```
EE49 C3E5D6                JMP   :D6E5    Via D6E5 to 1EE4C
                    *
EE4C C5             L1E248 PUSH  B        Save length array
EE4D D5                    PUSH  D
EE4E CDA8CB                CALL  :CBA8    Erase stringreference
                                          in heap and symtab
EE51 2B                    DCX   H
EE52 2B                    DCX   H
EE53 D1                    POP   D
EE54 E5                    PUSH  H
EE55 1A                    LDAX  D
EE56 CD8BD1                CALL  :D18B    Get place in heap for string
EE59 E5                    PUSH  H
EE5A CD72D1                CALL  :D172    Transfer string into heap
EE5D C1                    POP   B
EE5E E1                    POP   H        )
EE5F 71                    MOV   M,C      ) Length into heap at
EE60 23                    INX   H        ) begin of string
EE61 70                    MOV   M,B      )
EE62 23                    INX   H
EE63 C1                    POP   B        Get length array
EE64 0B                    DCX   B        ) Update it
EE65 0B                    DCX   B        )
EE66 78                    MOV   A,B
EE67 B1                    ORA   C
EE68 C24CEE                JNZ   :EE4C    Next string if not ready
EE6B C32FEE                JMP   :EE2F    Stop reading, select ROM
                                          bank 0; abort
                    *
                    *
                    *    ============
                    *** SOUND MODULE ***
                    *    ============
                    *
                    *
EE6E 0E00           TEMPO  MVI   C,:00    Count SCB
EE70 21C201                LXI   H,:01C2  Addr sound control block 0
EE73 E5             L1E250 PUSH  H        Preserve addr SCB
EE74 7E                    MOV   A,M      Get value of volume counter
EE75 FEFE                  CPI   :FE
EE77 CA7EEE                JZ    :EE7E    If FE: No increment (sound
                                          forever)
EE7A D29DEF                JNC   :EF9D    If FF: Goto next block
                                          (sound off)
EE7D 34                    INR   M        ) Incr duration count volume
EE7E 3C             L1E251 INR   A        )
EE7F E5                    PUSH  H        Preserve addr SCB
EE80 47                    MOV   B,A      Save incr duration count
EE81 23                    INX   H
EE82 5E                    MOV   E,M      ) Get pntr envelope count
EE83 23                    INX   H        ) in DE
EE84 56                    MOV   D,M      )
EE85 1A                    LDAX  D        Get envelope duration count
EE86 B8                    CMP   B        Comp with volume count
EE87 D2B8EE                JNC   :EEB8    Jump if env. not counted out

                    * Envelope counted out:

EE8A EB                    XCHG
EE8B E3                    XTHL           Addr env.duration on stack;
                                          addr SCB in HL
EE8C 3600                  MVI   M,:00    Present duration count is 0
```

```
EE8E E1                     POP   H          Get pntr to env.table
EE8F 23                     INX   H          +1
EE90 7E                     MOV   A,M        Get next env. duration
EE91 B7                     ORA   A          Is it FF ?
EE92 FC93EF                 CM    :EF93      Then restart envelope
EE95 47                     MOV   B,A        Env duration in B
EE96 23                     INX   H          Pnts to next pos env table
EE97 7E                     MOV   A,M        Value in A
EE98 EB                     XCHG
EE99 72                     MOV   M,D        ) Set env pointer in SCB
EE9A 2B                     DCX   H          ) to new time field
EE9B 73                     MOV   M,E        )
EE9C 23                     INX   H
EE9D 23                     INX   H
EE9E 23                     INX   H
EE9F 23                     INX   H          HL pnts to vol.multiplier
EEA0 E5                     PUSH  H
EEA1 6E                     MOV   L,M        Sound volume *8 in L
EEA2 2600                   MVI   H,:00
EEA4 29                     DAD   H          *16 in HL
EEA5 EB                     XCHG             Multiplier in DE
EEA6 218000                 LXI   H,:0080    Init.value
EEA9 05           L1E252    DCR   B          Decr. envelope duration
EEAA FAB1EE                 JM    :EEB1      Jump if ready
EEAD 19                     DAD   D          Add multiplier
EEAE C3A9EE                 JMP   :EEA9      Again
EEB1 44           L1E253    MOV   B,H        New eff. volume in B
EEB2 E1                     POP   H
EEB3 23                     INX   H
EEB4 70                     MOV   M,B        Set new basic volume
EEB5 C3BDEE                 JMP   :EEBD

                  * If envelope not counted out:

EEB8 D1           L1E254    POP   D
EEB9 23                     INX   H
EEBA 23                     INX   H
EEBB 23                     INX   H
EEBC 23                     INX   H

                  * Handle tremolo:

EEBD 46           L1E255    MOV   B,M        Get basic volume in B
EEBE 23                     INX   H
EEBF 7E                     MOV   A,M        Get tremolo count
EEC0 B7                     ORA   A
EEC1 CAEBEE                 JZ    :EEEB      Jump if no tremolo adj.
