

Newtechniques

Newtechniques is published by Newtech Computer Systems, Inc. for dealers and purchasers of its products. Programs contained herein are the property of Newtech Computer Systems, Inc., except as otherwise noted. Permission is hereby granted for non-commercial use by dealers and purchasers of Newtech's music boards. All rights for commercial applications are retained by Newtech Computer Systems, Inc.

Newtechniques #1 - Dec. 1977
Copyright (C) 1977
Newtech Computer Systems, Inc.
230 Clinton Street
Brooklyn, New York 11201

The Model 6 Music Board for S-100 bus computers is sold with a Users Manual that contains MICROSCORE, a BASIC program for producing musical scores, and MICROPLAY, an Assembly Language program for playing the scores. The Users Manual for the Model 68 Music Board has similar programs respectively entitled SCORE68 and PLAY68.

Newtech music board owners and others interested in microcomputer music may want to read the October and November 1977 issues of ROM magazine. They contain a more detailed description of concepts used in our software and examples of improvements the user can make (e.g., how to make the computer "transpose", and how to do "ritards" and "accelerandos" by using notes with odd duration values). The articles also contain data statements for three tunes -- Chopin's "Minute Waltz", a movement from a Bach Flute Sonata, and one part of a Bach two-part Invention (with the music for the second part so you can play along). These issues of ROM are available from computer stores at \$2.00 per copy, or \$3.00 ppd. from the publisher. However, readers of Newtechniques can purchase both copies at the special price of \$4.00 ppd. for both copies directly from ROM Publications, Route 97, Hampton, Connecticut 06247.

This issue of Newtechniques contains listings of the Rev.B version of MICROSCORE and MICROPLAY. Changes from Rev.A are indicated by *. Similar changes can be made fairly easily for SCORE68 and PLAY68. The main change incorporated into the Rev.B programs is the capability of specifying one of eight different amplitude envelopes for each musical note. The user can now output loud, soft, short (staccato), long (legato), and all sorts of differently shaped notes, as well as rests. Note the processing of the optional envelope specification character in lines 183 through 197 of MICROSCORE. The Rev.B version of MICROPLAY, like the Rev.A version, was written to end by looping endlessly in place at location HERE. The user may change this to his own ending.

We also include some favorite Christmas tunes that you can type in. Show your friends and relatives what your computer can do -- play "Auld Lang Syne" for them at midnight on New Years Eve!

Several users of the Model 6 and the Model 68 have taken the time to write or call us with helpful suggestions and comments. We'd like to thank, in particular, Nancy Gleason, Rudy Libenschek, Aileen Harrison, Walt Caldwell, Cornell Frank, and Robert Doolittle. Both Rudy and Walt are running the Model 6 on a Poly 88 and were kind enough to send us information on changes necessary for running the Newtech programs on that system.

Newtech would like to hear from people using microcomputers for sound generation.

The Model 6 is demonstrated and discussed on the nationally broadcast NBC-TV show "Not For Women Only" hosted by Frank Field and Lynn Redgrave. The segment (called "Computers in the Home") is scheduled for Thursday of the week whose topic is "Computers in our Lives". Check local listings for date and time.

```

0000 ; MICROPLAY REV.B DECEMBER 1977
0000 ; NEWTECH COMPUTER SYSTEMS INC.
0000 ; 230 Clinton Street
0000 ; BROOKLYN, N.Y. 11201
0000 ;
0000 ; MICROPLAY STARTS AT THE BEGINNING OF THE
0000 ; MEMORY AREA DESIGNATED "SCORE" AND
0000 ; TRANSFERS INTO THE PLAY ROUTINE A 1-BYTE
0000 ; PITCH PARAMETER AND A 2-BYTE DURATION
0000 ; PARAMETER. (BITS A5,A4 AND A3 OF THE
0000 ; DURATION PARAMETER MOST SIGNIFICANT BYTE
0000 ; IS USED AS A FIELD TO SPECIFY WHICH
0000 ; AMPLITUDE ENVELOPE IS TO BE USED.)
0000 ; MICROPLAY CONTINUES TRANSFERRING NOTE
0000 ; PARAMETERS AND CALLING THE PLAY ROUTINE
0000 ; UNTIL A PITCH CONSTANT OF ZERO IS
0000 ; ENCOUNTERED WHICH INDICATES THE END OF
0000 ; THE MUSICAL SCORE.
0000 ; THIS VERSION OF MICROPLAY WAS WRITTEN
0000 ; FOR A 8080 HAVING 0 WAIT STATES BUT CAN
0000 ; BE MODIFIED FOR 8080'S WITH WAIT STATES
0000 ; OR FOR Z80 PROCESSORS.
0000 ;
0000 31 C1 00 BEGIN LXI SP,STACK ;INIT. STACK POINTER.
0003 21 00 01 INIT LXI H,SCORE ;INIT. SCORE POINTER.
0006 22 AF 00 SHLD PLACE
0009 2A AF 00 NEXT LHLD PLACE ;IF END OF SCORE THEN
000C 3E 00 MVI A,0 ;LOOP HERE.
000E BE CMP M
000F CA 0F 00 HERE JZ HERE ;YOUR ENDING?
0012 ; ;ELSE TRANSFER
0012 ; ;PARAMETERS FOR NEXT
0012 ; ;NOTE OF SCORE INTO
0012 ; ;PLAY ROUTINE.
0012 7E MOV A,M ;LOAD PITCH.
0013 32 4B 00 STA XFER2+1
0016 32 6A 00 STA XFER4+1
0019 23 INX H
001A 7E MOV A,M ;LOAD NOTE DURATION
001B 32 44 00 STA XFER1+1 ;LSD.
001E 32 67 00 STA XFER3+1
0021 23 INX H
0022 7E MOV A,M ;LOAD NOTE DURATION
0023 E6 07 ANI 07H ;MASK 3 LSB'S.
0025 32 45 00 STA XFER1+2 ;MSD.
0028 32 68 00 STA XFER3+2
* 002B 7E MOV A,M ;GET MSD AGAIN
* 002C E6 38 ANI 38H ;MASK 3 BITS
* 002E C6 6F ADI 6FH ;ENVELOPE SPEC ADDRESS
* 0030 32 3E 00 STA PLAY+1
0033 23 INX H
0034 22 AF 00 SHLD PLACE ;SAVE PLACE IN SCORE.
0037 ;
0037 CD 3D 00 CALL PLAY ;PLAY ONE NOTE.
003A C3 09 00 JMP NEXT ;GO DO NEXT NOTE.
003D ;
003D ;
003D ;
003D 21 6F 00 PLAY LXI H,TBL1 ;INIT ENVELOPE POINTER
0040 1E 08 MVI E,8 ;INIT. SEGMENT COUNT.
0042 7E MOV A,M ;GET STARTING
0043 ; ;AMPLITUDE.
0043 01 00 00 XFER1 LXI B,LNGTH ;INIT. DURATION COUNT.
0046 40 LOOP2 MOV B,B ;WASTE TIME (WTZ)
0047 C3 4A 00 JMP XFER2 ;(WT)
004A 16 00 XFER2 MVI D,PITCH ;INIT. PITCH CONSTANT.
004C D3 24 OUT MODL6 ;OUTPUT HALF WAVE TO
004E ; ;MUSIC BOARD.

