

MUSIC SYSTEM

COPYRIGHT (C) 1980 BY HOWARD ARRINGTON

HARDWARE -- Assembled A/D converter connects to parallel port. Simply connect an amplifier and speaker such as your stereo to listen to music. A \$10 amplifier from Radio Shack works, but low notes saturate the 2 inch speaker.

MUSIC GENERATION -- Machine language routine generates four part harmony of true frequencies over a 4 and 1/2 octave range. The routine easily interfaces with your Basic programs to add sound effects and music. Well documented.

EDITOR -- System includes a screen editor that displays the two musical staves upon which musical notes are placed. Quickly create song data files using the following features. Data file creation is done graphically on the screen.

TO USE -- Type **BYE** (Exit to Monitor)
 >LO MUSIC (Load music generation routine)
 >PP (Exit back to Basic)
 RUN (Execute editor, screen displays staves and keypad definitions)
 PRESS 'ALTR KPAD' KEY (Second set of 10-keypad definitions displayed on screen)
 PRESS 'LOAD FILE' KEY (Reads the peppy demonstration song file FANF)
 PRESS 'PLAY SONG' KEY (The demonstration song of FANFARE is played. People love it.)

FEATURES -- Tempo control from slow to exceedingly fast.
Set the music's key signature by placing sharps or flats on the staves.
Select notes by moving cursor up and down musical staves. It's like copying what you see on music sheets.
Graphically select note durations and rests.
Full editing of inserting, deleting or overwriting chord sets in the song data file.
Hear a pitch in a chord before the note is entered into the data file.
Play the last ten chord sets to verify the accuracy of the data just entered. Or play the entire song.
Copy refrains for rapid duplication of repeated measures.
Transpose the playing of the song into another key. Adjust to suit your vocal range for singing along.
Automatic loading and saving of data files from and to cassette.
Restoration of data file notes to the screen for rapid editing and file verification.
Single step through the file listening to each chord. Aids in file editing.
User prompts and error protections.
Select individual voices to play only their part. Example: hear the Bass and Soprano parts together.

EASY TO USE -- Customers have often written saying they are interested, but hesitant to buy music software because they know nothing about music. You don't need to know all about music to enjoy this software. You can experiment with the editor, dabble in song writing, and enjoy prewritten music such as PIANO PLAYER, FANFARE, EACH AND BOGGY WOOGY. Creating song data files is done graphically. Just copy sheet music onto the screen. You are invited to call and hear a demonstration song played over the phone. Dial 208-377-1938 after 7 p.m. Mountain Standard time.

PIANO PLAYER

This program is compatible with the above music system and adds delightful graphical animation of a high resolution piano player playing your songs. The little man's arms move in synchronization with the beat of the music. A large keyboard is displayed upon which four cursors jump around on the keys to the four notes being played. Piano player comes with a medley of favorite Christmas songs to brighten the coming season. Being both graphical and musical, this program is a crowd pleaser twice over. (Note: MUSIC SYSTEM necessary.) This program will make a super Christmas gift.

TO USE: **>LO MUSIC** } **ONE LOAD** (Load in MUSIC system)
 >LO PIANO } (Load in Piano Player routine)
 PRESS 'LOAD FILE' KEY (Execute editor and load a song data file such as the included XMAS file)
 PRESS 'PIAN PLAY' KEY (Whatever song file you have loaded is played)

* You must re-establish the key signature after running or after loading a file. *

MUSIC SYSTEM

COPYRIGHT (C) 1980 BY HOWARD ARRINGTON

HARDWARE -- Assembled D/A converter connects to the parallel port. Simply connect an amplifier and speaker such as your stereo to listen to music. A \$10 amplifier from Radio Shack works, but low notes saturate the 2 inch speaker. As with any musical devices, the better the amplifier and speaker system, the better the quality of the sound reproduced.

TO USE -- Type BYE (Exit to Monitor)
)LO MUSIC (Load music generation routine and editor.)
)PP (Exit back to Basic)
 RUN (Execute editor, screen displays staves and keypad definitions)
 PRESS 'ALTR KPAD' KEY (Press the + key on the 16-keypad to display the alternate keypad)
 PRESS 'LOAD FILE' KEY (Follow prompt and load data file on tape called FANF)
 PRESS 'PLAY SONG' KEY (The demonstration song FANFARE will be played)

MUSIC GENERATION -- A machine language routine generates four part harmony of true frequencies over a 4 and 1/2 octave range. The routine easily interfaces with your Basic programs to add sound effects and music. A second harmonic sinusoidal waveform has been digitized and stored with one period of the waveform occupying 256 bytes of memory. One may want to experiment with the timbre and tone generated by using a different waveform than the one supplied with this software. The following Basic statements outline how the supplied waveform was generated:

```
100 PI = 2 * 3.14159 : FOR I=0 TO 255 : ANGLE = PI * I / 255 : VALUE = SIN(ANGLE) + SIN(ANGLE*2)
110 POKE OFFSET + I,VALUE*SCALE + SHIFT : NEXT I
```

Values used: OFFSET = 10752 ;Locates table at 2A00H in memory.
 SCALE = 16 ;Range now approximately +30 to -30.
 SHIFT = 32 ;Keeps value from being negative.

The following description is for one voice, but four voices are achieved by duplicating the process independently four times. Suppose that the speed of the microprocessor is such that it can look up a value in the 256 byte waveform table and output the value to the parallel port at a rate of 256 times per second. At this speed it would cycle through one period of the waveform each second and generate a one hertz frequency. If, however, every other value in the waveform were skipped, it would pass through two periods of the table in one second and output a 2 hertz frequency. This is the basic principle by which frequencies are obtained: varying the increment that is added to the pointer as it cycles through the waveform table and looks up values to send to the D/A converter at the parallel port.

To generate a composite waveform of four voices, the four values obtained for four voice are added together. That is why the scaling of the waveform is between 0 and 64. Thus, if the four voices each retrieve a value of 63 from the waveform table, the sum will be less than 255, the maximum value represented on the 8-bit parallel port.

TEMPO TABLE -- The tempo table contains tempo information so that the meter of a song can be changed. If one specifies a meter of 60 then the tempo of the song will be played such that there will be 60 quarter note beats per minutes. A meter of 100 will speed up the tempo to 100 quarter beats per minute. One can select a tempo from 40 to 158 beats per minutes. The numbers from 159 through 166 are special fast tempos that exceed 175 beats per minutes.

Note timing is stored as a multiple of 1/32 of a whole note. A 1/32 note has a timing of 1. A quarter note is an 8.

