```
SOURCE FILE: MIDIMOCK.SC0
SOURCE FILE: MIDIMOCK.SC1
0000:               ; MOCKINGBOARD MIDI TONE GENERATOR FROM THE 1990´S
0000:               ; SOURCE RECONSTRUCTED FROM MACHINE CODE IN 2022
0000:               ;
0000:               ; SLOT DEPENDENT ADDRESSES
C0A8:              PPMIDICR EQU $C0A8     ;PASSPORT MIDI IN SLOT 2
C0A9:              PPMIDIDR EQU $C0A9
0000:               ; ADD TIMER 2 REGISTERS FOR VOLUME DIMINISH LOGIC
0000:               ;
C400:              MOCKBASE EQU $C400     ;MOCKINGBOARD IN SLOT 4
0000:               ;
0000:               ; APPLE ADDRESSES
C000:              APPLEKBD EQU $C000
C010:              KBDSTROB EQU $C010
0000:               ;
0000:               ; DATA SECTIONS TO BE LOADED FROM BASIC
0D00:              MIDISCRL EQU $0D00     ;SCREEN ADDRESS LO OF EACH MIDI NOTE N
UMBER
0D80:              MIDISCRH EQU $0D80     ;SCREEN ADDRESS HI OF EACH MIDI NOTE N
UMBER
0E00:              NUMDBYTS EQU $0E00     ;NUMBER OF DATA BYTES PER MIDI STATUS
MESSAGE ($980-$9FF)
0E80:              MOUSETXT EQU $0E80     ;MOUSETXT CHARACTER FOR EACH MIDI NOTE
 (00-7F)
0F00:              MIDMOCKL EQU $0F00     ;LO BYTES FOR MKB TUNINGS BY MIDI NOTE
 #
0F80:              MIDMOCKH EQU $0F80     ;HI BYTES FOR MKB TUNINGS BY MIDI NOTE
 #
0000:               ;
0000:               ; ZERO PAGE USAGE
0006:              ZPTEXTL      EQU       $06   ;USED FOR POKING TEXT SCREEN
0007:              ZPTEXTH      EQU       $07
0000:               ;
0020:              KEYEXIT      EQU       $20             ;SPACE BAR EXITS PROGR
AM
0000:               ;
----- NEXT OBJECT FILE NAME IS MIDIMOCK.OBJ
0900:                                            ORG $0900           ;ALLOWS SMAL
L BASIC PROGRAM AT $0800
0900:               ;
0900:4C F3 09                                    JMP TONEGEN         ;PLAY MIDI I
N ON MOCKINGBOARD WITH PIANO DISPLAY
0903:4C 1D 0A                                    JMP PHASPRCS        ;TO TEST: PO
KE INBYTE, 903G
0906:4C 44 09                                    JMP INITPSPT        ;INIT PASSPO
RT CARD
0909:4C 4F 09                                    JMP INITMOCK        ;INIT MOCKIN
GBOARD
090C:4C D3 09                                    JMP MIDIOUT         ;TEST SENDIN
G ONE BYTE THRU MIDI OUT ROUTINE
090F:4C 94 09                                    JMP MOCKPOKE        ;TEST POKING
 ONE MOCKINGBOARD REGISTER
0912:4C C1 09                                    JMP PANIC           ;TURN ALL SO
UNDS OFF
0915:               ;
0915:               ; USER POKABLE PARAMETERS
0915:00            INBYTE                 DFB $00             ;MIDI BYTE TO SEND
 - CAN USE FOR TESTING
0916:01            PIANOON      DFB       $01             ;SET TO 0 TO TURN OFF
TEXT PIANO DISPLAY
0917:01            USEVEL                 DFB $01             ;USE MIDI VELOCITY
 WHEN PLAYING NOTES
0918:00            USEBRIT      DFB       $00                       ;1=DISPLAY ONLY ONE MO
```

```
USETEXT CHARACTER
0919:5D         44 BRITISH      DFB       $5D              ;MOUSETEXT CHARACTER F
OR NOTE ON
091A:00         45 SPEAKER      DFB       $00              ;MOCKINGBOARD SPEAKER:
 0=LEFT, $80=RIGHT
091B:00         46 MOCKREG      DFB       $00              ;MOCKINGBOARD REGISTER
: 0-13
091C:00         47 MOCKBYTE DFB $00                    ;BYTE TO POKE IN MOCKINGBO
ARD REGISTER
091D:           48 ;
091D:           49 ; INTERNAL VARIABLES
091D:           50 ;
091D:00         51 PHASE                  DFB $00
091E:00         52 DATA1                  DFB $00
091F:00         53 DATA2                  DFB $00
0920:00         54 NOTESAVE DFB $00                              ;MUST BE THE BYT
E BEFORE NNUMSAVE
0921:00 00 00   55 NNUMSAVE DFB $00,$00,$00,$00,$00,$00   ;NOTE NUMBER FOR EACH