```

```

004E 34           INR M      ;WASTE MUCH TIME (WMT)
004F 35           DCR M
0050 34           INR M
0051 35           DCR M
0052 34           INR M
0053 35           DCR M
0054 15           LOOP3 DCR D   ;DELAY ACCORDING TO
0055 C2 54 00     JNZ LOOP3  ;PITCH CONSTANT.
0058 AE           XRA M    ;COMPLEMENT A.
0059 0D           DCR C    ;COUNT DOWN DURATION .
005A C2 46 00     JNZ LOOP2
005D 05           DCR B
005E C2 4A 00     JNZ XFER2
0061 D3 24       OUT MODL6
0063 23           INX H    ;SET UP NEXT SEGMENT.
0064 1D           DCR E    ;DCR SEGMENT COUNT.
0065 C8           RZ      ;RETURN IF ALL
0066                 ;SEGMENTS DONE.
0066 01 00 00     XFER3 LXI B,LNGTH
0069 16 00     XFER4 MVI D,PITCH
006B 7E           MOV A,M  ;SET NEW AMPLITUDE.
006C C3 54 00     JMP LOOP3
006F
006F           ;ENVELOPE SPECIFICATION:
006F
006F           ; TBL1 THRU TBL8 ARE 8 DIFFERENT NOTE
006F           ; ENVELOPE SPECIFICATIONS, EACH ONE
006F           ; CONSISTING OF 8 AMPLITUDE SEGMENTS.
006F
*006F FF FF E0 C0 TBL1 DB 0FFH,0FFH,0E0H,0C0H ,ATTACK:#5
*0073 A0 90 70 50 DB 0A0H,90H,70H,50H
*0077 00 00 00 00 TBL2 DB 0,0,0,0 ,REST:#R
*007B 00 00 00 00 DB 0,0,0,0
*007F C0 FF 90 60 TBL3 DB 0C0H,0FFH,90H,60H ,STACCATO:#8
*0083 35 00 00 00 DB 35H,0,0,0
*0087 C0 FF FF FF TBL4 DB 0C0H,0FFH,0FFH,0FFH ,LEGATO:#L
*008B FF FF A0 90 DB 0FFH,0FFH,0A0H,90H
*008F 70 80 60 20 TBL5 DB 70H,80H,60H,20H ,SOFT STACCATO:#1
*0093 00 00 00 00 DB 0,0,0,0
*0097 70 80 80 80 TBL6 DB 70H,80H,80H,80H ,SOFT LEGATO:#2
*009B 80 80 60 40 DB 80H,80H,60H,40H
*009F A0 B7 D0 FF TBL7 DB 0A0H,0B7H,0D0H,0FFH ;"SHAPED":#3
*00A3 FF FF C7 A0 DB 0FFH,0FFH,0C7H,0A0H
*00A7 70 85 A0 B5 TBL8 DB 70H,85H,0A0H,0B5H ,CRESCENDO:#4
*00AB D0 FF FF A0 DB 0D0H,0FFH,0FFH,0A0H
00AF
00AF           ;SP EQU 6   ;FIX FOR OUR ASSEMBLER
00AF 00 00       PLACE DW 0   ;SCORE POINTER.
00B1           SCORE EQU 0100H ;YOUR SCORE LOCATION?
00B1           MODL6 EQU 24H  ;YOUR OUTPUT PORT?
*00B1           PITCH EQU 0   ;DUMMY EQUATE
*00B1           LNGTH EQU 0   ;DUMMY EQUATE
00B1           STACK EQU $+10H

