

Rev.A Sept. 1977

I. General

II. Theory of Operation

- A. Address decoding
- B. Data Latching
- C. Digital to Analog Conversion (DAC)
- D. Audio Amplifier
- E. Power Supply

III. Users Guide

- A. External Speakers
- B. Use with Audio System

IV. Software

- A. General
- B. SCORE68 and PLAY68
- C. Test program RAMP
- D. PHASOR
- E. NOISE
- F. Cassette Software
- H. Program Listings

V. Schematic and Parts List

VI. Warranty and Warranty Registration Form

Copyright (C) 1977 Newtech Computer Systems, Inc.
ALL RIGHTS RESERVED

I. General

The Model 68 Music Board was brought out in response to demand by SWTPC computer system owners for a music board similar to our S-100 bus compatible Model 6 Music Board.

The Model 68 has many possible applications: generating sound effects for computer games; playing along while you play your guitar or clarinet or trumpet; simulating train sounds for your model railroad; generating audible prompts from your computer while running other programs. The list goes on and on. Newtech welcomes your ideas and suggestions as well, and we'll be happy to pass them on to users, perhaps through the SWTPC newsletter.

The programs in this manual are a good introduction to computer-generated sound. The user can modify them as he learns how they work.

For those wishing to go beyond these programs, there is a growing body of literature on computer-generated music in the personal computing magazines. Although most of the articles are for processors other than the 6800, they are good reading for developing an appreciation of the general principles involved in computer music. Many of the programs offered can be readily translated for the 6800. To name just a few:

1. Wright, Malcolm, "Alphanumeric Music with Amplitude Control" (available from People's Computer Company, Menlo Park, California).
2. Chamberlain, Hal, "Computer Bits" in "Popular Electronics", vol.10#3, 116-119 (Sept.1976) and vol.10#4, 88-91 (Oct.1976).
3. Chamberlain, Hal, "A Sampling of Techniques for Computer Performance of Music" in "Byte", Vol.2#9, 62-83 (Sept.1977).
4. Siegel, Dorothy, "Scott Joplin on your Sci-Fi Hi-Fi", in "ROM", vol.1#4, 61-65 (Oct.1977) and "A Musical Smorgasboard", in "ROM", vol.1#5 (Nov.1977).

The Popular Electronics articles contain useful programs for Touch-tone synthesis and "weird" sound generation. The ROM articles contain a description of simplified versions of Newtech's Model 6 software, suggestions on how to modify them, and pre-coded listings for other music.

For those who wish to delve more deeply into the "roots" of computer music and the fundamentals of sound generation, the following may be of interest:

1. Helmholtz, H.L.F., On the Sensations of Tone as a Physiological Basis for the Theory of Music (Dover, New York, 1954).
2. Mathews, Max V., The Technology of Computer Music (M.I.T. Press, Cambridge, Mass., 1969).

II. THEORY OF OPERATION

A. Address Decoding:

The Model 68 Music Board is designed for use with the Southwest Technical Products Corporation 6800 computer system. It will work equally well at any I/O port address so long as the software used conforms to the slot selected. The programs provided herein use I/O slot 4 (address 8010H) but can be readily modified for a different I/O assignment. A Triple 3-Input NAND Gate, IC1, is used to derive a strobe from I/O#, / ϕ 2 and R/W.

B. Data Latching:

When an instruction to write into the Model 68 I/O slot memory location is performed, the write data will appear on the data bus lines D0 through D7. The most significant 6 bits of this data, D7 through D2, will be latched by the Hex latch IC2.

C. Digital to Analog Conversion (DAC):

The data latched in IC2 is buffered by CMOS buffers. These buffers drive an R/2R ladder network with voltages very close to 0 volts and the +5 volt supply. The ladder network employs precision resistors and is minimally loaded by a high impedance voltage divider into the audio amplifier.

D. Audio Amplifier:

The analog output from the R/2R DAC ladder network is AC coupled through capacitor C7 and resistive divider network R14, R13 to an audio amplifier IC4. The output low pass filter R15 and C5 and capacitors C1 and C2 on the amplifier 8 volt supply, are required to maintain amplifier stability. The output of the R/2R ladder network can swing from 0 to 5 volts. The input to IC4 can thus range from near zero to $5 \text{ volts} \times 25\text{K}/(2\text{M}+25\text{K}) \approx .062$ volts peak to peak. The gain of IC4 is about 50 so its output will swing about 3.08 volts peak to peak. For a maximum amplitude square wave, output will thus be $3.08/2 \approx 1.54$ VRMS. The maximum direct power output to an 8 Ohm speaker will thus be $(1.54)^2/8 \approx .3$ Watt and peak speaker load current will be $V/R = 1.54/8 = .19$ Ampere.

This rather detailed explanation of the audio amplifier section has been given for two reasons. The first is to make its operation easy to understand. The second is that it is important to understand the limitations of the Model 68 and what it will and will not do. It WILL drive the 2" speaker on the Model 68 Music Board or, alternatively, a small external efficient 8 Ohm speaker. Of course, the Model 68 can be used as an input to an audio system. One of the things the Model 68 will NOT do (nor for that matter will any other music board designed to run off a hobbyist computer power supply), is replace a home high fidelity system power amplifier to drive large inefficient speaker systems.