VOICE PLAYING
0924:00 00 00
0927:00 00 00   56                             DFB $00,$00,$00,$00,$00,$00   ;V
OICES RESERVED FOR PHASOR
092A:00 00 00
092D:00         57 VELSAVE      DFB       $00                        ;MUST BE THE
 BYTE BEFORE VOICEVOL
092E:00 00 00   58 VOICEVOL DFB $00,$00,$00,$00,$00,$00   ;MOCK VOLUME FOR EACH
VOICE PLAYING
0931:00 00 00
0934:00 00 00   59                             DFB $00,$00,$00,$00,$00,$00   ;V
OICES RESERVED FOR PHASOR
0937:00 00 00
093A:           60 ;
093A:00         61 XSAVE                  DFB $00
093B:00         62 YSAVE                  DFB $00
093C:00         63 TEXTL                  DFB $00
093D:00         64 TEXTH                  DFB $00
093E:00         65 TEXTNOTE DFB $00
093F:0A         66 CALCVOL      DFB       $0A              ;CONVERSION OF MIDI VO
L&VELOCITY TO MOCK VOLUME
0940:00         67 VOICENUM DFB $00
0941:00         68 CHNLSAVE DFB $00
0942:00         69 GOODSTS      DFB       $00
0943:00         70 APPLEKEY DFB $00
0944:           71 ;
0944:A9 13      72 INITPSPT LDA           #$13
0946:8D A8 C0   73                             STA   PPMIDICR
0949:A9 11      74                             LDA   #$11
094B:8D A8 C0   75                             STA   PPMIDICR
094E:60         76                              RTS
094F:           77 ;
094F:A9 FF      78 INITMOCK LDA           #$FF           ;SEE MOCKINGBOARD MANUA
L FOR PRIMARY ROUTINES
0951:8D 03 C4   79                             STA   MOCKBASE+$03
0954:8D 83 C4   80                             STA   MOCKBASE+$83
0957:A9 07      81                             LDA   #$07
0959:8D 02 C4   82                             STA   MOCKBASE+$02
095C:8D 82 C4   83                             STA   MOCKBASE+$82
095F:A9 00      84 RSETMOCK LDA           #$00             ;RESET
0961:8D 00 C4   85                             STA   MOCKBASE
0964:8D 80 C4   86                             STA   MOCKBASE+$80
0967:A9 04      87                             LDA   #$04
0969:8D 00 C4   88                             STA   MOCKBASE
096C:8D 80 C4   89                             STA   MOCKBASE+$80
096F:A9 00      90                             LDA   #$00
0971:8D 1A 09   91                             STA   SPEAKER
0974:A9 07      92                             LDA   #$07            ;ENABLE TONE
```

```
 ON ALL 3 CHANNELS
0976:8D 1B 09   93                               STA   MOCKREG
0979:A9 38      94                               LDA   #$38
097B:8D 1C 09   95                               STA   MOCKBYTE
097E:20 94 09   96                               JSR   MOCKPOKE
0981:A9 80      97                               LDA   #$80          ;SAME ON RIG
HT SPEAKER
0983:8D 1A 09   98                               STA   SPEAKER
0986:20 94 09   99                               JSR   MOCKPOKE
0989:A2 06     100                               LDX   #$06          ;RESET SILEN
CED ALL CHANNELS, SO ZERO OUT
098B:A9 00     101                               LDA   #$00          ;OUR TRACKIN
G FOR VOLUME OF EACH CHANNEL
098D:9D 20 09  102 PLUGVOL      STA          NOTESAVE,X
0990:CA        103                                DEX
0991:D0 FA     104                               BNE   PLUGVOL
0993:60        105                                RTS
0994:          106 ;
0994:AD 1A 09  107 MOCKPOKE LDA             SPEAKER
0997:29 80     108                               AND   #$80
0999:A8        109                                TAY
099A:AD 1B 09  110                               LDA   MOCKREG
099D:29 0F     111                               AND   #$0F
099F:C8        112                                INY
09A0:99 00 C4  113                               STA   MOCKBASE,Y