EEC4 C601                   ADI   :01        )
EEC6 CE00                   ACI   :00        ) Incr tremolo count
EEC8 77                     MOV   M,A        )
EEC9 1F                     RAR
EECA 1F                     RAR
EECB 1F                     RAR
EECC D2DCEE                 JNC   :EEDC      No adj. if bit 2 of
                                             <T> is 0
EECF 04                     INR   B          )
EED0 04                     INR   B          ) Else add 4 units to
EED1 04                     INR   B          ) basic volume
EED2 04                     INR   B          )
EED3 1F                     RAR
EED4 00                     NOP
EED5 DADCEE                 JC    :EEDC      No adjust if bit 2 of
```

```
                                                 <T> = 0
EED8 78                  MOV    A,B
EED9 D608                SUI    :08          Else: basic vol. -2 units
EEDB 47                  MOV    B,A
                *
EEDC 00         L1E256   NOP
EEDD 78                  MOV    A,B          Get updated basic volume
EEDE 0600                MVI    B,:00
EEE0 B7                  ORA    A
EEE1 FAEBEE              JM     :EEEB        Jump if 81-FF
EEE4 060F                MVI    B,:0F
EEE6 B8                  CMP    B
EEE7 D2EBEE              JNC    :EEEB        Jump if >=0F (max.value)
EEEA 47                  MOV    B,A
                *
EEEB 23         L1E257   INX    H            Pnts to actual volume
EEEC 78                  MOV    A,B          Get new basic volume
EEED 96                  SUB    M            Minus actual volume
EEEE 07                  RLC                 )
EEEF 1F                  RAR                 )
EEF0 3F                  CMC                 ) New actual volume is
EEF1 CE00                ACI    :00          ) old one + 0.5
EEF3 07                  RLC                 ) (difference +/- 1)
EEF4 1F                  RAR                 )
EEF5 1F                  RAR                 )
EEF6 86                  ADD    M            )
EEF7 77                  MOV    M,A          Store in SCB
EEF8 47                  MOV    B,A          and in B
EEF9 E5                  PUSH   H
EEFA C5                  PUSH   B
EEFB 119402              LXI    D,:0294      Addr POR0M
EEFE 2104FD              LXI    H,:FD04      Addr POR0
EF01 79                  MOV    A,C          Get SCB count
EF02 0EF0                MVI    C,:F0        Mask for vol. SCB0,SCB2
EF04 0F                  RRC
EF05 D212EF              JNC    :EF12        Jump if SCB0,SCB2
EF08 0E0F                MVI    C,:0F        Mask for vol. SCB1,NCB
EF0A F5                  PUSH   PSW
EF0B 78                  MOV    A,B          )
EF0C 87                  ADD    A            ) Actual volume into
EF0D 87                  ADD    A            ) hinibble for SCB1
EF0E 87                  ADD    A            ) and NCB
EF0F 87                  ADD    A            )
EF10 47                  MOV    B,A          )
EF11 F1                  POP    PSW
EF12 1F         L1E258   RAR
EF13 D218EF              JNC    :EF18        Jump if SCB0,SCB1
EF16 13                  INX    D            ) POR0M+1,POR0+1
EF17 23                  INX    H            ) for SCB2,NCB
EF18 1A         L1E259   LDAX   D            Get POR0M,POR1M
EF19 A1                  ANA    C            Only reqd volume
EF1A B0                  ORA    B            Update it
EF1B 12                  STAX   D            Back in POR0M,POR1M
EF1C 77                  MOV    M,A          and in POR0,POR1
EF1D C1                  POP    B
EF1E E1                  POP    H
EF1F 0C                  INR    C            SCB count +1
EF20 79                  MOV    A,C
EF21 FE04                CPI    :04