FREQUENCY TABLE -- A table resides in memory with the necessary pointer increments to generate the pitches in a music scale. Select frequencies by subscripting the table with the numbers that follow. Each half step on the musical scale is the next entry in this table of pointer increments. Middle C is entry #25. The C above the treble clef is #49.

OCTAVE	0	1	2	3	4	FREQUENCY (OCTAVE *2)	EDITOR NOTE TIMING
B	12	24	36	48		493.9 Hz	First character 1/32
A#	11	23	(35)	47		466.2	Second char 1/16
A	10	22	34	46		440	Third character 3/32
G#	9	21	33	45		415.3	Fourth 1/8
G	8	20	(32)	44	56	392	Fifth 3/16
F#	7	19	31	43	55	370	Sixth 1/4
F	6	18	30	42	54	349.2	Seventh 3/8
E	5	17	29	41	53	329.7	Eighth 1/2
D#	4	16	28	40	52	311.2	Ninth 3/4
D	3	15	27	39	51	293.7	Tenth whole note
C#	2	14	26	38	50	277.2	Eleventh two wholes
C	1	13	25	(37)	49	261.7	Twelfth four wholes
SILENCE	0						Thirteenth silence

INTERFACING TO YOUR PROGRAMS -- The music generation routine occupies 400H bytes from 2A00H through 2DFFH. There are two ways to utilize the music routine: 1) play a single chord, or 2) play a series of chords from a file.

PLAY SINGLE CHORD -- To play a single chord of up to four voices, the note number for each voice must be placed in memory in bytes 1, 2, 3 and 4. For example, to play a C7 chord consisting of E, G, B flat, and C, one would look up the note numbers in the frequency table and use the following Basic statements to place the numbers in memory:

```
100 POKE 1,29 : POKE 2,32 : POKE 3,35 : POKE 4,37 : REM Notes E, G, B flat (same as A#), and C
```

Any of the above voices could be silenced by poking a zero for the note number. Now, store the duration for the notes to be played in bytes 5 and 6.

```
110 POKE 5,0 : POKE 6,6 : REM 6 * 256 times through loop, or about one second.
```

Now call the entry address at 2D44H through the USR() function.

```
120 POKE 260,68 : POKE 261,45 : X=USR(0) : REM This POKEs then USR() address.
```

The routine is called and the chord is played. Basic will return to the statement after the USR() function call. The contents of bytes 1 through 6 are not destroyed which allows one to replay the chord by repeating line 120. Or, you can change the notes or the tempo and play something different.

PLAY SONG FILE -- Store in memory a file of musical sets which consist of four voice numbers and a fifth byte for the timing of the chord as a multiple of 1/32 of a whole note. Enter zeroes for silent voices. The example places two sets in memory plus the STOP FLAG:

```
200 FOR I=0 TO 14 : READ A : POKE FILE+I,A : NEXT I
210 DATA 29,32,35,37,8 : REM C7 chord played as a quarter note
220 DATA 25,25,0,0,32 : REM 2 voices play C, 2 silent. Whole note.
230 DATA 0,0,0,0,0 : REM 0 timing byte is the stop flag.
```

The starting address of your file is placed in memory in the two bytes at 2D11H. For example, if you started your file at 3800H, the following Basic statements place the address:

```
300 POKE 11537,0 : POKE 11538,56 : REM High byte (38H) of address in byte 2D12H, ie. 11538 decimal.
```

The end of the file is flagged with the timing byte of the last dummy set being a zero. Be sure to put in this extra set so that the routine knows that it is finished playing your song. Now, place the meter speed in byte 0 to control the tempo. The value of the byte is between 0 and 127, but the tempo table has been set up so that the first meter in the table is a meter of 40 beats per minute. The relation between value stored and the meter is: Value = Meter - 39.

310 POKE 0,100-39 : REM Select meter of 100 beats per minute.

Call the music routine at 2D00H which is a different entry address than the one used to play a single chord. Entry at this address will cause the routine to compute the duration for bytes 5 and 6 from the tempo in byte 0 and the timing byte stored with each chord set. When the entire file is finished playing, the routine will return to your program.

400 POKE 260,0 : POKE 261,45 : X=USR(0) : REM Poke USR address and call routine.

TRANPOSE -- One can transpose all of the notes played either up or down according to the number of half steps stored in byte 9. To shift all notes up one half step, store a 1 in byte 9. To shift down 2 half steps, store -2 in byte 9.

500 IF TRANPOSE < 0 THEN TRANPOSE = TRANPOSE + 256 : REM For negative numbers.

510 POKE 9,TRANPOSE

ADDRESS REFERENCES	—	WAVEFORM	FREQUENCY	TEMPO	PLAYFILE	FILEADDRESS	PLAYCHORD
	Hex	2A00H	2B00H	2C00H	2D00H	2D11-2H	2D44H

MUSIK EDITOR -- System includes a screen editor that displays the two musical staves upon which musical notes are placed. Quickly create song data files using the following features. Data file creation is done graphically on the screen. The editor is a Basic program loaded with the music package as part of the XLO MUSIC.

FEATURES -- Tempo control from slow to exceedingly fast. Graphically select note durations and rests.
 Set the music's key signature by placing sharps or flats on the staves.
 Select notes by moving cursor up and down musical staves. It's like copying what you see on music sheets.
 Full editing of inserting, deleting or overwriting chord sets in the song data file.
 Hear a pitch as a member of the selected chord before the note is entered into the data file.
 Play the last ten chord sets to verify the accuracy of the data just entered. Or play the entire song.
 Copy refrains for rapid duplication of repeated measures.
 Transpose the playing of the song into another key. Adjust to suit your vocal range for singing along.
 Restoration of the data file notes to the staves. Excellent aid in transposing and in file editing.
 Single step through a song to locate an incorrect entry for easy editing.
 Automatic loading and saving of data files to and from cassette.

ALTERNATE KEYPAD -- All features of this music editor are accessed from the 10-keypad. Keypad definitions are displayed on the screen. An alternate set of key definitions is obtained by pressing the 'ALTR KPAD' key. A second pressing of this key will restore the initial keypad definitions.

SELECT NOTES -- Notes are selected by their position on the two staves. Move the cursor arrow up and down by holding down the '8' or the '2' keys on the 10 keypad. The cursor moves as long as the 8 or the 2 is depressed. Now that the cursor has been moved to the desired staff position, press the '5' key and the note will appear on the staff as well as register as one of the four notes selected at the top right corner of the screen. Continue moving the cursor and pressing '5' to select the other three voices in the chord.

SCREEN RANGE -- Unfortunately, there are not enough lines on the screen to cover the entire range of notes possible. However, as the cursor accesses notes beyond the screen limits, messages are displayed to indicate the beyond screen range, and the cursor is mapped back onto the screen one octave's distance. Thus, all 56 pitches can be accessed.