09A3:88        114                                DEY
09A4:A9 07     115                               LDA   #$07          ;LATCH
09A6:99 00 C4  116                               STA   MOCKBASE,Y
09A9:A9 04     117                               LDA   #$04
09AB:99 00 C4  118                               STA   MOCKBASE,Y
09AE:AD 1C 09  119                               LDA   MOCKBYTE
09B1:C8        120                                INY
09B2:99 00 C4  121                               STA   MOCKBASE,Y
09B5:88        122                                DEY
09B6:A9 06     123                               LDA   #$06          ;WRITE
09B8:99 00 C4  124                               STA   MOCKBASE,Y
09BB:A9 04     125                               LDA   #$04
09BD:99 00 C4  126                               STA   MOCKBASE,Y
09C0:60        127                                RTS
09C1:          128 ;
09C1:A9 B0     129 PANIC                    LDA   #$B0          ;MIDI MESSAGE: AL
L NOTES OFF
09C3:20 D3 09  130                               JSR   MIDIOUT
09C6:A9 7D     131                               LDA   #$7D          ;OMNI MODE O
N
09C8:20 D3 09  132                               JSR   MIDIOUT
09CB:A9 00     133                               LDA   #$00
09CD:20 D3 09  134                               JSR   MIDIOUT
09D0:4C 5F 09  135                               JMP   RSETMOCK
09D3:          136 ;
09D3:8D 15 09  137 MIDIOUT      STA          INBYTE
09D6:8E 3A 09  138                               STX   XSAVE
09D9:8C 3B 09  139                               STY   YSAVE
09DC:AD 15 09  140                               LDA   INBYTE
09DF:8D A9 C0  141                               STA   PPMIDIDR
09E2:AD A8 C0  142 MOUTTDRE LDA             PPMIDICR
09E5:29 02     143                               AND   #$02
09E7:F0 F9     144                               BEQ   MOUTTDRE
09E9:AD 15 09  145                               LDA   INBYTE
09EC:AE 3A 09  146                               LDX   XSAVE
09EF:AC 3B 09  147                               LDY   YSAVE
09F2:60        148                                RTS
09F3:          149 ;
09F3:          150                               CHN MIDIMOCK.SC1
09F3:            1 ; MOCKINGBOARD MIDI TONE GENERATOR FROM THE 1990´S - SOURCE 1
09F3:            2 ; SOURCE RECONSTRUCTED FROM MACHINE CODE IN 2022
```

```
09F3:            3 ;
09F3:20 44 09    4 TONEGEN      JSR         INITPSPT      ;ENTRY POINT: INITIALI
ZE CARDS
09F6:20 4F 09    5                                JSR   INITMOCK
09F9:AD A8 C0    6 MIDIIN                   LDA   PPMIDICR       ;WAIT FOR MIDI REC
EIVE DATA REGISTER FULL
09FC:29 01       7                                AND   #$01
09FE:D0 11       8                                BNE   GRABMIDI
0A00:AD 00 C0    9                                LDA   APPLEKBD      ;CHECK APPLE
 KEYBOARD WHILE WAITING
0A03:10 F4       10                               BPL   MIDIIN
0A05:2C 10 C0    11 APPLEMSG BIT             KBDSTROB
0A08:8D 43 09    12                               STA   APPLEKEY
0A0B:20 D2 0B    13                               JSR   PRCSAPPL      ;APPLE KBD H
ANDLER WILL SET CARRY IF EXITING
0A0E:90 E9       14                               BCC   MIDIIN
0A10:60          15                                RTS
0A11:AD A9 C0    16 GRABMIDI LDA             PPMIDIDR      ;GRAB THE MIDI BYTE
0A14:20 1A 0A    17                               JSR   PHASCHK
0A17:B8          18                                CLV
0A18:50 DF       19                               BVC   MIDIIN
0A1A:            20 ;
0A1A:8D 15 09    21 PHASCHK      STA         INBYTE        ;SAVE MIDI BYTE
