bit 3 = Noise to channel A. } enables the respective function and
Bit 4 = Noise to channel B. } a one disables it.
Bit 5 = Noise to channel C. }
Bit 6 = Not used.
Bit 7 = Not used.

Register 8 = 4-bit channel A amplitude. } In each case if bit 4 is set then
Register 9 = 4-bit channel B amplitude. } the channel amplitude is controlled
Register 10 = 4-bit channel C amplitude. } by the envelope generator and not
by the value set in bits 0 to 3.

Register 11 = 8-bit envelope fine tune. } 16-bit envelope period.
Register 12 = 8-bit envelope coarse tune. }

Register 13 = 4-bit envelope shape control.

Register Access :-

PSG chip 0 - OUT 124 (7Ch) , Register number.
OUT 125 (7Dh) , Register data.

PSG chip 1 - OUT 126 (7Eh) , Register number.
OUT 127 (7Fh) , Register data.

Appendix B. PSG Tone period values.

Note	Tone period Hex	Tone period Dec	Register entry Coarse	Register entry Fine
Octave 1 (bottom octave)				
C	0D3E	3390	13	62
C#	0CB0	3200	12	128
D	0BCD	3021	11	205
D#	0B23	2851	11	35
E	0A83	2691	10	131
F	09EC	2540	9	236
F#	095D	2397	9	93
G	08D7	2263	8	215
G#	0858	2136	8	88
A	07E0	2016	7	224
A#	076F	1903	7	111
B	0704	1796	7	4
Octave 2				
C	069F	1695	6	159
C#	0640	1600	6	64
D	05E6	1510	5	230
D#	0592	1426	5	146
E	0542	1346	5	66
F	04F6	1270	4	246
F#	04AF	1199	4	175
G	046B	1131	4	107
G#	042C	1068	4	44
A	03F0	1008	3	240
A#	03B7	951	3	183
B	0382	898	3	130
Octave 3				
C	0350	848	3	80
C#	0320	800	3	32
D	02F3	755	2	243
D#	02C9	713	2	201
E	02A1	673	2	161
F	027B	635	2	123
F#	0257	599	2	87
G	0236	566	2	54
G#	0216	534	2	22
A	01F8	504	1	248
A#	01DC	476	1	220
B	01C1	449	1	193
Octave 4				
C	01A8	424	1	168
C#	0190	400	1	144
D	017A	378	1	122
D#	0164	356	1	100
E	0150	336	1	80
F	013E	318	1	62
F#	012C	300	1	44
G	011B	283	1	27
G#	010B	267	1	11
A	00FC	252	0	252
A#	00EE	238	0	238
B	00E1	225	0	225

Note	Tone period Hex	Tone period Dec	Register entry Coarse	Register entry Fine
Octave 5 (middle octave)				
C (mid-C)	00D4	212	0	212
C#	00C8	200	0	200
D	00BD	189	0	189
D#	00B2	178	0	178
E	00A8	168	0	168
F	009F	159	0	159
F#	0096	150	0	150
G	008D	141	0	141
G#	0086	134	0	134
A	007E	126	0	126
A#	0077	119	0	119
B	0070	112	0	112
Octave 6				
C	006A	106	0	106
C#	0064	100	0	100
D	005E	94	0	94
D#	0059	89	0	89
E	0054	84	0	84
F	004F	79	0	79
F#	004B	75	0	75
G	0047	71	0	71
G#	0043	67	0	67
A	003F	63	0	63
A#	003B	59	0	59
B	0038	56	0	56
Octave 7				
C	0035	53	0	53
C#	0032	50	0	50
D	002F	47	0	47
D#	002D	45	0	45
E	002A	42	0	42
F	0028	40	0	40
F#	0025	37	0	37
G	0023	35	0	35
G#	0021	33	0	33
A	0020	32	0	32
A#	001E	30	0	30
B	001C	28	0	28
Octave 8 (top octave)				
C	001A	26	0	26
C#	0019	25	0	25
D	0018	24	0	24
D#	0016	22	0	22
E	0015	21	0	21
F	0014	20	0	20
F#	0013	19	0	19
G	0012	18	0	18
G#	0011	17	0	17
A	0010	16	0	16
A#	000F	15	0	15
B	000E	14	0	14

Appendix C. PSG Envelope shapes.

Envelope shape number.	Envelope features.
0 to 7	Not Used.
8	Fast attack then slow decay, continuously repeating.
9	Fast attack then slow decay, one shot.
10	Slow decay then slow attack, continuously repeating.
11	Slow decay then fast attack, remaining at high level.
12	Slow attack then fast decay, continuously repeating.
13	Slow attack to constant high level.
14	Slow attack then slow decay, continuously repeating.
15	Slow attack then fast decay, one shot.

Note.

The envelope period control registers determine the speed of a slow attack or decay cycle. A value of 6930 (27 coarse, 18 fine) gives an envelope time of 1 second. A value of 3465 (13,137) gives half a second.