```

```

100 REM MICROSCORE REV.B DECEMBER 1977
110 REM NEWTECH COMPUTER SYSTEMS INC.
120 REM 230 Clinton Street
130 REM BROOKLYN, N.Y. 11201
140 REM
150 LET U=256 \REM U DEFINES SCORE AREA IN MEMORY.
160 LET K1=2^(1/12)
170 LET K6=2.1 \REM TEMPO CONTROL
175 LET E1=0 \REM ENVELOPE POINTER DEFAULT.
*180 DIM Z$(6)
181 FOR V=1TO 1000
182 READ Z$
*183 LET E=100 \REM PROCESS ENVELOPE POINTER. E WILL
*184 REM BE ADDED TO THE MOST SIGNIFICANT BYTE OF THE
*185 REM DURATION COUNT SO THAT BITS A5,A4 AND A3 ARE
*186 REM USED AS AN ENVELOPE POINTER FIELD.
*187 REM
*188 IF Z$(1,1)="5" THEN E=0 \REM ATTACK
*189 IF Z$(1,1)="R" THEN E=8 \REM REST
*190 IF Z$(1,1)="S" THEN E=16 \REM STACCATO
*191 IF Z$(1,1)="L" THEN E=24 \REM LEGATO
*192 IF Z$(1,1)="1" THEN E=32 \REM SOFT STACCATO
*193 IF Z$(1,1)="2" THEN E=40 \REM SOFT LEGATO
*194 IF Z$(1,1)="3" THEN E=48 \REM SHAPED
*195 IF Z$(1,1)="4" THEN E=56 \REM CRESCENDO
*196 IF E=100 THEN GOTO 200
*197 LET E1=E
*198 LET Z$=Z$(2)
200 LET C=1
220 LET N=100
230 IF Z$(1,1)="A" THEN N=1
240 IF Z$(1,1)="B" THEN N=3
250 IF Z$(1,1)="C" THEN N=4
260 IF Z$(1,1)="D" THEN N=6
270 IF Z$(1,1)="E" THEN N=8
280 IF Z$(1,1)="F" THEN N=9
290 IF Z$(1,1)="G" THEN N=11
300 IF Z$(1,1)="X" THEN GOTO 720
310 IF N=100 THEN GOTO 760
320 LET C=2
330 LET M=100
340 IF Z$(2,2)!="!" THEN M=N-1
350 IF Z$(2,2)="#" THEN M=N+1
360 IF Z$(2,2)="/" THEN M=N
370 IF M=100 THEN GOTO 760
380 LET C=3
390 LET P=100
400 IF Z$(3,3)="1" THEN P=M
410 IF Z$(3,3)="2" THEN P=M+12
420 IF Z$(3,3)="3" THEN P=M+24
430 IF P=100 THEN GOTO 760
440 LET C=4
450 LET T=100
460 IF Z$(4,4)="S" THEN T=16
470 IF Z$(4,4)="E" THEN T=8
480 IF Z$(4,4)="Q" THEN T=4
490 IF Z$(4,4)="H" THEN T=2
500 IF Z$(4,4)="W" THEN T=1
510 IF T=100 THEN GOTO 760
520 IF LEN(Z$)=4 THEN GOTO 560
530 LET C=5
540 IF Z$(5,5)=". ." THEN T=2*T/3
550 REM CALCULATE CONSTANTS
560 LET F1=220*(K1^(P-1))
570 LET T1=10^6/(2*F1)
580 LET K3=(T1-56.5)/7.5
590 LET K4=F1/(K6*T)
600 LET D3=INT(K4) \REM MAKE DURATION EVEN#
610 LET D4=2*D3-2*INT(D3/2)
620 LET D5=INT(D4/256) \REM CALC. 2 BYTES
*630 LET D6=D5+1+E1 \REM D6=MSB
640 LET D7=D4-D5*256 \REM D7=LSB

```

```

650 REM TRANSFER CONSTANTS TO SCORE AREA.
660 FILL U+3*(V-1),INT(K3+.5)
670 FILL U+3*(V-1)+1,D7
680 FILL U+3*(V-1)+2,D6
690 PRINT V,
700 NEXT V
710 STOP
720 FILL U+3*(V-1),0
730 PRINT
740 PRINT "SCORE COMPIILATION COMPLETE!"
750 STOP
760 PRINT "ERROR IN NOTE #",V
770 PRINT "DATA STRING ",Z$
780 PRINT "CHARACTER #",C
790 STOP
800 END
810 REM
820 REM "AULD LANG SYNE"
901DATA"LC 1Q.,"F 1E.,"F 1Q","A 2Q"
902DATA"G 1Q.,"F 1E","G 1Q","A 2E","G 1E"
903DATA"F 1Q.,"F 1E","A 2Q","C 2Q"
904DATA"3D 2H.,"LD 2Q"
905DATA"C 2Q.,"A 2E","A 2Q","F 1Q"
906DATA"G 1Q.,"F 1E","G 1Q","A 2E","G 1E"
907DATA"F 1Q.,"D 1E","D 1Q","C 1Q"
908DATA"F 1H.,"2D 2Q"
909DATA"C 2Q.,"A 2E","A 2Q","F 1Q"
910DATA"G 1Q.,"F 1E","G 1Q","D 2Q"
911DATA"C 2Q.,"A 2E","A 2Q","D 2Q"
912DATA"4D 2H.,"LF 2Q"
913DATA"C 2Q.,"A 2E","A 2Q","F 1Q"
914DATA"G 1Q.,"F 1E","G 1Q","A 2E","G 1E"
915DATA"F 1Q.,"D 1E","D 1Q","C 1Q"
916DATA"3F 1W"
9000 DATA "X"

```

```

820 REM "GOD REST YE MERRY GENTLEMEN"
901DATA"5E 1Q.,"E 1Q","B 2Q","B 2Q","A 2Q"
902DATA"G 1Q.,"F#1Q","E 1Q","D 1Q"
903DATA"E 1Q.,"F#1Q","G 1Q","A 2Q"
904DATA"3B 2H.,"5E 1Q"
905DATA"E 1Q.,"B 2Q","B 2Q","A 2Q"
906DATA"G 1Q.,"F#1Q","E 1Q","D 1Q"
907DATA"E 1Q.,"F#1Q","G 1Q","A 2Q"
908DATA"3B 2H.,"5B 2Q"
909DATA"C 2Q.,"A 2Q","B 2Q","C 2Q"
910DATA"D 2Q.,"E 2Q","B 2Q","A 2Q"
911DATA"G 1Q.,"E 1Q","F#1Q","G 1Q"
912DATA"LA 2H.,"5G 1Q","A 2Q"
913DATA"LB 2H.,"5C 2Q","B 2Q"
914DATA"B 2Q.,"A 2Q","G 1Q","F#1Q"
915DATA"LE 1H.,"G 1E","F#1E","E 1Q"
916DATA"LA 2H.,"5G 1Q","A 2Q"
917DATA"B 2Q.,"C 2Q","D 2Q","E 2Q"
918DATA"B 2Q.,"A 2Q","G 1Q","F#1Q"
919DATA"3E 1W."
9000 DATA "X"