The Model 68 Music Board audio amplifier IC and output capacitor C8 are conservatively rated to operate as supplied. But DO NOT ATTEMPT TO SOUP-UP THIS CIRCUIT TO PROVIDE HIGHER POWER OUTPUT. It would be dangerous, as the components are not rated for the increased stresses.

E. Power Supply:

The audio amplifier works off the SWTPC 8 volt unregulated supply. Capacitors C1 and C2 are required for improved high frequency supply regulation and for stability of the audio amplifier IC4.

IC5 inputs +8 volts unregulated from the SWTPC I/O bus and provides 5 volts regulated for the logic and DAC circuitry. Capacitor C2 on the regulator input is required for regulator stability and capacitors C3 and C4 for improved high frequency regulation.

III. USERS GUIDE

A. External Speakers:

Considerable improvement in sound quality can be readily achieved over the 2" speaker supplied with the Model 68 by using an external speaker.

When this is done, jumper JMPR1 should be removed (or cut) to disconnect the on-board speaker. The external speaker can be connected at J1. Excellent sound production has been achieved using very inexpensive 5" speakers.

B. Use with Audio System:

Best sound production is achieved when the Model 68 is connected to a high fidelity amplifier. An auxiliary input jack is likely your best choice.

CAUTION: The Model 68 is capable of producing a very wide range of frequencies extending from the subaudible through the entire audio range and up to the ultrasonic. You should use the common sense you would use with any other signal source, such as a record turntable or radio. Play too loud, particularly on the bass drive, and you can damage your speaker.

Additionally, your software can cause subaudible or ultrasonic range outputs from the Model 68 which you can only see on an oscilloscope and yet may be driving your speaker system. So, a good rule of thumb would be to first set up your amplifier and Model 68 volume controls for the test program given herein, or for some other program known to be in the audio range, and then NEVER substantially exceed those amplifier settings.

IV. SOFTWARE

A. General:

The software supplied herein represents a very small fraction of the applications possible for the Model 68. As a matter of fact, books devoted entirely to the subject of microcomputer music will undoubtedly appear in the near future. The intent here is to supply the Model 68 user with basic music and sound effects software. The serious student of computer music will find ways to elaborate on these routines and to create others himself.

One major goal of the software as written is to make the programs readily understandable. Readers may want to rewrite the programs with other goals in mind: minimizing memory requirements; maximizing execution speed; adding other features, etc..

B. SCORE68 and PLAY68:

The two music programs contained herein are named SCORE68 and PLAY68. SCORE68 is a BASIC program that uses alphanumeric notations of musical notes to produce a table of constants which, in turn, is used by the Assembly Language PLAY68 program to "play" the music. The calculation of constants is compatible with the SWTPC 0.8986MHz 6800 clock.

SCORE68 was written in SWTPC 8K BASIC (Rev. 2.0) and may require modification if you are using a different BASIC.

The music used in SCORE68 is "The Entertainer" by Scott Joplin. Other music can be coded and inserted as DATA statements instead of those used for the Joplin piece.

Each note in the music is specified by a four- or five-character string appearing as a DATA statement in SCORE68. For example, the first note of the music, a sixteenth-note "D", is specified as "Db3S".

Pitch is specified by the first three characters:

First character: "A", "B", "C", "D", "E", "F", "G"

Indicates pitch within an octave.

Second character: "#" (sharp), "!" (flat), " " (blank, or natural)

Indicates whether the pitch is raised or lowered (or neither) by a half-step.

Third character: "1", "2", "3"

Indicates the octave, with "1" the lowest of the three octaves starting at A=220Hz, "2" the second octave starting at A=440Hz, and "3" the third octave starting at A=880Hz. N.B. PLAY68 as written does not allow the use of the notes below C in octave 1.

Duration is specified by the fourth and fifth characters:

Fourth character: "S"(sixteenth), "E"(eighth), "Q"(quarter),
"H"(half), "W"(whole)

Indicates relative length of each pitch.

Fifth character (optional): ".."

Indicates increasing by half the duration of the note
specified by the fourth character.

Thus, for example, the first six notes of "The Entertainer" used in SCORE68:



are coded as:

0830 DATA D 3S,E 3S,C 3S,A 3E,B 3S,G 2E

SCORE68 has a special end-of-score one-character string "X" which enters a zero pitch constant in the score to indicate the end of the music.

Tempo can be increased or decreased by changing the constant K6. The speed is proportional to K6, so doubling K6 would double the speed of the music.

Illegal characters in a note specification string will produce an error message and terminate score compilation. The error message would be of the form:

ERROR IN NOTE #1
DATA STRING "A#3P."
CHARACTER #4

In this example the fourth character is illegal. When the score is completely compiled without error messages, "SCORE COMPILED COMPLETE" will be indicated.

The starting location of the score produced is specified by the variable U. Standard location is 2C00H (specified as 11264 in BASIC) but can be changed to any free memory location, provided the score location pointer in PLAY68 is made to agree.