NOTE TIMING -- The graphic notes at the bottom of the screen are used to select the duration of the notes placed on the screen. The '6' and the '4' keys move the timing pointer to the desired note timing. This pointer is also used to silence a voice with the rest symbol when '5' is pressed, i.e. press 'SAVE NOTE' while pointing to the rest symbol.

SAVE SET -- The four notes selected and the timing pointed to at the bottom of the screen become the 5 bytes stored in the data file when the 'COPY SET' key is pressed. The counter of the number of sets stored in memory is incremented. This counter is displayed at the top of the screen as 'SET # xx OF yy' where xx is the set number that will be stored the next time 'COPY SET' is pressed, and yy is the total number of sets in the file.

REPLACE A SET -- One of the edit features is the ability to replace an incorrectly keyed chord. Use the '1' and the '3' keys to change the set number back to the one that is in error. Using the '3' key, the four voices and the note timing are reestablished for easy editing of an incorrect voice or timing. Press 'COPY SET' to resave an edited set.

SHARPS & FLATS -- The '9' key moves the key pointer to the sharp and flat symbols. Now when 'SAVE NOTE' is pressed, the note will appear on the staff with an accompanying sharp, flat or natural symbol. Be sure to return the pointer to the SPACE position so that other voices don't accidentally become sharpened or flattened.

KEY SIGNATURE -- One can set the music's key signature by moving the note select cursor to the staff position, moving the key pointer to the sharp or the flat symbol, and then pressing the '7' key. Two graphic symbols will appear on the left side of the staves to represent that the notes placed on these lines are automatically sharpened or flattened. This signature will be selected whenever the key pointer is in the SPACE position. The signature is overridden by the key pointer pointing to the sharp, flat or natural graphic symbols.

STAFF DISPLAY -- As each set is entered into the file with 'COPY SET', the notes on the staff shift to the left to make room for the next set to be placed on the staves. If the next set is the same as the previous, simply press 'COPY SET' again and the set is duplicated. One only needs to change the notes that differ from the previous set.

VOICE # EDIT -- The note selected with 'SAVE NOTE' will be for the voice number shown at the top of the screen. That voice number can be changed by pressing the '=' key: 'SAVE NOTE' automatically increments voice # in anticipation of the next voice being entered, and 'COPY SET' resets the voice number to voice #1.

HEAR PITCH -- Holding down the '-' key on the 10 keypad will repeat the pitch of the note pointed to by the cursor, in company with the other three voices making up the chord. Use 'NEXT VOIC' to substitute pitch for a different voice #.

PLAY SCREEN -- The '/' key will play the 10 sets prior to the current set number. This helps reduce entry errors and ease correction by allowing one to hear just the last section entered into the file.

PLAY SONG -- The 'x' key will play the entire song file.

TEMPO CONTROL -- The '0' key allows you to select a tempo from 40 to 166 inclusive. For example, enter 100 and press RETURN for a tempo of 100 quarter beats per minute. Numbers 159 to 166 are special fast tempos.

TRANPOSE -- The 'TRNS POSE' key allows one to transpose the number of half steps entered. Example: enter 1 to transpose the playing of the song up one half step. A -3 will play the music 3 half steps lower. A zero value plays the music as entered. Notes restored to the screen do so transposed. * up or down $\frac{1}{2}$ octave *

DELETE SET -- Unwanted chord sets can be removed by pressing the 'DEL SET' key on the alternate keypad. First, move the set number to the set that is to be removed and then press 'DEL SET'. The total number of sets will decrease by 1.

INSERT SET -- One can insert a set into the file by pressing the 'INSR SET' key. First, move the set number to the set where the new set is to be inserted ahead of. Press 'INSR ST'. The total number of sets will increase by one and the current set number will remain unchanged. Now select the chord notes and timing and press 'COPY SET'. Typically, you would move the set number back to place where you were previously entering.

COPY REFRAINS -- The 'COPY RFRN' key allows one to copy a group of sets to the location of the current set number. Enter the set number at the start of the refrain and the set number to copy through. Example: REFRAIN #5 TO #17 will copy set 5 through and including set 17 starting at the current set number. The current set number will increase by the number of sets copied, but the total number may not change if recoding in the middle of a file.

CASSETTE -- Save and load song data files on cassette with 'SAVE FILE' and the 'LOAD FILE' keys. The data file starts at 3800 hex and its length is dependent on song length, which is automatically computed by the save command.

SINGLE STEP -- The 'STEP CORD' key on the alternate keypad will single step through the song data file making it easy to locate incorrectly keyed chords. When the bad chord is heard, release the 'STEP CORD' key, return to the main keypad, press the '1' and the '3' keys each once to put the chord on the screen for easy editing.

TOGGLE VOICE -- The 'TOGL VOIC' key is used to eliminate or restore individual voices from being played. This feature will allow one to hear one or more voices by turning off the other voices. Enter a voice number. If it is presently on, it will be turned off. If the voice is presently off, it is turned back on. This feature will be of great value to vocalists learning individual parts.

SET ADDRESS -- The 'SET ADDR' key prints the current set's memory location in decimal.

ERASE FILE -- The 'ERAS FILE' key resets certain variables so that a new file can be keyed in. It is not necessary to erase an existing file before loading one from cassette tape with the 'LOAD FILE' command.

PIANO PLAYER -- The optional program of Piano Player, if loaded into memory, is called with the 'PIAN PLAY' key. If one has not purchased and loaded this program, then the message 'NOT INSTALLED' is displayed on the screen.

EXIT -- 'EXIT' key is used to return to Basic.