```

```

820 REM "THE EYES OF TEXAS"
901DATA"D 1Q.,"G 1Q.,"D 1E","G 1E.,"D 1S","G 1E.,"A 2S"
902DATA"B 2H.,"G 1H"
903DATA"C 2Q.,"C 2E","G 1Q","A 2Q"
904DATA"B 2H.,"D 1Q"
905DATA"G 1Q.,"D 1E.,"G 1E.,"D 1S","G 1E.,"A 2S"
906DATA"B 2H.,"G 1Q","B 2Q"
907DATA"B 2E.,"A 2Q.,"A 2Q","B 2Q"
908DATA"A 2W"
909DATA"A 2Q.,"A 2E","G#1E.,"A 2S","B 2E.,"A 2S"
910DATA"G 1H.,"D 1Q","B 2Q"
911DATA"C 2Q.,"C 2E","G 1E.,"G 1S","A 2E.,"A 2S"
912DATA"B 2H.,"D 1Q"
913DATA"E 1Q.,"F#1E.,"G 1E.,"F#1S","G 1E.,"E 1S"
914DATA"D 1H.,"G 1Q","A 2Q"
915DATA"B 2Q.,"C 2E","B 2Q","A 2Q"
916DATA"G 1H."
9000 DATA "X"

```

```

820 REM "THE FIRST NOEL"
901DATA"LE 1E","D 1E","C 1Q.,""D 1E","E 1E","F 1E"
902DATA"G 1H","A 2E","B 2E"
903DATA"C 2Q","B 2Q","A 2Q"
904DATA"G 1H","A 2E","B 2E"
905DATA"C 2Q","B 2Q","A 2Q"
906DATA"G 1Q","A 2Q","B 2Q"
907DATA"C 2Q","G 1Q","F 1Q"
908DATA"E 1H","2E 1E","D 1E"
909DATA"C 1Q.,""D 1E","E 1E","F 1E"
910DATA"G 1H","A 2E","B 2E"
911DATA"C 2Q","B 2Q","A 2Q"
912DATA"G 1H","A 2E","B 2E"
913DATA"C 2Q","B 2Q","A 2Q"
914DATA"G 1Q","A 2Q","B 2Q"
915DATA"C 2Q","G 1Q","F 1Q"
916DATA"E 1H","E 1E","D 1E"
917DATA"C 1Q.,""D 1E","E 1E","F 1E"
918DATA"G 1H","C 2E","B 2E"
919DATA"A 2H","A 2Q"
920DATA"4G 1H."
921DATA"LC 2Q","B 2Q","A 2Q"
922DATA"G 1Q","A 2Q","B 2Q"
923DATA"C 2Q","G 1Q","F 1Q"
924DATA"3E 1H."
9000 DATA "X"

```

```

820 REM "HARK! THE HERALD ANGELS SING"
901 DATA "5D 1Q","G 1Q","LG 1Q.,""F#1E"
902 DATA"5G 1Q","B 2Q","LB 2Q","A 2Q"
903DATA"5D 2Q","D 2Q","LD 2Q.,""C 2E"
904DATA"5B 2Q","A 2Q","3B 2H"
905DATA"5D 1Q","G 1Q","LG 1Q.,""F#1E"
906DATA"5G 1Q","B 2Q","LB 2Q","A 2Q"
907DATA"LD 2Q","A 2Q","A 2Q.,""F#1E"
908DATA"F#1Q","E 1Q","D 1H"
909DATA"5D 2Q","D 2Q","D 2Q","G 1Q"
910DATA"C 2Q","B 2Q","LB 2Q","A 2Q"
911DATA"5D 2Q","D 2Q","D 2Q","G 1Q"
912DATA"C 2Q","B 2Q","LB 2Q","A 2Q"
913DATA"5E 2Q","E 2Q","LE 2Q.,""LD 2E"
914DATA"C 2Q","B 2Q","C 2H"
915DATA"2A 2Q","B 2E","C 2E","D 2Q.,""G 1E"
916DATA"G 1Q","A 2Q","4B 2H"
917DATA"LE 2Q.,""E 2E","E 2Q","D 2Q"
918DATA"C 2Q","B 2Q","C 2H"
919DATA"A 2Q","B 2E","C 2E","D 2Q.,""G 1E"
920DATA"G 1Q","A 2Q","G 1H"
9000 DATA "X"

```

NEWTECH COMPUTER SYSTEMS, INC.
230 Clinton Street
BROOKLYN, NEW YORK 11201

♪ Newtechniques ♪

Newtechniques is published by Newtech Computer Systems, Inc. for dealers and purchasers of its products. Programs contained herein are the property of Newtech Computer Systems, Inc., except as otherwise noted. Permission is hereby granted for non-commercial use by dealers and purchasers of Newtech's music boards. All rights for commercial applications are retained by Newtech Computer Systems, Inc.

Newtechniques #2 - July 1978
Copyright (C) 1978
Newtech Computer Systems, Inc.
* * * * *
NEW ADDRESS!!!!:
230 Clinton Street
Brooklyn, New York 11201
* * * * *

This second issue of Newtechniques will be of interest primarily to Model 68 users. Newtechniques #1 had listings of our multi-envelope program for the Model 6; Newtechniques #2 has an implementation for the Model 68. We also include a procedure summary for the cassette provided with each Model 68.