An envelope is triggered when the shape control value is written into PSG register 13.

Appendix D. Music compiler command summary.

Source program commands.

```
NTA p,nn          } Set note number nn (0 to 95) on channel A B or C of PSG
NTB p,nn          }   chip number p (0 or 1).
NTC p,nn          }

NTA p,#nnnn       } Set tone period nnnn (0 to 4095) on channel A B or C of
NTB p,#nnnn       }   PSG chip number p (0 or 1).
NTC p,#nnnn       }

CHD p,nn,aa,bb    } Set note number [nn] on channel A, [nn+aa] on channel B
 (Chord)          }   and [nn+bb] on channel C of PSG chip number p (0 or 1).
                  }   (aa and bb 0 to 15)     (note number 0 to 95).

NOS p,nn,ccc      } Switch channels specified in code ccc to noise instead of
 (Noise)          }   tone on PSG chip number p (0 or 1).
                  }   The noise period is set to nn (0 to 31).

VOL p,nn,ccc      } Set channels specified in code ccc to volume level
                  }   nn (0 to 15) on PSG chip number p (0 or 1).

ENV p,nn,ccc      } Trigger envelope shape number nn (8 to 15) for channels
 (Envelope)       }   specified in ccc on PSG chip number p (0 or 1).

EVP p,nnnn        } Set envelope period to nnnn (0 to 65535) on PSG chip
 (Env Period)     }   number p (0 or 1).

TEM nnnn          } Set the tempo of the music to value nnnn (0 to 4095).

DUR nn            } Wait for a duration of length nn (0 to 15) before
 (Duration)       }   proceeding with subsequent commands.

EUR c             } Re-trigger envelopes on execution of a DUR command on
 (Envelope        }   PSG chips specified in code c (0 to 3).
   Update)        }   c=1  PSG chip number 0  :  c=2  PSG chip number 1
                  }   c=3  Both PSG chips     :  c=0  Clear re-trigger mode.

RPT nn,lllll      } Repeat nn (1 to 15) times from label lllll (8 chars max).
                  }   If nn=0 then an uncondition jump to label lllll occurs.

JSR lllll         } Call subroutine at label lllll (8 chars max).

RET               } Return to main program from subroutine.

PRT $$$$$$        } Print message $$$$$$ on screen. Message can be up to 110
                  }   characters long and is terminated with NEW LINE or a colon.

EXT               } Exit from play module and enter command mode.
                  } Must be last statement.
```

FRED Sound Synthesiser.

Channel code ccc :-

A--	Channel A.	-B-	Channel B.
--C	Channel C.	---	No Channel.
AB-	Channels A and B. etc.		

lables :- Are preceeded by an asterisk and terminated with a colon or NEW LINE.
Only the first 8 characters are noted by the compiler.

Comments :- Can be inserted at will and should begin with an apostrophe.

Multiple Statements are permitted on a line if separated by a colon. No spaces should be entered before the colon.

Compiler loading :-

Tape version - Enter BASIC as normal with full memory size.

```
> SYSTEM
*? TCOMP
*? /
```

Disk version - From DOS

```
LOAD DCOMP/CIM
Enter BASIC as normal.
> SYSTEM
*? /30172
```

Compiler execution command :- NAME,opt1,opt2,.....

Options are :-

PL	Play M-Code after compilation.
NL	No listing of compiler source on screen.
NS	No listing of compiler symbol-table on screen.
LP	List compiler source on printer.

Options can be entered as required in any order.

Re-running M-Code once compiled.

Tape version -

```
> SYSTEM
*? /17152
```

Disk version -

```
> SYSTEM
*? /27264
```

Extending beyond 2 PSG Chips.

The compiler will support a maximum of 16 PSG chips ported from 7C (Hex) upwards. The chip number p then extends from 0 to 15 and the EUR argument to a full 16-bit integer, a one in each bit position updating the respective chip, ie bit 0 (lsb) = PSG chip 0.

Appendix E. Music compiler note numbers and lengths.

Note numbers used with the NTA, NTB, NTC and CHD commands.
Note number 48 is Middle-C.

Octave 1		Octave 2		Octave 3		Octave 4		Octave 5		Octave 6		Octave 7		Octave 8	
C	0	C	12	C	24	C	36	C	48	C	60	C	72	C	84
C#	1	C#	13	C#	25	C#	37	C#	49	C#	61	C#	73	C#	85
D	2	D	14	D	26	D	38	D	50	D	62	D	74	D	86
D#	3	D#	15	D#	27	D#	39	D#	51	D#	63	D#	75	D#	87
E	4	E	16	E	28	E	40	E	52	E	64	E	76	E	88
F	5	F	17	F	29	F	41	F	53	F	65	F	77	F	89
F#	6	F#	18	F#	30	F#	42	F#	54	F#	66	F#	78	F#	90
G	7	G	19	G	31	G	43	G	55	G	67	G	79	G	91
G#	8	G#	20	G#	32	G#	44	G#	56	G#	68	G#	80	G#	92
A	9	A	21	A	33	A	45	A	57	A	69	A	81	A	93
A#	10	A#	22	A#	34	A#	46	A#	58	A#	70	A#	82	A#	94
B	11	B	23	B	35	B	47	B	59	B	71	B	83	B	95

Note duration numbers used with the DUR command.