ADDR	OBJECT	STMT	LABEL	OPCD	OPERAND	COMMENT
		1				; FOUR PART HARMONY MUSIC FOR THE SORCERER
		2				
		3				; &MUSIC: DEC 1979 COPYRIGHT (C) BY HOWARD ARRINGTON
		4				
		5				; ROUTINE STORED AT ^{2A00} 6400H . ENTRY POINTS ARE:
		6				6400H - READS NOTES FROM USER'S FILE AT 1700H.
		7				6444H - FINDS NOTE INCREMENTS @ TABLE @ 6200H
		8				649EH - PLAYS NOTES - WAVEFORM TABLE AT 6100H
		9				
		10				; BASIC STATEMENTS WHICH CALL ROUTINE AT 6400H ARE:
		11				POKE 260,0:POKE 261,100:X=USR(0)
		12				
		13				; STORED 256 BYTE WAVEFORM TABLE AT 6100H-61FFH.
		14				; STORED 256 BYTE TIMING TABLE AT 6300H-63FFH.
		15				; STORED 122 BYTE FREQUENCY TABLE AT 6200H-62FFH.
		16				EACH NOTE WILL HAVE A TWO BYTE INCREMENT
		17				IN THE FREQUENCY TABLE.
		18				FIRST TWO BYTES IN TABLE ARE ZERO FOR SILENCE
		19				
		20				; STORE NOTE #1 IN BYTE 01H.
		21				NOTE #2 IN BYTE 02H.
		22				NOTE #3 IN BYTE 03H.
		23				NOTE #4 IN BYTE 04H.
		24				NOTE TIME BYTE 00H.
		25				TRANPOSE BYTE 09H.
		26				
		27				; STORE SEQUENCE OF MUSIC NOTES AND TIMING IN MEMORY
		28				STARTING AT 1700H. STORE FOUR VOICE BYTES FOLLOWED
		29				BY THE NOTE TIMING BYTE. LAST BYTE TIME=0 TO STOP
		30				
		31				;*****
		32				
		33				; DEFINE STORAGE SPACE FOR ROUTINE VARIABLES
		34				
		35				ORG 0H
0000	00	36	TEMPO	DEFB	0	;TEMPO NUMBER 40-166
0001	00	37	N1	DEFB	0	;NOTES #1 - #4
0002	00	38	N2	DEFB	0	
0003	00	39	N3	DEFB	0	
0004	00	40	N4	DEFB	0	
0005	0000	41	DUR	DEFW	0	;TEMPO TIMES NOTE TIME
0007	0000	42	MULT	DEFW	0	;MULTIPLIER FROM TIMING TABLE
0009	0000	43	TRANS	DEFW	0	;TRANPOSE ADD - USUALLY ZERO
		44				
000B	0000	45	P1	DEFW	0	;TABLE POINTERS #1 - #4
000D	0000	46	P2	DEFW	0	
000F	0000	47	P3	DEFW	0	
0011	0000	48	P4	DEFW	0	
		49				
0013	0000	50	I1	DEFW	0	;DETERMINES NOTE FREQUENCY
0015	0000	51	I2	DEFW	0	;LOOKED UP IN TABLE
0017	0000	52	I3	DEFW	0	
0019	0000	53	I4	DEFW	0	

```

54 ;
55 ;*****
56 ;
57 ; INITIALIZE ROUTINE POINTERS AND INCREMENTS
58 ;
>0061 59 HIGH EQU 61H ;HIGH AND PROG RELOCATE LOCA
>6100 60 PROG EQU 6100H ;WHERE ROUTINE IS STORED IN
>1700 61 MUSIC EQU 1700H
62 ;
63 ORG PROG+0300H ;SKIP OVER THREE TABLES
6400 210063 64 START: LD HL,PROG+0200H ;POINT TO TIMING TABLE
6403 3A0000 65 LD A,(TEMPO) ;OFFSET FOR TABLE
6406 CB27 66 SLA A ;MULTIPLY BY 2
6408 6F 67 LD L,A ;COMPLETE TABLE ADDRESS
6409 56 68 LD D,(HL) ;GET HIGH MULTIPLIER
640A 23 69 INC HL
640B 5E 70 LD E,(HL) ;GET LOW MULTIPLIER
640C ED530700 71 LD (MULT),DE ;SAVE DURATION'S MULTIPLIER
72 ;
6410 210017 73 LD HL,MUSIC ;START OF COMPILED MUSIC TAB
6413 4E 74 TOP: LD C,(HL)
6414 23 75 INC HL
6415 46 76 LD B,(HL) ;FIRST 2 NOTES
6416 ED430100 77 LD (N1),BC
641A 23 78 INC HL
641B 4E 79 LD C,(HL)
641C 23 80 INC HL
641D 46 81 LD B,(HL) ;NOTES 3 AND 4 MOVED
641E ED430300 82 LD (N3),BC
6422 23 83 INC HL
6423 7E 84 LD A,(HL) ;GET NOTE TIME
6424 B7 85 OR A ;SETS ZERO FLAG IF =0
6425 C8 86 RET Z ;EXIT TO CALLING PROGRAM
6426 23 87 INC HL
6427 E5 88 PUSH HL ;SAVE MUSIC FILES ADDRESS
89 ;
6428 0608 90 LD B,8 ;START OF MULTIPLICATION
642A ED5B0700 91 LD DE,(MULT) ;TIMING MULTIPLIER
642E 210000 92 LD HL,0 ;CLEAR SUMMING REGISTER
6431 CB1F 93 LOOP: RR A ;ROTATE MULTIPLIER
6433 3001 94 JR NC,NOADD ;SKIP ON NO CARRY
6435 19 95 ADD HL,DE ;ELSE ADD TO PARTIAL RESULT
6436 EB 96 NOADD: EX DE,HL ;SHIFT MULTIPLICANT BY ADDING
6437 29 97 ADD HL,HL
6438 EB 98 EX DE,HL
6439 10F6 99 DJNZ LOOP ;REPEAT UNTIL NO MORE BITS.
100 ;
643B 220500 101 LD (DUR),HL ;SAVE DURATION=TEMPO * TIMING
643E CD4464 102 CALL ENTRY ;PLAY NOTES IN ROUTINE
6441 E1 103 POP HL ;RESTORE MUSIC FILE ADDRESS
6442 18CF 104 JR TOP ;PLAY NEXT SET OF NOTES
105 ;