Program execution starts at the first statement -- requiring you to simply type "RUN".

Roughly six to eight bytes of memory are required in the BASIC program for each note of music specified. PLAY68 requires three bytes for each note of music.

C. RAMP:

RAMP is a test routine for verifying proper operation of the Model 68 with the user's computer. Note that the Newtech Model 68 standard output port address is for I/O slot 4 (8010H). The Model 68 will work equally well in any other I/O slot so long as its driving software is made to agree with its address. Note also that RAMP is relocatable and can be run almost anywhere in memory. If there is ever doubt that the Model 68 is working the RAMP routine can be quickly entered and executed using the MIKBUG* or other monitor program. Since RAMP loops forever, the user must reset or interrupt his computer to exit RAMP.

RAMP produces a 319Hz linear ramp of 64 discrete steps. (Note that the program outputs 256 values; however there are only $2^6 = 64$ values output from the Model 68 since its 6-bit DAC doesn't use the two least significant bits.) The linearity of the Model 68 can be verified on an oscilloscope by observing the monotonically increasing stepped waveform at the output of phono jack J1.

The ramp frequency is calculated as follows:

- * SWTPC 6800 clock frequency = $1.7971/2 = 0.89855$ MHz
Each processor state is thus $T = 1/F = 1/0.89855 \times 10 = 1.11\mu s$
- * Each pass through the program takes 11 processor states.
- * Each ramp takes 256 passes through the program.
- * The total ramp time is thus:
$$T_R = (1.11\mu s/\text{state}) \times (11 \text{ states/pass}) \times (256 \text{ passes/ramp})$$

$$T_R = 3134 \text{ us/ramp}$$

The ramp frequency $F_R = 1/T_R = 319\text{Hz}$.

D. Phasor:

PHASOR is a sound effect program demonstrating one way to generate frequency sweeps. As written, the default parameter values in PHASOR will generate 13 rapid sweeps from very high to very low frequencies and then exit by jumping to the MIKBUG* monitor. The user can change the number of sweeps, starting sweep frequency, ending sweep frequency, sweep rate, and amplitude by changing, respectively, SWEEPN, FIRSTF, LASTF, RATE, and TOGGLE. Very interesting effects can be attained with a main control program setting these parameters and using PHASOR as a subroutine.

* MIKBUG is a registered Motorola trademark.

E. NOISE

NOISE is a sound effect program based upon digital low-pass filtering of digitally generated white noise.

The RNDM subroutine within NOISE implements a 16-bit maximal length pseudorandom number generator. Each time it is called, the most significant byte of the number generated is returned in Accumulator A. The main routine in NOISE uses table TBL to shape the amplitude envelope of the noise. By ANDing the noise generated, with a mask from TBL, only certain bits are allowed to change. For example, while ANDing with mask FFH allows all bits to change and thus maximum amplitude noise to be produced, ANDing with mask 3FH will keep the two most significant bits (A7 and A6) constant and thus allow only lower noise levels to be produced.

The DURA parameter in NOISE determines the number of noise samples used for each segment in the amplitude envelope specification TBL. Thus decreasing DURA will shorten the duration of the sound. The FREQ parameter in NOISE determines the time between random noise samples and thus the frequency band of the noise generated.

A great variety of sound effects can be achieved by changing the parameters in NOISE. By calling NOISE as a subroutine, a main program can vary the NOISE parameters dynamically. NOISE has been used to simulate the sound of locomotives, guns, snare drums, rockets, etc.

Model 68 Software Cassette
Rev. A Sept. 1977

TAPE COUNTER ¹	FILE NAME	FILE TYPE	MEMORY REQUIREMENTS	COMMENTS
0-22/28	STING	MIKBUG ²	2B00H-2DFFH	STARTS 2B00H
30/38-48	PHASOR	MIKBUG	100H-1FFH	STARTS 100H
50/55-63	NOISE	MIKBUG	100H-1FFH	STARTS 100H
65/73-121	SCORE68 ³	BASIC	12K INCLUDING SWTPC 8K CASSETTE BASIC	16K FOR SWTPC DISK 8K BASIC
130/135-180	PLAY68 ³	CO-RES SOURCE	12K INCLUDING CASSETTE CO-RES ASSEMBLER	16K FOR SWTPC DISK SYSTEM
180/185-212	PHASOR	"	"	"
212/217-244	NOISE	"	"	"

(1) The tape counter designation 30/38-48 indicates that on Newtech's tape position counter an identifying voice track runs from 30 to 38 and the actual file runs from 38 to 48.

(2) MIKBUG file types are MIKBUG formated binary object tapes.

MIKBUG is a registered trademark of Motorola Inc.