Duration No.	Fraction of Crochet.	Definition.
0	1/32	Demi demi demi semi quaver.
1	1/16	Demi demi semi quaver.
2	1/8	Demi semi quaver.
3	1/6	Semi triplet.
4	3/16	Dotted demi semi quaver.
5	1/4	Semi quaver.
6	1/3	Triplet.
7	3/8	Dotted semi quaver.
8	1/2	Quaver.
9	3/4	Dotted Quaver.
10	1	Crochet.
11	3/2	Dotted crochet.
12	2	Minim.
13	3	Dotted minim.
14	4	Semi breve.
15	8	Breve.

Note:- The relative durations of the notes is determined only by the DUR command, but the actual duration is also affected by the TEM (tempo) command.

Appendix F. Chord table.

Notes can be played in certain pleasant combinations or chords. A chord is formed from a root note and several other notes which deviate from the root note by a fixed number of semitones. The table below shows the notes corresponding to the musical intervals from each root note.

Root note	3rd Minor	3rd Major	4th	5th	6th	7th
C	D#	E	F	G	A	A#
C#	E	F	F#	G#	A#	B
D	F	F#	G	A	B	C
D#	F#	G	G#	A#	C	C#
E	G	G#	A	B	C#	D
F	G#	A	A#	C	D	D#
F#	A	A#	B	C#	D#	E
G	A#	B	C	D	E	F
G#	B	C	C#	D#	F	F#
A	C	C#	D	E	F#	G
A#	C#	D	D#	F	G	G#
B	D	D#	E	F#	G#	A

Semitone intervals :-

0	+3	+4	+5	+7	+9	+10

A Major chord is made up of the root note combined with the 3rd Major and 5th intervals, thus a C-Major chord is C + E + G. An extra C note, one octave above the root C, can be added to improve the depth of the chord.

A Minor chord consists of the root with the 3rd Minor and 5th intervals, thus a C-Minor chord is C + D# + G. There are many more possible chords that can be generated by selecting different intervals from a root note.

Chords can easily be set up with the music compiler, since the CHD command allows you to specify the root note and two intervals. eg.

```
CHD 0,48,4,7        Specifies a Major chord with root middle-C

CHD 0,28,4,7        Specifies an E-Major chord.

CHD 0,26,3,7        Specifies a D-Minor chord.
```

Appendix G. Play Module M-Code.

Compiler Mnemonic.	Binary M-Code.			
NTA p,nn	0001pppp	0nnnnnnn		
NTB p,nn	0010pppp	0nnnnnnn		
NTC p,nn	0011pppp	0nnnnnnn		
NTA p,#nnnn	0001pppp	1000nnnn	nnnnnnnn	
NTB p,#nnnn	0010pppp	1000nnnn	nnnnnnnn	
NTC p,#nnnn	0011pppp	1000nnnn	nnnnnnnn	
CHD p,nn,aa,bb	0100pppp	0nnnnnnn	aaaabbbb	
NOS p,nn,ccc	1000pppp	nnnnnccc	} Channel code ccc	
			} represented as ones	
VOL p,nn,ccc	0101pppp	nnnn0ccc	} in respective positions	
			} when channel is selected.	
ENV p,nn,ccc	0110pppp	nnnn0ccc	}	
EVP p,nnnn	0111pppp	nnnnnnnn	nnnnnnnn	
TEM nnnn	1010nnnn	nnnnnnnn		
DUR nn	1001nnnn			
EUR cccc	10110000	cccccccc	cccccccc	
RPT nn,llllll	1101nnnn	aaaaaaaa	aaaaaaaa	00000000 } Address aaaa is the
				} absolute 16-bit of
JSR llllll	11100000	aaaaaaaa	aaaaaaaa	} labelled M-Code.
RET	11110000			
PRT $$$$$$	11000000	ASCII coded bytes.	00000000	
EXT	00000000			

Play Module Locations.	Cassette Version		Disk Version	
Start and Entry point	17152	4300 (Hex)	27264	6A80 (Hex)
End of Play Module	17973	4635 (Hex)	28085	6DB5 (Hex)
Entry point for Compiler	20060	4E5C (Hex)	30172	75DC (Hex)
End of Music Compiler	20182	4ED6 (Hex)	30294	7656 (Hex)
Pointer to start of M-Code	17958/9	4626/7 (Hex)	28070/1	6DA6/7 (Hex)