Software

The box below summarizes our "Americana Plus" software, available through local computer stores. These products contain pre-coded music tunes, ready to load and run. American favorites, arranged for two voices, are "Take Me Out to the Ball Game," "Turkey in the Straw", "Camptown Races", "Star Spangled Banner" and "Dixie". The disk versions feature Jukebox programs (written in BASIC). These make selection of any song from the disk extremely easy and are ideal for demonstrating your computer.

Two-Part Software

NEWTECH has implemented two-voice music software for the Model 6 and Model 68 Music Boards. The two-voice BASIC (SCORE) program is similar to the original single-voice SCORE program. The two-voice Assembly Language (PLAY) program uses two waveform look-up tables, one for each voice. Each voice is produced by stepping through a table at a different programmed rate. These tables store waveforms for sine, square, triangle, sine with higher harmonics, etc.: different waveforms produce different sound qualities. This software has been used on our Americana Plus software products. Full documentation, including listings and instructions, for creating your own two-voice songs will be made available by NEWTECH. This software will optimally produce two-channel stereo when two Music Boards are used.

Personal Computing '78

NEWTech has accepted an invitation to participate in the Computer Music Festival to be held at PC'78 in Philadelphia from August 24th-27th. Come hear us at the Festival, or stop by at Booth #754.

"Americana Plus" Software

Pre-coded tunes, ready to load and run, including American favorites arranged for two voices, for NEWTECH Music Boards:

MD-1NS	16 tunes on North Star Computers, Inc. compatible disk, plus master JUKEBOX program	\$24.95
MD-1SW	14 tunes on SWTPC minIFLEX compatible disk, plus master JUKEBOX program	19.95
MC-1SW	14 tunes on SWTPC AC-30 compatible cassette	15.95
AC-1	"Americana Plus" audio demo cassette	5.00

Our correspondance indicates that many SWTPC owners have limited familiarity with Assembly Language programming and memory allocation considerations. The following details procedures to make music with the original PLAY68 and SCORE68 on the software cassette provided with the Model 68 Music Board.

To make music on your Model 68 Music Board & SWTPC Computer:

To create music on the Model 68 using the SCORE68 (BASIC) and PLAY68 (Assembly Language) programs, the following steps are necessary:

1. Code music as DATA statements and insert in place of the DATA statements in SCORE68 program.
 2. RUN SCORE68 with the new DATA statements, thus creating a "score" of constants at 2C00H.
 3. Load and execute PLAY68 Object Code at 2B00H.
 4. Save the file from 2B00H to the end of the "score".
- This file is all you'll need to load when you want to hear your newly-created music again.

To create PLAY68 Object Code you may either:

1. Use PLAY68 included in the STING (first file on Newtech cassette):

- a. Load STING at 2B00H-2DFFH, using MIKBUG L command. PLAY68 Object Code is from 2B00H-2BFFH, and the "score" for the STING is from 2C00H-2DFFH.
 - b. Execute STING to confirm that the program loaded. Since it ends by looping in place, reset the computer to return to MIKBUG.
 - c. Use the P command (on a cassette other than the Newtech cassette) to save locations 2B00H through 2BFFH.
 - d. Use the L command to load back the file. Reexecute the STING. This will confirm that PLAY68 Object Code was saved without errors being introduced.

OR

2. Create a new PLAY68 at the same or new location:

- a. Load CO-RES, then PLAY68 Source.
- b. If desired, change the starting address of PLAY68 (ORG \$2B00) and location of the "score" (SCORE EQU \$2C00). Line 150 in SCORE68 will have to be changed to the decimal equivalent of the new "score" location.
- c. Assemble PLAY68.
- d. Save PLAY68 so you can use it with the "scores" of music you've obtained by running SCORE68 with new DATA statements.

Notes:

1. SWTBUG can replace MIKBUG.
2. DISC can replace cassette; however, memory allocations for DISC BASIC vs SWTPC cassette BASIC are different, and SCORE68 and PLAY68 must be moved.

In Newtechniques #1 we mentioned that a program similar to our Model 6 multi-envelope program could readily be written for the Model 68. Two Model 68 users did indeed write such programs and sent us copies of their excellent work.

Roger Abrahams, manager of the Olson Electronics store in West Allis, Wisconsin, introduced a new feature in his program -- amplitude envelopes with fifteen segments -- to achieve better envelope shaping and improved sound quality. Roger sent us versions for both the AC-30 and SWTPC Disk systems.

Jim Stutsman of Carrollton, Texas submitted an extremely well written and documented cassette implementation with a number of interesting features. One was a separate "Keytone Routine" which produces a short beep each time a key is struck on the I/O terminal when in BASIC. Essentially, the Keytone Routine creates a music score of just one note; each time a key is struck, it calls the PLAY68 routine which uses that one-note score to produce the beep. Jim also arranged his memory assignments so that his PLAY68 routine would be coresident with BASIC and SCORE68, without BASIC destroying PLAY68 or the compiled music score. This was accomplished by setting the BASIC Interpreter memory availability pointer at 014EH to start allocating program storage space at a higher memory address than the default address. This leaves memory between the BASIC Interpreter and the start of SCORE68 for the insertion of PLAY68 and a music score. Jim also employed a USER function to call PLAY68 from SCORE68.

PLAY68B as presented herein represents a judicious combination of the above contributions with our own 6800 multienvelope software.

The memory map below is for the PLAY68B program included herein. Since the disk version of BASIC is longer than the cassette version, some adjustments will be required for disk system users; however, the same principles apply. A sample procedure for using FLEX 1.0 with PLAY68B at 2B00H and SCORE68B at 3100H is shown in Figure 1. This sample procedure includes a short score for "Happy Birthday".