(3) Users of non-SWTPC cassette based software and users writing very large music scores that use more memory will want to change SCORE68 and PLAY68 pointers referring to the location of PLAY68 and the musical score constants.

```
0010 REM SCORE68 REV.A SEPT. 1977
0020 REM COPYRIGHT (C) 1977
0030 REM NEWTECH COMPUTER SYSTEMS, INC.
0040 REM ALL RIGHTS RESERVED
0050 REM
0150 LET U=11264 : REM U DEFINES SCORE AREA IN MEMORY
0160 LET K1=2^(1/12)
0170 LET K6=1.2 :REM TEMPO CONTROL
0190 FOR V=1 TO 1000
0200 LET C=1
0210 READ Z$
0220 LET N=100
0225 LET A$=MID$(Z$,1,1)
0230 IF A$="A" THEN N=1
0240 IF A$="B" THEN N=3
0250 IF A$="C" THEN N=4
0260 IF A$="D" THEN N=6
0270 IF A$="E" THEN N=8
0280 IF A$="F" THEN N=9
0290 IF A$="G" THEN N=11
0300 IF A$="X" THEN GOTO 720
0310 IF N=100 THEN GOTO 760
0320 LET C=2
0330 LET M=100
0335 LET A$=MID$(Z$,2,1)
0340 IF A$="!" THEN M=N-1
0350 IF A$="#" THEN M=N+1
0360 IF A$=" " THEN M=N
0370 IF M=100 THEN GOTO 760
0380 LET C=3
0390 LET P=100
0395 LET A$=MID$(Z$,3,1)
0400 IF A$="1" THEN P=M
0410 IF A$="2" THEN P=M+12
0420 IF A$="3" THEN P=M+24
0430 IF P=100 THEN GOTO 760
0440 LET C=4
0450 LET T=100
0455 LET A$=MID$(Z$,4,1)
0460 IF A$="S" THEN T=16
0470 IF A$="E" THEN T=8
0480 IF A$="Q" THEN T=4
0490 IF A$="H" THEN T=2
0500 IF A$="W" THEN T=1
0510 IF T=100 THEN GOTO 760
0530 LET C=5
0540 IF MID$(Z$,5,1)=". " THEN T=2*T/3
```

```

0550 REM CALCULATE CONSTANTS
0560 LET F1=220*(K1^(P-1))
0570 LET T1=10^6/(2*F1)
0580 LET K3=(T1*1.7971/2 -185)/6
0590 LET K4=F1/(K6*T)
0600 LET D3=INT(K4) : REM MAKE DURATION EVEN#
0610 LET D4=2*D3-2*INT(D3/2)
0620 LET D5=INT(D4/256) : REM CALC. 2 BYTES
0640 LET D7=D4-D5*256 : REM D7=LSB
0650 REM TRANSFER CONSTANTS TO SCORE AREA
0660 POKE( U+3*(V-1),INT(K3+.5))
0670 POKE( U+3*(V-1)+1,D5)
0680 POKE( U+3*(V-1)+2,D7)
0690 PRINT V;
0700 NEXT V
0710 STOP
0720 POKE( U+3*(V-1),0)
0730 PRINT
0740 PRINT "SCORE COMPILATION COMPLETE"
0750 STOP
0760 PRINT "ERROR IN NOTE #";V
0770 PRINT "DATA STRING ";Z$
0780 PRINT "CHARACTER #";C
0790 STOP
0810 REM
0820 REM "THE ENTERTAINER" BY SCOTT JOPLIN
0830 DATA D 3S,E 3S,C 3S,A 3E,B 3S,G 2E
0840 DATA D 2S,E 2S,C 2S,A 2E,B 2S,A 2S,A!2S
0850 DATA G 1Q,G 3E,D 1S,D#1S
0860 DATA E 1S,C 2E,E 1S,C 2E,E 1S,C 2Q.
0870 DATA C 3S,D 3S,D#3S
0880 DATA E 3S,C 3S,D 3S,E 3E
0890 DATA B 3S,D 3E
0900 DATA C 3Q.,D 1S,D#1S
0910 DATA E 1S,C 2E,E 1S,C 2E,E 1S,C 2Q.
0920 DATA C 2S,A 3S,G 2S
0930 DATA F#2S,A 3S,C 3S,E 3E,D 3S,C 3S,A 3S
0940 DATA D 3Q.,D 1S,D#1S
0950 DATA E 1S,C 2E,E 1S,C 2E,E 1S,C 2Q.
0960 DATA C 3S,D 3S,D#3S
0970 DATA E 3S,C 3S,D 3S,E 3E,B 3S,D 3E
0980 DATA C 3Q.,C 3S,D 3S
0990 DATA E 3S,C 3S,D 3S,E 3E,C 3S,D 3S,C 3S
1000 DATA E 3S,C 3S,D 3S,E 3E,C 3S,D 3S,C 3S
1010 DATA E 3S,C 3S,D 3S,E 3E,B 3S,D 3E
1020 DATA C 3Q,C 3S,E 2S,F 2S,F#2S
1030 DATA G 2E,A 3S,G 2E,E 2S,F 2S,F#2S
1040 DATA G 2E,A 3S,G 2E,E 2S,C 2S,G 1S
1050 DATA A 2S,B 2S,C 2S,D 2S,E 2S,D 2S,C 2S,D 2S
1060 DATA C 2E,X
9000 END