0A1D:AD 1D 09    22 PHASPRCS LDA             PHASE
0A20:D0 45       23                               BNE   PHNOT00
0A22:AD 15 09    24                               LDA   INBYTE        ;PHASE 0: LO
OK FOR MIDI STATUS MESSGE
0A25:29 F0       25 STSMESSG AND             #$F0
0A27:C9 90       26                               CMP   #$90
0A29:D0 19       27                               BNE   NOTNOTE
0A2B:AD 15 09    28                               LDA   INBYTE        ;9X: SAVE MI
DI CHANNEL FOR NOTE ON
0A2E:29 0F       29                               AND   #$0F
0A30:8D 41 09    30                               STA   CHNLSAVE
0A33:A9 01       31                               LDA   #$01          ;NEXT PHASE=
1, ZERO OUT DATA REGS
0A35:8D 1D 09    32                               STA   PHASE
0A38:8D 42 09    33                               STA   GOODSTS       ;GOODSTS=1
0A3B:A9 00       34                               LDA   #$00
0A3D:8D 1E 09    35                               STA   DATA1
0A40:8D 1F 09    36                               STA   DATA2         ;DATA2 IS AL
SO A SIGNAL USED IN PHASE LOGIC
0A43:60          37                                RTS
0A44:            38 ;
0A44:C9 80       39 NOTNOTE      CMP         #$80
0A46:F0 06       40                               BEQ   PHAS80
0A48:A9 00       41                               LDA   #$00          ;SIGNAL TO I
GNORE ALL BYTES UNTIL NEXT NOTE FOUND
0A4A:8D 42 09    42                               STA   GOODSTS
0A4D:60          43                                RTS                ;ADD ADDITI
ONAL PHASE LOGIC HERE
0A4E:            44 ;
0A4E:A9 01       45 PHAS80                  LDA   #$01           ;8X: NEXT PHASE=1,
 DATA1=0 DATA2=1
0A50:8D 1D 09    46                               STA   PHASE
0A53:8D 1F 09    47                               STA   DATA2
0A56:8D 42 09    48                               STA   GOODSTS       ;GOODSTS=1
0A59:A9 00       49                               LDA   #$00
0A5B:8D 1E 09    50                               STA   DATA1
0A5E:AD 15 09    51                               LDA   INBYTE        ;SAVE CHANNE
L FOR NOTE BEING TURNED OFF
0A61:29 0F       52                               AND   #$0F
0A63:8D 41 09    53                               STA   CHNLSAVE
0A66:60          54                                RTS
0A67:            55 ;
0A67:AD 42 09    56 PHNOT00      LDA         GOODSTS       ;IGNORE DATA BYTES IF
```

```
NOT PROCESSING A NOTE
0A6A:D0 01      57                          BNE   CHKDATA
0A6C:60         58                           RTS
0A6D:AD 15 09   59 CHKDATA      LDA         INBYTE         ;HANDLE STATUS BYTE BE
FORE DATA COMPLETE
0A70:10 03      60                          BPL   YESDATA
0A72:4C 25 0A   61                          JMP   STSMESSG
0A75:AD 1E 09   62 YESDATA      LDA         DATA1          ;ARE WE PROCESSING A 9
X OR 8X?
0A78:D0 19      63                          BNE   PHDATA2
0A7A:AE 15 09   64                          LDX   INBYTE         ;PROCESSING
A 9X
0A7D:BD 00 0D   65                          LDA   MIDISCRL,X
0A80:8D 3C 09   66                          STA   TEXTL
0A83:BD 80 0D   67                          LDA   MIDISCRH,X
0A86:8D 3D 09   68                          STA   TEXTH          ;SET UP ADDR
ESS TO SCREEN LINE
0A89:8D 1D 09   69                          STA   PHASE          ;NEXT PHASE=
1
0A8C:8D 1E 09   70                          STA   DATA1          ;SIGNAL TO J
UMP TO PHASE 2
0A8F:8E 3E 09   71                          STX   TEXTNOTE       ;SAVE MIDI N
OTE NUMBER
0A92:60         72                           RTS
0A93:           73 ;
0A93:AD 1F 09   74 PHDATA2      LDA         DATA2          ;ARE WE PROCESSING DAT
A1 OR DATA2?