Memory Map -- PLAY68B

0000 - 1EAFH	BASIC
1EE0 - 1FFFH	PLAY68B & KEYTONE
2000 - 22DFH	Reserved Score Area
22E0 -	SCORE68B or other BASIC routine

The PLAY68B notation is the same as for the single-voice PLAY68 program, except that an optional character may be added at the beginning of each string to specify the following amplitude envelopes:

R	Rest	2	Soft Legato
S	Staccato	3	Shaped Note
L	Legato	4	Crescendo
I	Soft Staccato	5	Attack

We'd like to thank Roger and Jim very much for sending us their work. We'd also like to say thanks to John Kleban of Staten Island, New York from whom we received much valuable assistance. We invite all users to let us know what they've done so we can share it with other Music Board users.

Fig.1 SAMPLE PROCEDURE USING MUSIC SOFTWARE & SWTPC DOS

```

$D                                     BOOT OPERATING SYSTEM
FLEX 1.0

+++TTTYYSSSETT,,DDXX==HH
+++BASIC                                Load BASIC
READY
#MON                                    Go to Monitor

$M 014E                                Change 014EH to 3100H. This
$014E 24      31                         is the Memory Availability
$014F 42      00                         Pointer..
$0150 01
$J 0100                                Go hardstart BASIC
READY
#DOS                                    Return to DOS

+++GET,PLAY68F
+++JUMP,0103
READY
#LOAD HAPPY1                            Load data for a song from
                                         a file or type data in by hand.

READY
#LIST

0998 REM K6=2
0999 REM "HAPPY BIRTHDAY"
1000 REM
1001 DATA 5C 1E.,LC 1S
1002 DATA D 1Q,C 1Q,F 1Q
1003 DATA 3E 1H,5C 1E.,LC 1S
1004 DATA D 1Q,C 1Q,G 1Q
1005 DATA 3F 1H,5C 1E.,LC 1S
1006 DATA 3C 2Q,A 2Q,2F 1Q
1007 DATA E 1Q,4D 1Q,5B!2E.,LB!2S
1008 DATA LA 2Q,F 1Q,G 1Q
1009 DATA SF 1H

READY
#APPEND MULTIEWB                      Overlay data with SCOR68 routine

READY
#170 K6=2                                Set tempo to desired value.
#RUN                                     Run program.

PLAY (P) OR SCORE (S)                   Score. Constants will be poked into
                                         memory.
? S
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25
SCORE COMPILATION COMPLETE
STOP 3750
READY
#
RUN                                     Run program.
PLAY (P) OR SCORE (S)                   Play.

? P
ARE YOU SURE (Y/N)?
? Y
STOP 0086                                Music will play.
READY
#DOS                                    Return to DOS

+++SAVE,HAPPY1.BIN,2B00,2CFF,7103  Save file for future use
+++MON                                     with BASIC routines.
$H 2B08
$2B08 39    7E
$2B09 01    71
$2B0A 01    03
$2B0B PE
                                         Optionally, goto monitor
                                         and change PLAY routine
                                         return to a jump to softstart
                                         DOS.

$J 7103                                 Go softstart DOS.

+++SAVE,HAPPY2.BIN,2B00,2CFF,2B00  Save this version for direct
+++HAPPY2.BIN                               calling from DOS.
+++
```

SCORE68B

```
0010 REM SCORE68 REV.B JUNE 1978
0020 REM COPYRIGHT (C) 1978
0030 REM NEWTECH COMPUTER SYSTEMS, INC.
0040 REM ALL RIGHTS RESERVED
0050 REM
0060 PRINT "PLAY (P) OR SCORE (S)"
0062 INPUT Q$
0064 IF Q$="P" THEN GOTO 70
0066 IF Q$="S" THEN GOTO 150
0068 GOTO 60
0070 PRINT "ARE YOU SURE (Y/N)?"
0072 INPUT Q$
0074 IF Q$= "Y" THEN GOTO 80
0076 GOTO 60
0080 POKE( 103,31) :REM PLAY ROUTINE AT 1F00H
0082 POKE( 104,00)
0084 LET W2=USER(0)
0086 STOP
0150 LET U=8192 : REM U DEFINES SCORE AREA IN MEMORY
0160 LET K1=2^(1/12)
0170 LET K6=1.2 :REM TEMPO CONTROL
0172 LET K6=K6*15/7 :REM 15/7 ADJUSTS FOR # OF SEGMENTS
0175 LET E1=0 :REM ENVELOPE POINTER DEFAULT
0180 FOR V=1TO 1000
0181 READ Z$
0183 LET E=100 :REM PROCESS ENVELOPE POINTER. E WILL
0184 REM BE ADDED TO THE MOST SIGNIFICANT BYTE OF THE
0185 REM DURATION COUNT SO THAT BITS A6,A5 AND A3 ARE
0186 REM USED AS AN ENVELOPE POINTER FIELD.
0187 REM
0188 LET AS=MIDS(Z$,1,1)
0189 IF AS="5" THEN E=0 :REM ATTACK
0190 IF AS="R" THEN E=1*16 :REM REST
0191 IF AS="S" THEN E=2*16 :REM STACCATO
0192 IF AS="L" THEN E=3*16 :REM LEGATO
0193 IF AS="1" THEN E=4*16 :REM SOFT ATACCATO
0194 IF AS="2" THEN E=5*16 :REM SOFT LEGATO
0195 IF AS="3" THEN E=6*16 :REM SHAPED
0196 IF AS="4" THEN E=7*16 :REM CRESCENDO
0197 IF E=100 THEN GOTO 200
0198 LET E1=E
0199 LET Z$=MIDS(Z$,2)
0200 LET C=1
0220 LET N=100
0225 LET AS=MIDS(Z$,1,1)
0230 IF AS="A" THEN N=1
0240 IF AS="B" THEN N=3
0250 IF AS="C" THEN N=4
0260 IF AS="D" THEN N=6
0270 IF AS="E" THEN N=8
0280 IF AS="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$=MIDS(Z$,2,1)
0340 IF A$!="!" THEN M=N-1
0350 IF A$="#" THEN M=N+1
0360 IF A$=" " THEN M=N
0370 IF M=100 THEN GOTO 760
0380 LET C=3
0390 LET P=100
0395 LET A$=MIDS(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$=MIDS(Z$,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$="W" THEN T=1
0510 IF T=100 THEN GOTO 760
0530 LET C=5
0540 IF MIDS(Z$,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
0630 LET D6=D5+E1 :REM E1= ENVELOPE INFO.
0640 LET D7=D4-D5*256 : REM D7=LS8
0650 REM TRANSFER CONSTANTS TO SCORE AREA
0660 POKE( U+3*(V-1),INT(K3+.5))
0670 POKE( U+3*(V-1)+1,D6)
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
8999 DATA X ← INSERT DATA HERE!
9000 END
```