```

00100 NAM PLAY68 REV.A SEPT.1977
 00110 OPT 0,NOP
 00120 * COPYRIGHT (C) 1977 ALL RIGHTS RESERVED
 00130 * NEWTECH COMPUTER SYSTEMS, INC.
 00140 *
 00150 * PLAY68 STARTS AT THE BEGINNING OF THE MEMORY AREA
 00160 * DESIGNATED "SCORE" AND TRANSFERS INTO RAM LOCATION
 00170 * "PITCH" A 1-BYTE PITCH PARAMETER AND INTO RAM
 00180 * LOCATION "DURA" A 2-BYTE DURATION PARAMETER. THE
 00190 * ROUTINE THEN CALLS SUBROUTINE PLAY TO OUTPUT TO
 00200 * THE MODEL 68 THE MUSICAL NOTE SPECIFIED BY THESE
 00210 * NOTE PARAMETERS. PLAY68 CONTINUES TRANSFERRING
 00220 * NOTE PARAMETERS AND CALLING THE PLAY ROUTINE UNTIL
 00230 * A PITCH CONSTANT OF ZERO IS ENCOUNTERED WHICH
 00240 * INDICATES THE END OF THE MUSICAL SCORE.
 00250 *

00260	2B00	ORG	\$2B00	
00270	2C00	SCORE	EQU \$2C00	SCORE AREA CAN BE CHANGED.
00280	2B00	8E	2B90 LDS #STACK	INIT. STACK POINTER.
00290	2B03	CE	2C00 LDX #SCORE	INIT. SCORE POINTER.
00300	2B06	FF	2B77 STX PLACE	
00310	2B09	FE	2B77 NEXT LDX PLACE	
00320	2B0C	86	00 LDA A #0	IF END OF SCORE LOOP HERE.
00330	2B0E	A1	00 CMP A X	
00340	2B10	27	FE HERE BEQ HERE YOUR ENDING?	
00350	*		ELSE TRANSFER PARAMETERS FOR NEXT NOTE OR SCORE	
00360	*		INTO PLAY ROUTINE.	
00370	2B12	A6	00 LDA A X LOAD PITCH.	
00380	2B14	B7	2B7B STA A PITCH	
00390	2B17	08	INX	
00400	2B18	A6	00 LDA A X LOAD DURATION MSB	
00410	2B1A	B7	2B75 STA A DURA	
00420	2B1D	08	INX	
00430	2B1E	A6	00 LDA A X LOAD DURATION LSB.	
00440	2B20	B7	2B76 STA A DURA+1	
00450	2B23	08	INX	
00460	2B24	FF	2B77 STX PLACE SAVE SCORE POINTER.	
00470	2B27	8D	02 BSR PLAY CALL NOTE PLAYING ROUTINE.	
00480	2B29	20	DE BRA NEXT GO DO NEXT NOTE.	
00490	*	THE PLAY	ROUTINE PLAYS ONE NOTE & RETURNS.	
00500	2B2B	CE	2B6C PLAY LDX #TBL1 INIT. ENVELOPE POINTER.	
00510	2B2E	FF	2B79 STX TBL1P STORE ENV. POINTER.	
00520	2B31	E6	00 LDA B X PUT AMPLITUDE VALUE IN B.	
00530	2B33	FE	2B75 LDX DURA LOAD DURATION PARAMETER	
00540	*		INTO INDEX REGISTER.	

00550	2B36	BC	E000	LOOP3	CPX	\$E000	5-WASTE TIME (31 STATES)
00560	2B39	BC	E000		CPX	\$E000	5-
00565	2B3C	BC	E000		CPX	\$E000	5-
00570	2B3F	BC	E000		CPX	\$E000	5-
00580	2B42	BC	E000		CPX	\$E000	5-
00590	2B45	73	E000		COM	\$E000	6-
00600	2B48	86	16	LOOP2	LDA A	#22	4-FIXED DELAY TO ADJUST
00610	2B4A	4A		LOOP4	DEC A		2- LOWEST NOTE TO 262HZ
00620	2B4B	26	FD		BNE	LOOP4	4- (MIDDLE C) WHEN PITCH
00630		*					PARAMETER=FE.
00640	2B4D	B6	2B7B		LDA A	PITCH	4-LOAD PITCH PARAMETER.
00650	2B50	F7	8010		STA B	MOD68	5-OUTPUT TO MUSIC BOARD.
00660	2B53	4A		LOOP1	DEC A		2-DELAY AS PER PITCH PARAM.
00670	2B54	26	FD		BNE	LOOP1	4-
00680	2B56	53			COM B		2-COMPLEMENT WAVEFORM VALUE.
00690	2B57	09			DEX		4-DECREMENT DURATION COUNTER.
00700	2B58	26	DC		BNE	LOOP3	4-
00710	2B5A	7C	2B7A		INC	TBL1P+1	6-SET UP NEXT SEGMENT.
00720	2B5D	FE	2B79		LDX	TBL1P	5-
00730	2B60	E6	00		LDA B	X	5-
00740	2B62	C1	01		CMP B	#\$01	2-END OF ENVELOPE CHAR.=01.
00750	2B64	27	05		BEQ	RETRN	4-RETURN IF LAST SEGMENT.
00760	2B66	FE	2B75		LDX	DURA	5-RESET DURATION PARAMETER.
00770	2B69	20	DD		BRA	LOOP2	4-
00780	2B6B	39		RETRN	RTS		
00790		*					
00800		*					* AMPLITUDE ENVELOPE SPECIFICATION:
00810		*					* MAXIMUM AMPLITUDE IS OUTPUT WHEN ACCUMULATOR B IS
00820		*					* COMPLEMENTED FROM 00 TO FF AND BACK. MINIMUM
00830		*					* AMPLITUDE IS OUTPUT WHEN B IS COMPLEMENTED
00840		*					* BETWEEN 80 AND 7F. AN END OF ENVELOPE RECORD
00850		*					* OF \$01 MARKS THE END OF THE SPECIFICATION.
00860		*					
00870	2B6C	FF		TBL1	FCB	\$FF,\$FF,\$F0,\$E0,\$D0,\$C0,\$B0,\$85,\$01	
	2B6D	FF					
	2B6E	F0					
	2B6F	E0					
	2B70	D0					
	2B71	C0					
	2B72	B0					
	2B73	85					
	2B74	01					
00880		*					
00890	2B75	0002		DURA	RMB	2	DURATION CONSTANT.
00900		8010		MOD68	EQU	\$8010	MUSIC BOARD IN I/O SLOT 4.
00910	2B77	0002		PLACE	RMB	2	
00920	2B79	0002		TBL1P	RMB	2	TABLE POINTER
00930	2B7B	0001		PITCH	RMB	1	PITCH PARAMETER.
00940	2B7C	0014			RMB	20	
00950		2B90		STACK	EQU	*	
00960					END		