0A96:F0 2C      75                          BEQ   NOTEDOWN
0A98:AD 3C 09   76 NOTEUP                LDA   TEXTL          ;GET SAVED SCREEN
LINE BASE ADDRESS
0A9B:85 06      77                          STA   ZPTEXTL
0A9D:AD 3D 09   78                          LDA   TEXTH
0AA0:85 07      79                          STA   ZPTEXTH
0AA2:A9 01      80                          LDA   #$01           ;SET NEXT PH
ASE TO HANDLE RUNNING STATUS
0AA4:8D 1D 09   81                          STA   PHASE
0AA7:8D 42 09   82                          STA   GOODSTS
0AAA:A9 00      83                          LDA   #$00
0AAC:8D 1E 09   84                          STA   DATA1
0AAF:8D 1F 09   85                          STA   DATA2
0AB2:AD 16 09   86                          LDA   PIANOON
0AB5:F0 09      87                          BEQ   SKIPPOKE
0AB7:AE 3E 09   88                          LDX   TEXTNOTE       ;GET SAVED N
OTE NUMBER
0ABA:BD 80 0E   89                          LDA   MOUSETXT,X     ;RESTORE ORI
GINAL SCREEN CHARACTER
0ABD:20 10 0B   90                          JSR   POKETEXT
0AC0:20 B0 0B   91 SKIPPOKE JSR          FINDVOIC       ;ZERO OUT VELOCITY FOR
CURRENT NOTE
0AC3:60         92                           RTS
0AC4:           93 ;TODO: TEST/HANDLE RUNNING STATUS
0AC4:           94 ;
0AC4:AD 15 09   95 NOTEDOWN LDA          INBYTE         ;PROCESSING DATA2 - GET
 NOTE VELOCITY
0AC7:F0 CF      96                          BEQ   NOTEUP         ;PROCESS 9X
NN 00 AS NOTE OFF
0AC9:8D 2D 09   97                          STA   VELSAVE
0ACC:A9 01      98                          LDA   #$01           ;SET PHASE I
N CASE OF RUNNING STATUS
0ACE:8D 1D 09   99                          STA   PHASE
0AD1:8D 42 09  100                          STA   GOODSTS
0AD4:A9 00     101                          LDA   #$00
0AD6:8D 1E 09  102                          STA   DATA1
0AD9:8D 1F 09  103                          STA   DATA2

0ADC:AD 3C 09  104                          LDA   TEXTL          ;THIS IS A N
```

```
OTE ON.  GET SCREEN BASE ADDRESS
0ADF:85 06     105                              STA   ZPTEXTL
0AE1:AD 3D 09  106                              LDA   TEXTH
0AE4:85 07     107                              STA   ZPTEXTH
0AE6:A0 00     108                              LDY   #$00
0AE8:B1 06     109                              LDA   (ZPTEXTL),Y
0AEA:AE 3E 09  110                              LDX   TEXTNOTE      ;GET SAVED N
OTE NUMBER
0AED:AD 16 09  111                              LDA   PIANOON
0AF0:F0 0B     112                              BEQ   DONTPOKE
0AF2:AD 18 09  113                              LDA   USEBRIT       ;IF NOT USIN
G ONE MOUSETXT CHAR GET CHNL NUMBER
0AF5:F0 0A     114                              BEQ   CHNLCHAR
0AF7:AD 19 09  115                              LDA   BRITISH       ;TODO: REPLA
CE WITH NUMBER OF MIDI CHANNEL
0AFA:20 10 0B  116 POKEIT                 JSR   POKETEXT
0AFD:20 1A 0B  117 DONTPOKE JSR            PRCSVOIC      ;STORE METADATA ABOUT N
OTES BEING PLAYED
0B00:60        118                               RTS
0B01:          119 ;
0B01:AD 41 09  120 CHNLCHAR LDA           CHNLSAVE
0B04:18        121                               CLC
0B05:69 B0     122                              ADC  #$B0           ;CONVERT CHA