```



```

106 ;
107 ;*****
108 ;
109 ; LOOK UP INCREMENTS FOR FOUR NOTES IN BYTES 01H - 04H
110 ;
111 ; IF ONE PLACES THE NOTE NUMBERS IN BYTES 1 THRU 4,
112 ; AND A DURATION IN 5 AND 6, THEN ENTERING HERE WILL
113 ; PLAY JUST THAT SINGLE SET OF NOTES.
114 ;
6444 210000 115 ENTRY: LD HL,00H ;INITIALIZE POINTERS TO WAVEF
6447 220800 116 LD (P1),HL
644A 220D00 117 LD (P2),HL
644D 220F00 118 LD (P3),HL
6450 221100 119 LD (P4),HL
6453 210062 120 LD HL,PROG+0100H ;START OF INCREMENT TABLE
6456 ED4B0900 121 LD BC,(TRANS) ;LOAD TRANSPOSE INTO C
122 ;
645A 3A0100 123 LD A,(N1) ;GET FIRST NOTE
645D B7 124 OR A ;SET OR RESET Z FLAG
645E 2803 125 JR Z,E1 ;SKIP TRANSPOSE IF ZERO
6460 81 126 ADD A,C ;ADD TRANSPOSE ADJUSTS UP OR
6461 CB27 127 SLA A ;MULTIPLY BY 2
6463 6F 128 E1: LD L,A ;OFFSET COMPLETES TABLE ADDR
6464 56 129 LD D,(HL) ;GET INCREMENT INTEGER
6465 23 130 INC HL
6466 5E 131 LD E,(HL) ;GET INCREMENT FRACTION
6467 ED531300 132 LD (I1),DE ;SAVE IN FIRST INCREMENT REG
133 ;
646B 3A0200 134 LD A,(N2) ;REPEAT ABOVE CODING FOR EACH
646E B7 135 OR A
646F 2803 136 JR Z,E2
6471 81 137 ADD A,C
6472 CB27 138 SLA A
6474 6F 139 E2: LD L,A
6475 56 140 LD D,(HL)
6476 23 141 INC HL
6477 5E 142 LD E,(HL)
6478 ED531500 143 LD (I2),DE
144 ;
647C 3A0300 145 LD A,(N3)
647F B7 146 OR A
6480 2803 147 JR Z,E3
6482 81 148 ADD A,C
6483 CB27 149 SLA A
6485 6F 150 E3: LD L,A
6486 56 151 LD D,(HL)
6487 23 152 INC HL
6488 5E 153 LD E,(HL)
6489 ED531700 154 LD (I3),DE
155 ;
648D 3A0400 156 LD A,(N4)
6490 B7 157 OR A
6491 2803 158 JR Z,E4
6493 81 159 ADD A,C
6494 CB27 160 SLA A
6496 6F 161 E4: LD L,A
6497 56 162 LD D,(HL)
6498 23 163 INC HL
6499 5E 164 LD E,(HL)
649A ED531900 165 LD (I4),DE

```

```

166 ;
167 ;*****
168 ;
169 ; CREATE COMPOSITE 4 VOICE WAVEFORM AND SEND TO
170 ; PARALLEL PORT. THE NOTES WILL BE PLAYED UNTIL
171 ; DURATION DECREMENTS TO ZERO.
172 ;
649E ED4B0500 173 LD BC,(DUR) ;INITIALIZE DURATION
174 ;NOTE1
64A2 ED5B0B00 175 PLAY: LD DE,(P1) ;GET TABLE POINTER FOR A VOICE
64A6 2A1300 176 LD HL,(I1) ;GET INCREMENT TO ADD TO POINTER
64A9 19 177 ADD HL,DE ;INCREMENT POINTER ADDRESS
64AA 220B00 178 LD (P1),HL ;SAVE NEW ADDRESS
64AD 6C 179 LD L,H ;ADJUST TO LOWER 8 BITS
64AE 2661 180 LD H,HIGH ;ESTABLISH UPPER 8 BITS
64B0 7E 181 LD A,(HL) ;GET FIRST OF SUM
182 ;NOTE2
64B1 ED5B0D00 183 LD DE,(P2) ;REPEAT ABOVE CODING FOR EACH
64B5 2A1500 184 LD HL,(I2)
64B8 19 185 ADD HL,DE
64B9 220D00 186 LD (P2),HL
64BC 6C 187 LD L,H
64BD 2661 188 LD H,HIGH
64BF 86 189 ADD A,(HL) ;SECOND SUMMED
190 ;NOTE3
64C0 ED5B0F00 191 LD DE,(P3)
64C4 2A1700 192 LD HL,(I3)
64C7 19 193 ADD HL,DE
64C8 220F00 194 LD (P3),HL
64CB 6C 195 LD L,H
64CC 2661 196 LD H,HIGH
64CE 86 197 ADD A,(HL) ;THIRD SUMMED
198 ;NOTE4
64CF ED5B1100 199 LD DE,(P4)
64D3 2A1900 200 LD HL,(I4)
64D6 19 201 ADD HL,DE
64D7 221100 202 LD (P4),HL
64DA 6C 203 LD L,H
64DB 2661 204 LD H,HIGH
64DD 86 205 ADD A,(HL) ;FOURTH SUMMED
206 ;SEND TO PORT
64DE D3FF 207 OUT (255),A
64E0 0B 208 DEC BC ;DECREMENT DURATION
64E1 78 209 LD A,B
64E2 B1 210 OR C ;SEE IF B AND C ARE ZERO
64E3 20BD 211 JR NZ,PLAY ;REPEAT UNTIL ZERO
64E5 C9 212 RET ;RETURN FROM ROUTINE
64E6 213 END

```

TOTAL ASSEMBLER ERRORS = 0

DUR	0005	E1	6463	E2	6474	E3	6485
E4	6496	ENTRY	6444	HIGH	0061	I1	0013
I2	0015	I3	0017	I4	0019	LOOP	6431
MULT	0007	MUSIC	1700	N1	0001	N2	0002
N3	0003	N4	0004	NOADD	6436	P1	000B
OG	000D	P3	000F	P4	0011	PLAY	64A2
TRANS	0009	START	6400	TEMPO	0000	TOP	6413

ARRINGTON SOFTWARE SERVICE

Dear Music Owner:

26 December 1980

We sincerely hope that you received the Music System before the Christmas holidays and that it has been a source of delight for you during this festive season. We trust that you found the recent improvements to your liking and that greater customer satisfaction has been achieved.

Please take a moment to make several minor changes to the Music System. The enclosed lines of code will cause the TEMPO to be saved with the song data file when written to cassette tape, and restored when loaded. This will eliminate having to remember a proper tempo for the various songs that you write or purchase. Also, two bugs have been corrected, and the 'ENTER SET #' feature added. Many of the changes are needed to obtain memory space so that these new features could be added without disturbing the machine language routines which follow the Basic program. The lines that have been modified are flagged. Edit your copy of the Music Editor to match the accompanying printout. Thanks.

Now save your updated Music System using the Monitor:

```
>SE T=0                      ;1200 BAUD
>SA MUSIC 1B7 2FFF           ;SAVE MUSIC WITHOUT PIANO PLAYER
or
>SA MUSIC 1B7 37FF           ;SAVE MUSIC WITH PIANO PLAYER
>PP
```

Your old music files, including FANF, BACH, BOOGY, and XMAS, do not contain tempo information. However, by resaving the files after the additions have been made to the editor, the tempo information will be saved with the file. Perform the following steps to add the tempo to the data files recordings:

```
>LO MUSIC
>PP
RUN
PRESS 'LOAD FILE'           ;FOUND ON ALTERNATE KEYPAD.
                             ; READ IN AN OLD DATA FILE.
PRESS 'SET TMPO'            ;ENTER A CORRECT TEMPO BEFORE
                             ; PLAYING OR SAVING THE FILE!!!
PRESS 'SAVE FILE'           ;THE TEMPO SELECTED IS SAVED WITH
                             ; THE NEW RECORDING.
```

The TEMPO information which is stored in address byte 0, is saved with the data file in the byte ahead of the end-of-file flag byte. The end-of-file flag is a zero stored for the timing as part of the five bytes which describe each set. We hope that you are pleased with this latest improvement.