00100 NAM RAMP REV.A SEPT.1977
00110 OPT NOP,O
00120 * MODEL 68 MUSIC BOARD TEST ROUTINE.
00130 * THIS ROUTINE PRODUCES A TRIANGULAR RAMP OF 319 HZ
00140 * ON A 6800 COMPUTER SYSTEM RUNNING AT 0.89855 MHZ
00150 8010 MOD68 EQU \$8010 MUSIC BOARD IN I/O SLOT 4
00160 * CAN BE CHANGED FOR OTHER SLOTS
00170 0100 ORG \$0100
00180 0100 4C RAMP INC A 2-CALCULATE NEXT STEP.
00190 0101 B7 8010 STA A MOD68 5-OUTPUT TO MUSIC BOARD.
00200 0104 20 FA BRA RAMP 4-LOOP
00210 END

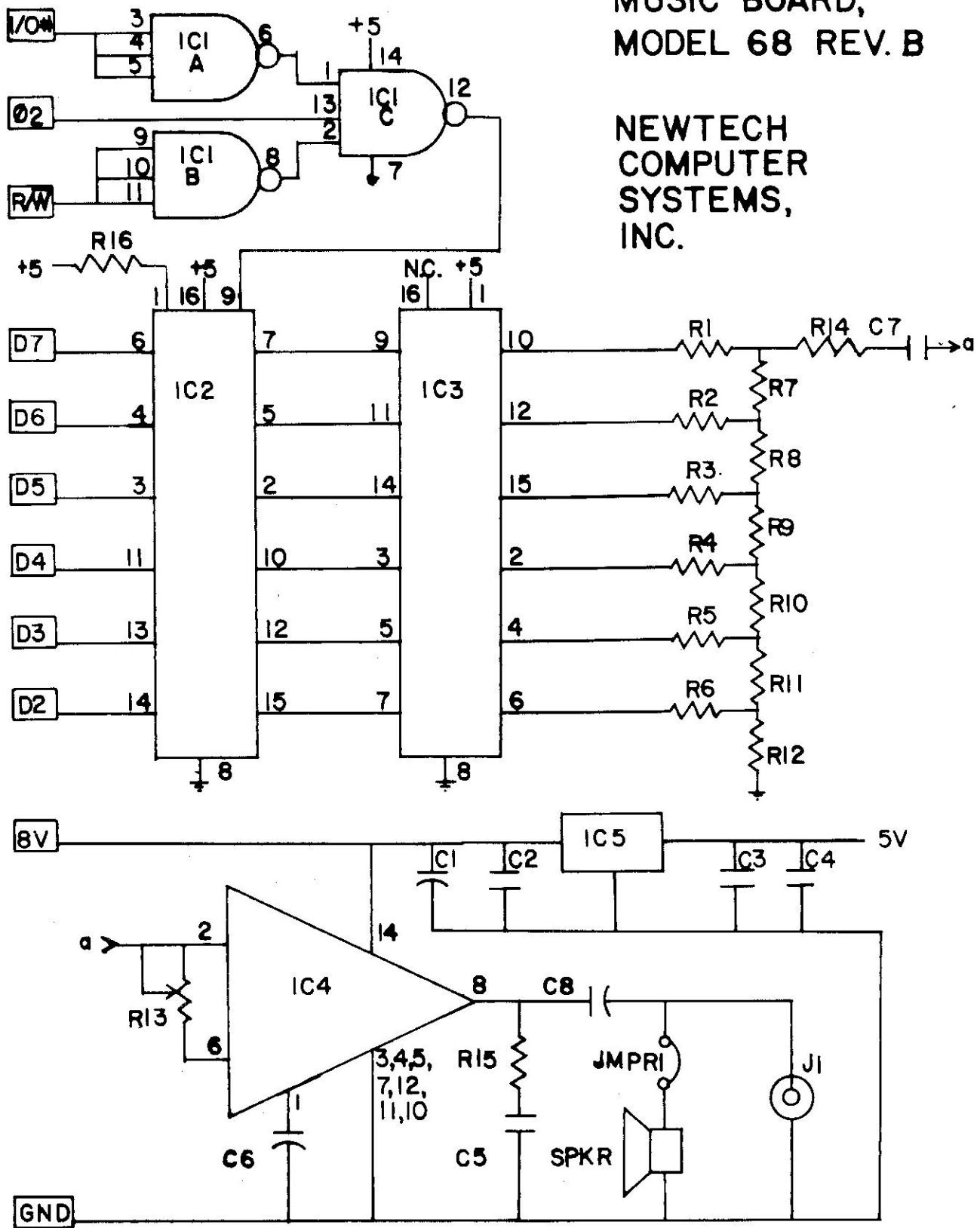
00100		NAM	PHASOR	REV.A SEPTEMBER 1977	
00110		OPT	NOP,O		
00120	* COPYRIGHT (C) ALL RIGHTS RESERVED				
00130	* NEWTECH COMPUTER SYSTEMS, INC.				
00140	*				
00150 0100		ORG	\$0100		
00160 0100 B6 0145	START	LDA A	SWEEPN	INIT. SWEEP COUNT	
00170 0103 B7 0146		STA A	SWEEPS		
00180 0106 86 00	NEXTS	LDA A	#0	EXIT IF SWEEPS COUNT=0.	
00190 0108 B1 0146		CMP A	SWEEPS	ELSE DECREMENT COUNT &	
00200 010B 27 35		BEQ	EXIT	DO ONE SWEEP.	
00210 010D 7A 0146		DEC	SWEEPS		
00220 0110 B6 0147		LDA A	FIRSTF	INIT.FREQ.PARAMETER.	
00230 0113 B7 0149		STA A	FREQ		
00240 0116 B6 014A		LDA A	RATE	GET SWEEP RATE PARAM.	
00250 0119 4A	LOOP3	DEC A		DECREMENT IT.	