NNEL # TO NUMERICAL CHARACTER
0B07:C9 BA     123                              CMP  #$BA
0B09:90 EF     124                              BCC  POKEIT
0B0B:69 08     125                              ADC  #$08           ;CHNL 10-15
DISPLAYED WITH LETTER A-F
0B0D:4C FA 0A  126                              JMP  POKEIT
0B10:          127 ;
0B10:A0 00     128 POKETEXT LDY            #$00          ;POKE A CHARACTER IN A
ON THE APPLE SCREEN
0B12:91 06     129                              STA   (ZPTEXTL),Y
0B14:A9 00     130                              LDA   #$00          ;NET PHASE =
 0: SEARCH FOR MIDI MESSAGE
0B16:8D 1D 09  131                              STA   PHASE
0B19:60        132                               RTS
0B1A:          133 ;
0B1A:A2 01     134 PRCSVOIC LDX            #$01          ;SAVE NOTE NUMBER IN AR
RAY FOR VOICE
0B1C:BD 20 09  135 CHKVOICE LDA            NOTESAVE,X    ;LOOK AT NOTES PLAYING
ARRAY ELEMENT
0B1F:10 06     136                              BPL   CHKTEXT       ;SLOT FREE
0B21:E8        137 NEXTVOIC INX
0B22:E0 07     138                              CPX   #$07          ;CHANGE TO #
$0D FOR PHASOR
0B24:D0 F6     139                              BNE   CHKVOICE
0B26:60        140                               RTS
0B27:          141 ;
0B27:AD 3E 09  142 CHKTEXT      LDA         TEXTNOTE      ;GET NOTE NUMBER
0B2A:F0 2C     143                              BEQ   TEXTEXIT      ;DON´T STORE
 NOTE OFFS
0B2C:8D 20 09  144                              STA   NOTESAVE      ;SAVE CURREN
T NOTE
0B2F:09 80     145                              ORA   #$80          ;SIGNAL NOTE
 IS ON
0B31:9D 20 09  146                              STA   NOTESAVE,X    ;MARK VOICE
AS IN-USE
0B34:AD 17 09  147                              LDA   USEVEL
0B37:D0 05     148                              BNE   VEL2VOL
0B39:A9 0A     149                              LDA   #$0A
0B3B:B8        150                               CLV
0B3C:50 0A     151                              BVC   SAVEMVOL
0B3E:AD 2D 09  152 VEL2VOL      LDA         VELSAVE       ;USE MIDI VOL TO SCALE
 VELOCITY
0B41:4A        153                              LSR   A             ;MOCK VOLUME
```

```
 = MIDI VOLUME BITS 3-6
0B42:4A         154                          LSR   A
0B43:4A         155                          LSR   A
0B44:29 0F      156                          AND   #$0F
0B46:F0 10      157                          BEQ   TEXTEXIT      ;IF MOCK VOL
 WILL BE ZERO, DON´T STORE IN SLOT
0B48:           158 ;
0B48:8D 3F 09   159 SAVEMVOL STA        CALCVOL       ;SAVE CALCULATED MOCK V
OLUME
0B4B:CA         160                           DEX
0B4C:8E 40 09   161                          STX   VOICENUM      ;SAVE CURREN
T SLOT
0B4F:AD 2D 09   162                          LDA   VELSAVE       ;TRACK VELOC
ITY OF EACH NOTE PLAYING
0B52:9D 2D 09   163                          STA   VELSAVE,X     ;IN CASE NEE
DED FOR A GRAPHIC DISPLAY
0B55:20 59 0B   164                          JSR   LOOKVOIC
0B58:60         165 TEXTEXIT RTS
0B59:           166 ;
0B59:A0 00      167 LOOKVOIC LDY        #$00          ;FIGURE OUT WHICH MOCKI
NGBOARD VOICE TO PLAY
0B5B:8C 1A 09   168                          STY   SPEAKER
0B5E:AD 40 09   169                          LDA   VOICENUM
0B61:29 07      170                          AND   #$07