PLAY68

00100 NAM PLAY68 REV.B MAY 1978
 00110 OPT O,NOP
 00120 * COPYRIGHT (C) 1978 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.
 00190 * THE ROUTINE THEN OUTPUTS TO
 00200 * THE MODEL 68 THE MUSICAL NOTE SPECIFIED BY THESE
 00210 * NOTE PARAMETERS. PLAY68 CONTINUES TRANSFERRING
 00220 * NOTE PARAMETERS AND OUTPUTTING EACH NOTE UNTIL
 00230 * A PITCH CONSTANT OF ZERO IS ENCOUNTERED WHICH
 00240 * INDICATES THE END OF THE MUSICAL SCORE.
 00245 * THIS ROUTINE DOES NOT USE THE STACK.
 00250 *
 00255 LEED BASEND EQU \$1EE0 END OF SWTPC 8K BASIC 2.0
 00260 1F00 ORG BASEND+\$20
 00270 2000 SCORE EQU BASEND+\$0120 SCORE LOCATION
 00280 1F00 C0 2000 LDX #SCORE INIT. SCORE POINTER.
 00290 1F03 FF 1FF6 STX PLACE
 00300 1F06 20 03 BRA NEXT
 00307 1F08 39 LXIT1 RTS
 00308 1F09 01 NOP
 00309 1F0A 01 NOP
 00310 1F0B FE 1FF6 NEXT LDY PLACE
 00320 1F0E 86 00 LDA A #0 IF END OF SCORE LOOP HERE.
 00330 1F10 A1 00 CMP A X
 00340 1F12 27 F4 HERE BEQ EXIT1 YOUR ENDING?
 00350 * ELSE TRANSFER PARAMETERS FOR NEXT NOTE OR SCORE
 00360 * INTO PLAY ROUTINE.
 00370 1F14 A6 00 LDA A X LOAD PITCH.
 00380 1F16 B7 1FFA STA A PITCH
 00390 1F19 08 INX
 00400 1F1A A6 00 LDA A X LOAD DURATION MSB
 00405 1F1C 84 07 AND A #00000011B MASK 3 LSB'S
 00410 1F1E B7 1FF4 STA A DURA
 00412 1F21 A6 00 LDA A X GET MSB AGAIN
 00414 1F23 84 70 AND A #01110000B MASK 3 BITS
 *****ERROR 210
 00416 1F25 8B 74 9416 ADD A #TELL ENVELOPE SPEC ADDRESS
 00417 1F27 B7 1F36 STA A PLAY+2 LOAD ENVELOPE POINTER
 00420 1F2A 08 INX
 00430 1F2b A6 00 LDA A X LOAD DURATION LSB.
 00440 1F2D B7 1FF5 STA A DURA+1
 00450 1F30 08 INX
 00460 1F31 FF 1FF6 STX PLACE SAVE SCORE POINTER.
 00490 * THE PLAY ROUTINE PLAYS ONE NOTE
 00500 1F34 C0 1F74 PLAY LLX #TBL1 INIT. ENVELOPE POINTER.
 00510 1F37 FF 1FF8 STX TBL1P STORE ENV. POINTER.
 00520 1F3A B6 30 LDA B X PUT AMPLITUDE VALUE IN B.
 00530 1F3C FE 1FF4 LDX DURA LOAD DURATION PARAMETER
 00540 * INTO INDEX REGISTER.
 00550 1F3F BC E000 LOOP3 CPX \$E000 5-WASTE TIME (31 STATES)
 00560 1F42 BC E000 CPX \$E000 5-
 00565 1F45 BC E000 CPX \$E000 5-
 00570 1F48 BC E000 CPX \$E000 5-
 00580 1F4B BC E000 CPX \$E000 5-
 00590 1F4E 73 E000 COM \$E000 6-
 00600 1F51 86 16 LOOP2 LDA A #22 4-FIXED DELAY TO ADJUST
 00610 1F53 4A LOOP4 DEC A 2- LOWEST NOTE TO 262Hz
 00620 1F54 26 FD BNE LOOP4 4- (MIDDLE C) WHEN PITCH
 00630 * PARAMETER=FE.
 00640 1F56 D6 1FFA LDA A PITCH 4-LOAD PITCH PARAMETER.
 00650 1F59 F7 B010 STA B MOD68 5-OUTPUT TO MUSIC BOARD.
 00660 1F5C 4A LOOP1 DEC A 2-DELAY AS PER PITCH PARAM.
 00670 1F5D 26 FD BNE LOOP1 4-
 00680 1F5F 53 COM B 2-COMPLEMENT WAVEFORM VALUE.
 00690 1F60 09 DEX 4-DECREMENT DURATION COUNTER.
 00700 1F61 26 DC BNE LOOP3 4-
 00710 1F63 7C 1FF9 INC TELL1P+1 6-SET UP NEXT SEGMENT.
 00720 1F66 FE 1FF8 LDX TBL1P 5-
 00730 1F69 B6 00 LDA B X 5-
 00740 1F6B C1 01 CMP B #\$01 2-END OF ENVELOPE CHAR.=01
 00750 1F6D 27 9C BEQ NEXT 4-GO DO NEXT NOTE.
 00760 1F6F FE 1FF4 LDX DURA 5-RESET DURATION PARAMETER.
 00770 1F72 20 DD BKA LOOP2 4-