00260 011A 26 10		BNE	LOOP1	IF N.E.0 BRANCH.	
00270 011C 7C 0149		INC	FREQ	ELSE DECREASE FREQ.	
00280 011F B6 0148		LDA A	LASTF	IF LOWEST FREQUENCY	
00290 0122 B1 0149		CMP A	FREQ	THEN SWEEP IS DONE SO	
00300 0125 27 DF		BEQ	NEXTS	GO NEXT SWEEP.	
00310 0127 B6 014A		LDA A	RATE	ELSE RESTORE RATE PARAM.	
00320 012A 20 05		BRA	LOOP2		
00330 012C 63 00	LOOP1	COM	X	WASTE TIME.	
00340 012E 63 00		COM	X		
00350 0130 01		NOP			
00360 0131 F6 0149	LOOP2	LDA B	FREQ	HALF-WAVE TIMEOUT	
00370 0134 5A	LOOP4	DEC B			
00380 0135 26 FD		BNE	LOOP4		
00390 0137 73 014B		COM	TOGGLE	OUTPUT COMPLEMENT TO	
00400 013A F6 014B		LDA B	TOGGLE	MUSIC BOARD.	
00410 013D F7 8010		STA B	MOD68		
00420 0140 20 D7		BRA	LOOP3		
00430 0142 7E E0D0	EXIT	JMP	MIKBUG		
00440 8010		MOD68	EQU	\$8010	MUSIC BOARD ADDRESS.
00450 0145 0D		SWEEPN	FCB	\$0D	DESIRED # OF SWEEPS.
00460 0146 0001		SWEEPS	RMB	1	TEMPORARY SWEEP COUNT
00470 0147 01		FIRSTF	FCB	\$01	STARTING SWEEP PARAM.
00480 0148 FF		LASTF	FCB	\$FF	ENDING SWEEP PARAM.
00490 0149 0001		FREQ	RMB	1	TEMPORARY FREQ.PARAM.
00500 014A 01		RATE	FCB	\$01	SWEEP RATE PARAMETER.
00510 014B 00		TOGGLE	FCB	0	
00520 E0D0		MIKBUG	EQU	\$E0D0	
00530 014C 01		NOP			
00540		END			