0B63:C9 06      171                          CMP   #$06
0B65:90 01      172                          BCC   GOODVOIC
0B67:60         173                           RTS
0B68:C9 03      174 GOODVOIC CMP        #$03
0B6A:90 08      175                          BCC   VOICADRS
0B6C:38         176                           SEC
0B6D:E9 03      177                          SBC   #$03
0B6F:A0 80      178                          LDY   #$80
0B71:8C 1A 09   179                          STY   SPEAKER
0B74:0A         180 VOICADRS ASL        A             ;REGISTER 0, 2, OR 4
0B75:8D 1B 09   181                          STA   MOCKREG
0B78:AE 20 09   182                          LDX   NOTESAVE
0B7B:BD 00 0F   183                          LDA   MIDMOCKL,X
0B7E:8D 1C 09   184                          STA   MOCKBYTE
0B81:20 94 09   185                          JSR   MOCKPOKE
0B84:AD 1B 09   186                          LDA   MOCKREG
0B87:18         187                           CLC
0B88:69 01      188                          ADC   #$01          ;REGISTER 1,
 3, OR 5
0B8A:8D 1B 09   189                          STA   MOCKREG
0B8D:AE 20 09   190                          LDX   NOTESAVE
0B90:BD 80 0F   191                          LDA   MIDMOCKH,X
0B93:8D 1C 09   192                          STA   MOCKBYTE
0B96:20 94 09   193                          JSR   MOCKPOKE
0B99:AD 1B 09   194                          LDA   MOCKREG
0B9C:38         195                           SEC
0B9D:E9 01      196                          SBC   #$01
0B9F:4A         197                          LSR   A
0BA0:18         198                           CLC
0BA1:69 08      199                          ADC   #$08          ;REGISTER 8,
 9, OR A
0BA3:8D 1B 09   200                          STA   MOCKREG
0BA6:AD 3F 09   201                          LDA   CALCVOL
0BA9:8D 1C 09   202                          STA   MOCKBYTE
0BAC:20 94 09   203                          JSR   MOCKPOKE
0BAF:60         204                           RTS
0BB0:           205 ;
0BB0:A2 06      206 FINDVOIC LDX        #$06          ;FIND THE VOICE PLAYING
 THE NOTE BEING TURNED OFF
0BB2:BD 20 09   207 FINDVLOO LDA        NOTESAVE,X
0BB5:29 7F      208                          AND   #$7F
0BB7:CD 3E 09   209                          CMP   TEXTNOTE
```

```
0BBA:F0 04     210                                                  BEQ   FNDVOICE
0BBC:CA        211                                                   DEX
0BBD:D0 F3     212                                                  BNE   FINDVLOO
0BBF:60        213                                                   RTS
0BC0:          214 ;
0BC0:A9 00     215 FNDVOICE LDA                   #$00          ;FREE UP VOICE FOR NOTE
 THAT TURNED OFF
0BC2:9D 20 09  216                                                  STA   NOTESAVE,X
0BC5:CA        217                                                   DEX
0BC6:8E 40 09  218                                                  STX   VOICENUM
0BC9:A9 00     219                                                  LDA   #$00
0BCB:8D 2D 09  220                                                  STA   VELSAVE
0BCE:20 59 0B  221                                                  JSR   LOOKVOIC      ;IS THIS A P
ROBLEM THAT CAN CAUSE STACK OVFL?