Sincerely

Howard Arrington
Howard Arrington

100 REM
--- MUSIC EDITOR ---
110 REM COPYRIGHT (C) 1980 BY HOWARD ARRINGTON

120 GOTO2180
140 POKE31,R:POKE32,C:X=USR(Q):PRINTA\$;:RETURN
150 REM

[REMOVE LINE 130]

--- MAIN KEYPAD ---

160 FORX=1TO20:NEXTX:GOSUB1000
170 X=USR(K):IFAL=1THEN350
180 A=PEEK(237):IFA=8THENRA=RA-(RA<30):GOSUB560:GOTO170
190 IFA=4THENNA=NA+(NA>0):GOSUB640
200 IFA=2THENSN=SN+(SN>0):GOTO850
210 IFA=16THEN920
220 A=PEEK(238):IFA=8THENNA=NA-(NA<13):GOSUB640
230 IFA=2THENRA=RA+(RA)-2):GOSUB560:GOTO170
240 IFA=16THENGOSUB530
250 IFPEEK(239)=16THEN1810
260 A=PEEK(253):IFA=1THEN960
270 A=PEEK(254):IFA=4THENGOSUB660:GOTO760
280 IFA=1THEN780
290 IFPEEK(239)=8THEN760
300 IFPEEK(236)=8THEN870
310 A=PEEK(252):IFA=4THEN1040
320 IFA=2THENPOKEE+17,0:POKEE+18,LS:X=USR(S)
330 IFA=1THENAL=1:RESTORE1310:GOTO1430
340 GOTO170
350 REM

--- ALTR KEYPAD ---

360 A=PEEK(253):IFA=16THEN2010
370 IFA=8THEN1230
380 IFA=4THEN1120
390 IFA=1THEN1080
400 A=PEEK(237):IFA=2THEN862
410 A=PEEK(254):IFA=16THEN1190
420 IFA=8THEN1610
430 IFA=4THEN1560
440 IFA=2THEN2160
450 IFA=1THENEND
460 IFPEEK(236)=8THEN1770
470 A=PEEK(252):IFA=4THEN1040
480 IFA=2THEN2110
490 IFA=1THENAL=0:RESTORE1450:GOTO1430
500 IFPEEK(239)=16THENSN=SN-(SN<MN):GOTO850
510 IFPEEK(255)=8THENPOKEST+5,0:GOTO2280
520 GOTO170
530 REM

--- KEY > ---

540 KA=KA+1AND3:K=-(KA<>3):F=NOT(KA<>2):S=NOT(KA<>1)

550 POKEOF+KB*2,32:POKEOF+KA*2,210:KB=KA:RETURN
REM

--- NOTE ^ ---

570 IFRA>-1ANDRA<27THENA\$=" " :RG=0:GOTO600
-580 IFRA>26THENRG=-7:A\$=" ABOVE SCREEN RANGE":GOTO600
-590 IFRA<0THENRG=7:A\$=" BELOW SCREEN RANGE"
600 POKERB,32:RB=OH-(RA+RG)*64:POKERB,211
610 Q=INT(RA/7):W=RA-Q*7:Q=Q*12
-620 R=29:C=41:GOSUB140:RETURN
630 REM

--- TIME < ---

640 POKEOG+NB*2,32:POKEOG+NA*2,210:NB=NA:RETURN
650 REM

--- SAVE NOTE ---

660 GOSUB560:A=N(W)+K(W)*K+S+F+Q-5
670 X=A(V):IFXTHENI=PEEK(X-1):POKEX-2,I:POKEX-3,I:POKEX+1,32
680 IFNA=13THENV(V)=0:A(V)=0:A\$="Q " :GOTO730
690 V(V)=A+5:A(V)=RB:A=A-INT(A/12)*12:A\$=N\$(A)+" "
700 IFA=0THENA=11
710 IFK(W)*KORSTHENA\$=N\$(A-1)+"A"
720 IFLEN(A\$)=2THENA\$=A\$+" "
730 R=5:C=47+V*3:GOSUB140:IFV(V)=0THENRETURN
740 A=PEEK(RB-1)*(S=FANDK)-194*(F=-1)-193*(S=1)-195*(S=FANDK=0)
POKERB-2,196+NA:POKERB-3,A:POKERB+1,48+V:RETURN
REM

--- NEXT VOICE ---

770 V=(VAND3)+1:R=5:C=46:A\$=STR\$(V):GOSUB140:GOTO150
780 REM

--- COPY SET ---

790 IFSN=0THENSN=1
800 IFNA=13THENA\$="MOVE R OFF Q FIRST":GOSUB620:GOTO170
810 A=FNA(-5):FORI=1TO4:POKEA+I,V(I):NEXTI
820 POKEA+5,T(NA):R=5:C=47:A\$="1":GOSUB140:V=1:SN=SN+1
-830 X=USR(L):IFSN>MNTHEMN=SN:R=ST+MN*5:POKER,0:POKER-1,PEEK(0)
840 R=0:C=32:A\$=RIGHT\$(STR\$(SN-1),2)+" " :GOSUB140
850 R=7:C=46:A\$=STR\$(SN)+" OF "+STR\$(MN-1)+" " :GOSUB140
860 GOTO150
-862 GOSUB1010:R=11:C=41:A\$="SET #"
-864 POKE32,47:X=USR(Q):INPUTA\$:SN=VAL(A\$):IFSN>MNTHEMN862
-866 GOTO850
870 REM

--- PLAY CHORD ---

880 P=PEEK(9):POKE9,0
890 POKE1,N(W)+K(W)*K+S+F+Q:POKE2,V((VAND3)+1)
900 POKE3,V((V+1AND3)+1):POKE4,V((V+2AND3)+1)
910 POKE5,0:POKE6,7:X=USR(C):POKE9,P:GOTO150
REM

--- KEY SIG ---

930 K(W)=F+S:A=OH-W*64-34+(W(3)*448
940 B=192+KA:IFKA=0ORKA=3THENB=PEEK(A-1)
950 POKEA,B:POKEA-896,B:GOTO150
960 REM

--- SET TEMPO ---

970 GOSUB1010:R=9:C=47:A\$="" :GOSUB140:X=USR(Q)
980 INPUTA\$:A=VAL(A\$):IFA<400RA>166THENGOSUB1020:GOTO970
990 POKE0,A-39:POKEST+MN*5-1,A-39:GOTO150
1000 A\$="" :R=11:C=41:GOSUB140:GOTO620
1010 A\$="Enter # and RETURN":GOTO620
1020 A\$=" ILLEGAL REQUEST " :GOSUB620
1030 FORR=0TO400:NEXTR:RETURN
1040 REM