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 L IS COMPLEMENTED
 00840 * BETWEEN 80 AND 7F. AN END OF ENVELOPE RECORD
 00850 * OF \$01 MARKS THE END OF THE SPECIFICATION.
 00851 *
 00856 1F74 TBL1 EQU * ENVELOPE SPECIFICATIONS
 0857 * TABLE 1 - ATTACK, #5
 0858 FCB \$FF,\$FF,\$F8,\$F0,\$E8,\$E0,\$D8,\$D0
 0859 FCB \$C8,\$C0,\$B8,\$B0,\$A0,\$90,\$85,\$01
 0860 * TABLE 2 - REST, #R
 0861 FCB \$80,\$80,\$80,\$80,\$80,\$80,\$80,\$80
 0862 FCB \$80,\$80,\$80,\$80,\$80,\$80,\$80,\$01
 0863 * TABLE 3 - STACCATO, #S
 0864 FCB \$E0,\$F0,\$FF,\$FF,\$FF,\$FF,\$FF
 0865 FCB \$98,\$8D,\$80,\$80,\$80,\$80,\$80,\$01
 0866 * TABLE 4 - LEGATO, #L
 0867 FCB \$E0,\$F0,\$FF,\$FF,\$FF,\$FF,\$FF
 0868 FCB \$FF,\$FF,\$FF,\$FF,\$FF,\$FF,\$FF
 0869 * TABLE 5 - SOFT STACCATO, #1
 0870 FCB \$L8,\$BD,\$C0,\$B8,\$B0,\$A0,\$90,\$88
 0871 FCB \$80,\$80,\$80,\$80,\$80,\$80,\$80,\$01
 0872 * TABLE 6 - SOFT LEGATO, #2
 0873 FCB \$B8,\$BD,\$C0,\$C0,\$C0,\$C0,\$C0
 0874 FCB \$C0,\$C0,\$C0,\$B8,\$B0,\$A8,\$A0,\$J1
 0875 * TABLE 7 - "SHAPED", #3
 0876 FCB \$D0,\$D6,\$D0,\$E3,\$E8,\$F5,\$FF,\$FF
 0877 FCB \$FF,\$FF,\$FF,\$F3,\$E5,\$DA,\$D0,\$01
 0878 * TABLE 8 - CRESCENDO, #4
 0879 FCB \$B8,\$BE,\$C4,\$CA,\$D0,\$D6,\$DC,\$E2
 0880 FCB \$L8,\$F4,\$FF,\$FF,\$FF,\$CD,\$A0,\$01
 00881 *
 00890 1FF4 0002 DURA RMB 2 DURATION CONSTANT.
 00900 8010 MOD68 EQU \$8010 MUSIC BOARD IN I/O SLOT 4.
 00910 1FF6 0002 PLACE RMB 2
 00920 1FF8 0002 TBLIP RMB 2 TABLE POINTER.
 00930 1FFA 0001 PITCH RMB 1 PITCH PARAMETER.
 01000 1EE0 ORG BASEND
 01010 * KEYTONE ROUTINE
 01020 * THE FOLLOWING ROUTINE PROVIDES A SHORT TONE
 01030 * OR "BEEP" EACH TIME A KEY IS STRUCK ON THE
 01040 * KEYBOARD AT PORT 1. THIS IS THE PORT THAT
 01050 * BASIC NORMALLY USES AS ITS CONTROL PORT.
 01060 1EE0 20 DEEP PCB \$20,\$28,\$B8,\$00 "BEEP" SCORE
 1EE1 28
 1EE2 08
 1EE3 00
 01070 1EAC INEED EQU \$1EAC MIKEBUG/SWTBUG INPUT
 01080 0085 PRMFLG EQU \$85 BASIC PROMPT FLAG
 01090 1EE4 BD 1EAC KEYTON JSR INEED GET INPUT CHARACTER
 01100 1EE7 36 PSH A SAVE CHARACTER
 01110 1EE8 96 85 LDA A PRMFLG PROMPT REQUIRED?
 01120 1EEA 26 09 BNE ENDTON GO IF NOT
 01130 1EEC CE 1LL0 LDX #BEEP PCINT TO SCORE
 01140 1EEF FF 1FF6 STX PLACE
 01150 1EF2 BD 1FFB JSR NEXT PLAY BEEP TONE
 01160 1EF5 32 ENDTON PUL A RESTORE CHARACTER
 01170 1EF6 39 RTS EXIT
 01180 * FORCE PORT 1 INPUT TO GO TO KEYTON ROUTINE
 01190 0112 ORG \$112 PORT 1 INPUT JUMP
 01200 0112 7E 1EE4 JMP KEYTON INPUT VECTOR
 01210 * FORCE BASIC TO LEAVE PLAY68 MEMORY UNUSED
 01220 * FOR STORAGE OF SCORE.
 01230 014E ORG \$14E BASIC WORK SPACE POINTER
 01240 014E 22E0 FDB BASEND+\$400 RESERVE 1K
 01250 * PLUG MIKEBUG/SWTBUG PROGRAM COUNTER TO CAUSE
 01260 * ENTRY INTO SWTPC 8K BASIC 2.0.
 01270 A048 ORG \$A048 PROGRAM COUNTER
 01280 A048 0100 FDB \$100 BASIC "COLD" ENTRY POINT
 01290 END