0BD1:60        222                                                   RTS
0BD2:          223 ;
0BD2:18        224 PRCSAPPL CLC
0BD3:AD 43 09  225                                                  LDA APPLEKEY
0BD6:29 7F     226                                                  AND #$7F
0BD8:C9 20     227                                                  CMP #KEYEXIT
0BDA:D0 02     228                                                  BNE NOTEXIT
0BDC:38        229                                                   SEC
0BDD:60        230                                                   RTS
0BDE:18        231 NOTEXIT                      CLC
0BDF:60        232                                                   RTS       ;KEYPRESS HANDLER GOE
S HERE
0BE0:          233 ;

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

```
 C000 APPLEKBD       0943 APPLEKEY      ?0A05 APPLEMSG       0919 BRITISH
 093F CALCVOL        0A6D CHKDATA        0B27 CHKTEXT        0B1C CHKVOICE
 0B01 CHNLCHAR       0941 CHNLSAVE       091E DATA1          091F DATA2
 0AFD DONTPOKE       0BB2 FINDVLOO       0BB0 FINDVOIC       0BC0 FNDVOICE
 0942 GOODSTS        0B68 GOODVOIC       0A11 GRABMIDI       0915 INBYTE
 094F INITMOCK       0944 INITPSPT       C010 KBDSTROB         20 KEYEXIT
 0B59 LOOKVOIC       09F9 MIDIIN         09D3 MIDIOUT        0D80 MIDISCRH
 0D00 MIDISCRL       0F80 MIDMOCKH       0F00 MIDMOCKL       C400 MOCKBASE
 091C MOCKBYTE       0994 MOCKPOKE       091B MOCKREG        0E80 MOUSETXT
 09E2 MOUTTDRE      ?0B21 NEXTVOIC      ?0921 NNUMSAVE       0AC4 NOTEDOWN
 0920 NOTESAVE       0A98 NOTEUP         0BDE NOTEXIT        0A44 NOTNOTE
?0E00 NUMDBYTS       09C1 PANIC          0A4E PHAS80         0A1A PHASCHK
 091D PHASE          0A1D PHASPRCS       0A93 PHDATA2        0A67 PHNOT00
 0916 PIANOON        098D PLUGVOL        0AFA POKEIT         0B10 POKETEXT
 C0A8 PPMIDICR       C0A9 PPMIDIDR       0BD2 PRCSAPPL       0B1A PRCSVOIC
 095F RSETMOCK       0B48 SAVEMVOL       0AC0 SKIPPOKE       091A SPEAKER
 0A25 STSMESSG       0B58 TEXTEXIT       093D TEXTH          093C TEXTL
 093E TEXTNOTE       09F3 TONEGEN        0918 USEBRIT        0917 USEVEL
 0B3E VEL2VOL        092D VELSAVE        0B74 VOICADRS       0940 VOICENUM
?092E VOICEVOL       093A XSAVE          0A75 YESDATA        093B YSAVE
   07 ZPTEXTH          06 ZPTEXTL
```

```
                                            06 ZPTEXTL          07 ZPTEXTH
    20 KEYEXIT        0915 INBYTE
  0916 PIANOON        0917 USEVEL         0918 USEBRIT        0919 BRITISH
  091A SPEAKER        091B MOCKREG        091C MOCKBYTE       091D PHASE
  091E DATA1          091F DATA2          0920 NOTESAVE      ?0921 NNUMSAVE
  092D VELSAVE       ?092E VOICEVOL       093A XSAVE          093B YSAVE
  093C TEXTL          093D TEXTH          093E TEXTNOTE       093F CALCVOL
  0940 VOICENUM       0941 CHNLSAVE       0942 GOODSTS        0943 APPLEKEY
  0944 INITPSPT       094F INITMOCK       095F RSETMOCK       098D PLUGVOL
  0994 MOCKPOKE       09C1 PANIC          09D3 MIDIOUT        09E2 MOUTTDRE
  09F3 TONEGEN        09F9 MIDIIN        ?0A05 APPLEMSG       0A11 GRABMIDI
  0A1A PHASCHK        0A1D PHASPRCS       0A25 STSMESSG       0A44 NOTNOTE
  0A4E PHAS80         0A67 PHNOT00        0A6D CHKDATA        0A75 YESDATA
  0A93 PHDATA2        0A98 NOTEUP         0AC0 SKIPPOKE       0AC4 NOTEDOWN
  0AFA POKEIT         0AFD DONTPOKE       0B01 CHNLCHAR       0B10 POKETEXT
  0B1A PRCSVOIC       0B1C CHKVOICE      ?0B21 NEXTVOIC       0B27 CHKTEXT
  0B3E VEL2VOL        0B48 SAVEMVOL       0B58 TEXTEXIT       0B59 LOOKVOIC
  0B68 GOODVOIC       0B74 VOICADRS       0BB0 FINDVOIC       0BB2 FINDVLOO
  0BC0 FNDVOICE       0BD2 PRCSAPPL       0BDE NOTEXIT        0D00 MIDISCRL
  0D80 MIDISCRH      ?0E00 NUMDBYTS       0E80 MOUSETXT       0F00 MIDMOCKL
  0F80 MIDMOCKH       C000 APPLEKBD       C010 KBDSTROB       C0A8 PPMIDICR
  C0A9 PPMIDIDR       C400 MOCKBASE
```