--- PLAY SCREEN ---

1050 A=SN-9:IFA<1THENA=1
1060 R=ST+A*5-4:POKEE+17,RAND255:POKEE+18,R/256
1070 I=PEEK(R+54):POKER+54,0:X=USR(S):POKER+54,I:GOTO150
1080 REM

--- TRANSPOSE ---

1090 GOSUB1010:R=11:C=41:A\$="TRANSPOSE":GOSUB140
1100 POKE32,51:X=USR(Q):INPUTA\$:IFVAL(A\$)=0AND A\$<>"0"THEN1080
1110 A=VAL(A\$):POKE9,A-256*(A<0):GOTO150
1120 REM

--- REFRAINS ---

1130 GOSUB1010:R=11:C=41:A\$="REFRAIN":GOSUB140
1140 POKE32,49:X=USR(Q):INPUTB
1150 IFB<10RB>MNTHEGOSUB1020:GOTO1130
1160 C=55:A\$="TO " :GOSUB140:INPUTA:R=(A-B+1)*5:C=ST+SN*5-4
1170 IFA<BORA>MN-1THENGOSUB1020:GOTO1130
1180 X=ST+B*5-4:SN=SN+A-B+1:GOTO1220
1190 REM

--- DELETE ---

1200 IFSN=0THENGOSUB1020:GOTO150
1210 R=(MN-SN+1)*5:C=ST+SN*5-4:X=C+5:MN=MN-1
1220 GOSUB1270:X=USR(P):GOTO830
1230 REM

--- INSERT ---

1240 IFSN=0THENGOSUB1020:GOTO150
1250 R=(MN-SN+1)*5:X=ST+MN*5:C=X+5:MN=MN+1
1260 GOSUB1270:X=USR(N):GOTO830
1270 POKE33,RAND255:POKE34,R/256
1280 POKE35,CAND255:POKE36,C/256
1290 POKE37,XAND255:POKE38,X/256:GOTO1000
1300 REM

--- ALTERNATE KEYPAD ---

1310 DATA 13,"STEP"TOGL"INSR"DEL "

```

1320 DATA 14,"CORD"VOIC"SET "SET "
1330 DATA 15," " " " "
1340 DATA 17,"PLAY"COPY"LOAD"SAVE"
1350 DATA 18,"SCRN"RFRN"FILE"FILE"
1360 DATA 19," " " " "
1370 DATA 21,"PIAN"ENTR"SET "SET#"
1380 DATA 22,"PLAY"SET#"ADDR"FAST"
1390 DATA 23," " " " " > "
1400 DATA 25,"ALTR"TRNS" "ERAS"
1410 DATA 26,"KPAD"POSE"EXIT"FILE"
1420 REM

```

--- MAIN KEYPAD ---

```

1430 POKE32,42:FORI=1TO11:READA,A$:POKE31,A:X=USR(Q)
1440 PRINTA$:NEXTI:GOTO150
1450 DATA 13,"PLAY"KEY "NOTE"KEY "
1460 DATA 14,"CORD"SIG " " "
1470 DATA 15," " " " ^ " > "
1480 DATA 17,"PLAY"TIME"SAVE"TIME"
1490 DATA 18,"SCRN" "NOTE" "
1500 DATA 19," " < " " > "
1510 DATA 21,"PLAY"SET#"NOTE"SET#"
1520 DATA 22,"SONG" " " "
1530 DATA 23,"FILE" < " v " > "
1540 DATA 25,"ALTR"NEW "COPY"NEXT"
1550 DATA 26,"KPAD"TMPO"SET "VOIC"
1560 REM

```

--- LOAD FILE ---

```

1570 PRINTCHR$(12):R=10:C=14
1580 A$="Press PLAY on recorder, then type Y":GOSUB140
1590 GOSUB1730:IFA$="Y"THENX=USR(T)
1600 GOTO2280
1610 REM

```

--- SAVE FILE ---

```

1620 PRINTCHR$(12):R=10:C=13
1630 A$="Press RECORD on recorder, then type Y":GOSUB140
1640 GOSUB1730:IFA$(<)"Y"THEN2280
1650 PRINT:INPUT"SAVE FILE NAME ";A$:A$=A$+" "
1660 PRINT:PRINT"SAVING ";A$
1670 FORI=1TO5:POKE10517+I,ASC(MID$(A$,I,1)):NEXTI
1680 A=ST+MN*5+5:B=INT(A/256):I=AAND255
1690 C=(BAND15)+48:IFC>57THENC=C+7
1700 POKE10529,B/16+48:POKE10530,C
1710 D=(IAND15)+48:IFD>57THEND=D+7
1715 I=INT(I/16)+48:IFI>57THENI=I+7
1720 POKE10531,I:POKE10532,D:X=USR(O):GOTO2280
1730 GOSUB1030:R=16:C=0:A$="READY (Y/N) ?? ":GOSUB140
1740 POKE318,195:POKE320,224
1750 A=INP(9):IFA=0THEN1750
1760 A$=CHR$(A):PRINTA$:RETURN
1770 REM

```

--- STEP CHORD ---

```

1780 IFSN=MNTHENGOSUB1020:GOTO150
1790 SN=SN+1:A=FNA(-4):R=PEEK(A+9):POKEA+9,0
1800 POKEE+17,AAND255:POKEE+18,A/256:X=USR(S):POKEA+9,R:GOTO850
1810 REM

```

--- SET ---

```

1820 IFSN=MNTHENGOSUB1020:GOTO150
1830 N=FNA(0):T=PEEK(N+5):SN=SN+1:X=USR(L)
1840 FORNA=0TO13:IFT(NA)<>TTHENNEXTNA
1850 R=0:C=35:A$=RIGHT$(STR$(SN),2)+" ":GOSUB140:O=PEEK(9)
1860 GOSUB640:IFO>100THENO=O-256 - move arrow to current note time
1870 FORV=4TO1STEP-1:B=PEEK(N+V)-5+O:GOSUB1890:NEXTV
1880 KA=KA-1:GOSUB530:V=1:GOTO850
1890 IFB<-4THENB=NA:NA=13:GOSUB660:NA=B:RETURN
1900 RA=PEEK(10540+B):IFRA>200THENRA=RA-256
1910 P=B-INT(B/12)*12:W=RA-INT(RA/7)*7
1920 K=1:F=0:S=0:IFP=20ORP=40ORP=60ORP=90ORP=11THEN1950
1930 IFK(W)<>0THENK=0
1940 GOSUB660:RETURN
1950 IFW=0THEN1990
1960 IFK(W-1)=1THENRA=RA-1:GOTO1940
1970 IFK(W)=-1THEN1940
1980 F=-1:GOTO1940
1990 IFK(6)=1THENRA=RA-1:S=1:GOTO1940
2000 GOTO1970
2010 REM