00100		NAM	NOISE	REV.A SEPT.1977
00110		OPT	NOP,O	
00120	* COPYRIGHT (C) 1977 ALL RIGHTS RESERVED			
00130	* NEWTECH COMPUTER SYSTEMS, INC.			
00140	*			
00150 0100		ORG	\$0100	
00160 0100 CE 0157		LDX	#TBL	INIT. ENVELOPE POINTER.
00170 0103 FF 0166		STX	TBLP	
00180 0106 86 02	LOOP2	LDA A	#\$02	IF ENVELOPE IS
00190 0108 FE 0166		LDX	TBLP	COMPLETE RETURN.
00200 010B A1 00		CMP A	X	
00210 010D 27 24		BEQ	EXIT	
00220 010F FE 0166		LDX	TBLP	ELSE INC. ENV. POINTER.
00221 0112 08		INX		
00222 0113 FF 0166		STX	TBLP	
00230 0116 B6 0163		LDA A	DURA	OUTPUT N RANDOM VALUES
00240 0119 B7 0168		STA A	DURAT	WHERE N=DURA.
00247 011C B6 0162	LOOP1	LDA A	FREQ	DELAY ACCORDING TO
00248 011F 4A	LOOP3	DEC A		FREQUENCY PARAMETER.
00249 0120 26 FD		BNE	LOOP3	
00250 0122 8D 12		BSR	RNDM	:GET RANDOM NUMBER IN A.
00260 0124 FE 0166		LDX	TBLP	:SCALE AMPLITUDE ACCORDING
00270 0127 A4 00		AND A	X	: TO ENV. TABLE.
00280 0129 B7 8010		STA A	MOD68	:OUTPUT TO MUSIC BOARD.
00290 012C 7A 0168		DEC	DURAT	
00300 012F 26 EB		BNE	LOOP1	
00310 0131 20 D3		BRA	LOOP2	PROCESS NEXT AMPLITUDE.
00320 0133 7E E0D0	EXIT	JMP	MIKBUG	YOUR EXIT MAY DIFFER!
00330	*			
00340	* RANDOM NUMBER GENERATOR. GENERATES 16 BIT			
00350	* VALUE IN "NMBER". RETURNS MOST SIGNIFICANT			
00360	* BYTE IN A.			
00370 0136 B6 0164	RNDM	LDA A	MSB	EXCLUSIVE-OR SHIFT
00380 0139 46		ROR A		REGISTER BITS 15,14,12&3.
00390 013A B8 0164		EOR A	MSB	15 & 14
00400 013D 46		ROR A		
00410 013E 46		ROR A		
00420 013F B8 0164		EOR A	MSB	12
00430 0142 46		ROR A		
00440 0143 B8 0165		EOR A	LSB	3
00450 0146 46		ROR A		
00460 0147 46		ROR A		
00470 0148 84 01		AND A	#\$01	MASK BIT 0.
00480 014A 78 0165		ASL	LSB	SHIFT NMBER LEFT.
00490 014D 79 0164		ROL	MSB	SETTING BIT 0 ACCORDING
00500 0150 BB 0165		ADD A	LSB	TO EXCLUSIVE-OR CALC.
00510 0153 B7 0165		STA A	LSB	
00520 0156 39		RTS		
00530	*			

00540	* AMPLITUDE ENVELOPE SPECIFICATION:			
00550 0157 FF	TBL	FCB	\$FF,\$FF,\$FF,\$7F,\$7F,\$3F	
0158 FF				
0159 FF				
015A 7F				
015B 7F				
015C 3F				
00560 015D 3F	FCB		\$3F,\$1F,\$0F,\$07,\$02	
015E 1F				
015F 0F				
0160 07				
0161 02				
00570	*			
00579 0162 30	FREQ	FCB	\$30	NOISE BAND PARAMETER.
00580 0163 FF	DURA	FCB	\$FF	DURATION PARAMETER.
00590 0164 0001	NMBER	FDB	\$01	SHIFT REGISTER.
00600 8010	MOD68	EQU	\$8010	MUSIC BOARD I/O ADDRESS.
00610 E0D0	MIKBUG	EQU	\$E0D0	
00620 0166 0002	TBLP	RMB	2	ENVELOPE TABLE POINTER.
00630 0168 0001	DURAT	RMB	1	TEMPORARY DURATION COUNT.
00640 0164	MSB	EQU	NMBER	RANDOM NUMBER ROUTINE.
00650 0165	LSB	EQU	NMBER+1	
00660 0169 01		NOP		
00670		END		

MUSIC BOARD,
MODEL 68 REV. B

NEWTECH
COMPUTER
SYSTEMS,
INC.



NEWTECH MODEL 68 MUSIC BOARD

Parts List

IC1 74LS10
IC2 74LS174
IC3 CD4050
IC4 LM380
IC5 78L05

R1-R5 Resistor, 20K Ohm 1%
R6 Resistor, 20K Ohm 5%
R7-R11 Resistor, 10K Ohm 1%
R12 Resistor, 10K Ohm 5%
R13 Resistor, Potentiometer, 25K Ohm
R14 Resistor, 2M Ohm 10%
R15 Resistor, 1 Ohm 10% 1/2 Watt
R16 Resistor, 4.7K Ohm 10%

C1 Capacitor, 1uF 35 Volt Tantalum
C2,C3, Capacitor, 0.1uF 12 Volt disc
C4,C5,C7
C6 Capacitor, 4.7uF 10 Volt Tantalum
C8 Capacitor, 220uF 25 Volt

SPKR Speaker, 8 Ohm 0.2 Watt

J1 Phono Jack