```

--- TOGGLE VOICES ---

```

2020 A$="VOICE"
2030 FORI=1TO4:IFPEEK(11259+I)THENA$=A$+" ON":GOTO2050
2040 A$=A$+" OFF"
2050 NEXTI:A$=A$+" ":GOSUB620
2060 R=10:C=41:A$="Press RETURN to exit.":GOSUB140
2070 R=11:A$="TOGGLE VOICE $":GOSUB140
2080 POKE32,56:X=USR(Q):INPUTA$:A=VAL(A$)
2090 IFA>0ANDA<5THENPOKE11259+A,255-PEEK(11259+A):GOTO2010
2100 GOSUB1000:R=10:GOSUB140:GOTO150
2110 REM

```

--- PIANO PLAYER ---

```

2120 IFPEEK(14003)=65ANDPEEK(12289)=245THEN2140
2130 A$=" NOT INSTALLED":GOSUB620:GOSUB1030:GOTO150
2140 POKEE+17,0:POKEE+18,LS:X=USR(I)
2150 POKEE+63,68:POKEE+64,45:GOTO2280
2160 REM

```

--- SET ADDRESS ---

```

2170 A$=" ADDRESS =" +STR$(FNA(-4)):GOSUB620:GOTO170
2180 REM

```

--- INITIALIZE ---

```

2190 PRINTCHR$(12):CLEAR400:POKE260,0:POKE261,32
2200 DEFFNA(L)=ST+SN*5+L

```

0000 = 3D 3D35 = 3D 3803 = 3D 36F4 = 45 4E5452

530 Rem

560 Rem

835 For J I = 1 to 4: A = A(I) : IF A = 0 then Next: Goto 840

836 Poke A-3, Peek(A-1) : Poke A-2, Peek(A-1) : Next

1540 change 'Copy' to 'ENTR'


```

2210 ST=14558:LE=11520:LS=517258:ST=ST-1
2220 DIMN$(11):RESTORE2230:FORI=0TO11:READN$(I):NEXTI
2230 DATA E,F,GASC,G,AASC,A,BASC,B,C,DASC,D,EASC
2240 DIMT(13):FORI=0TO13:READT(I):NEXTI
2250 DATA 1,2,3,4,6,8,12,16,24,32,48,64,128,0
2260 DIMN(6),K(6):FORI=0TO6:READN(I):NEXTI
2270 DATA 5,6,8,10,12,13,15:OF=-2108:OG=-2099:OH=-2266
2280 PRINTCHR$(12):X=USR(G)
2290 FORMN=1TO15000:IFPEEK(ST+MN*5)THENNEXTMN
2300 FORI=1TO4:V(I)=0:POKE11259+I,255:NEXTI
2310 K=1:V=1:Q=12:W=5:POKE9,0:SN=MN:POKE0,PEEK(FNA(-1))
2320 R=9:C=48:A$=STR$(PEEK(0)+39)+"" :GOSUB 140
2330 AL=0:RA=12:RB=-3034:NB=5:NA=5:GOSUB1000:GOTO850
READY

```

ARRINGTON SOFTWARE SERVICE

SPACE TREK -- Use the Monitor's)LOG command to load this Basic program and its attendant machine language routines. The program will auto-execute by jumping to address C85BH. The program's sound effects can be heard by connecting an amplifier to any output bit and ground of the parallel port. It is recommended that one use the music board from our Music System to interface your amplifier to the parallel port.

As captain of a starship, your mission is to seek and destroy a fleet of enemy starships in a given number of stardates and with a limited number of starbases for resupplying and repairing your starship. The galaxy consists of 64 quadrants, each divided into an 8 by 8 section grid.

There are nine commands, each keyed as three letters such as 'HEL' for HELP, which will list the commands. The starship's four computer options assist in course computations, etc. You will be pleased by the controlled use of portions of the screen to display the scanners, status and workings of your SPACE TREK starship. Requires 32K or more.

BRUCE -- So, you've chosen to sharpen your chess abilities by playing BRUCE. Bruce is loaded using the Monitor's)LOG command which will auto-execute with a jump to address 63CH. You may play either black or white, as well as select the depth of analysis level by entering a digit to the PLYS question. I prefer level '3' or '4' for a good game. Moves are coded as, for example, 'D2D4', which moves the piece on column 'D', row 2, to square 'D4'. The board and playing pieces are displayed using the Sorcerer's high resolution graphic characters.

One may Castle, and capture En Passant. Bruce, however, does not allow one to configure the board pieces for analysis or replay purposes. Some have asked if Bruce is as good as SARGON I or II. I believe that SARGON is better because it has had a couple years to gain more maturity. Make no mistake, though, Bruce will provide you with a challenging game and hours of enjoyment.

HORSE RACE -- Gambling on a horse race is always fun when you don't have to cover your losses. This graphical animation is additionally delightful because to the tunes played through the parallel port's output bits. It is recommended that one use the music board from our Music System to interface your amplifier to the Sorcerer. Answer 'NO' to the first question, and then enter the player names and bets. To send the high resolution horses on their race across the screen, press RETURN when another player name is requested. You'll enjoy the 'Lone Ranger' and 'If I Were a Rich Man' melodies. Use the Monitor's)LOG command to load and auto-execute by jumping to C85BH.

JAIL BREAKOUT -- This reflex game also contains sound, and it is recommended that one use the music board from the Music System for interfacing an amplifier to the output bits of the parallel port. Use the Monitor's)LOG command to load and begin execution at address 100H. The left and right cursor keys move the game paddle back and forth, as one tries to keep a ball in play. The ball knocks out bricks from the wall at the top of the screen with each hit. To place a ball in motion, press 'HOME', ie. the '5' key on the keypad. This game is similar to the TV arcade game.

Send orders to: ARRINGTON SOFTWARE SERVICE
9522 LINSTOCK
BOISE, IDAHO 83704
208-377-1938 after 6 p.m. Mountain

[] SPACE TREK	\$15.00	Basic game with ML routines.
[] BRUCE	\$20.00	Machine language game.
[] HORSE RACE	\$10.00	Basic game with ML routines.
[] JAIL BREAKOUT	\$10.00	Machine language game.