EF23 CAA4EF              JZ     :EFA4        Ready if block was NCB
```

```
EF26 23                INX   H
EF27 7E                MOV   A,M       Get glissando flag
EF28 3D                DCR   A
EF29 FA8BEF            JM    :EF8B     Ready if end period
                                       reached
EF2C 23                INX   H
EF2D 5E                MOV   E,M       ) Get current period of
EF2E 23                INX   H         ) output in DE
EF2F 56                MOV   D,M       )
EF30 E5                PUSH  H
EF31 23                INX   H
EF32 7E                MOV   A,M       ) Get reqd final period
EF33 23                INX   H         ) in HL
EF34 66                MOV   H,M       )
EF35 6F                MOV   L,A
EF36 C23BEF            JNZ   :EF3B     Jump if end period not
                                       reached
EF39 54                MOV   D,H       ) HL=DE if 'set freq'
EF3A 5D                MOV   E,L       )
EF3B CD14DE   L1E260   CALL  :DE14     Compare HL-DE
EF3E F5                PUSH  PSW
EF3F D5                PUSH  D
EF40 D244EF            JNC   :EF44     Jump if final period >=
                                       current period
EF43 EB                XCHG            Else exchange values
EF44 CD1ADE   L1E261   CALL  :DE1A     Calc difference in HL
EF47 D1                POP   D
EF48 D5                PUSH  D
EF49 E5                PUSH  H
EF4A EB                XCHG
EF4B 1E40              MVI   E,:40     )
EF4D 7D       L1E262   MOV   A,L       )
EF4E 17                RAL             )
EF4F 6F                MOV   L,A       )
EF50 7C                MOV   A,H       )
EF51 17                RAL             ) HL = HL SHL 6
EF52 67                MOV   H,A       )
EF53 7B                MOV   A,E       )
EF54 17                RAL             )
EF55 5F                MOV   E,A       )
EF56 D24DEF            JNC   :EF4D
EF59 6C                MOV   L,H       ) HL = 1/64 orig. value
EF5A 63                MOV   H,E       )
EF5B 7C                MOV   A,H
EF5C B5                ORA   L
EF5D C261EF            JNZ   :EF61
EF60 23                INX   H
EF61 D1       L1E263   POP   D
EF62 CD14DE            CALL  :DE14     Compare HL-DE
EF65 0602              MVI   B,:02
EF67 DA6DEF            JC    :EF6D     Jump if DE > HL
EF6A 0600              MVI   B,:00
EF6C EB                XCHG
EF6D D1       L1E264   POP   D
EF6E F1                POP   PSW
EF6F D275EF            JNC   :EF75
EF72 CD26DE            CALL  :DE26     HL is its 2-compl.
EF75 19       L1E265   DAD   D
EF76 EB                XCHG
EF77 E1                POP   H
EF78 72                MOV   M,D       ) Set new current period
```

```
EF79 2B                   DCX    H            )
EF7A 73                   MOV    M,E          )
EF7B 2B                   DCX    H
EF7C 70                   MOV    M,B          Set glissando flag
EF7D C5                   PUSH   B
EF7E 79                   MOV    A,C          )
EF7F 3D                   DCR    A            ) Calc offset for oscill.
EF80 87                   ADD    A            ) address
EF81 4F                   MOV    C,A          )
EF82 0600                 MVI    B,:00
EF84 2100FC               LXI    H,:FC00      Addr osc. channel 0
EF87 09                   DAD    B            HL = addr current osc
EF88 73                   MOV    M,E          ) Load oscillator
EF89 72                   MOV    M,D          )
EF8A C1                   POP    B

                  * Block done:

EF8B 110E00       L1E266  LXI    D,:000E
EF8E E1                   POP    H            Get startadr prev. block
EF8F 19                   DAD    D            HL pnts to next block
EF90 C373EE               JMP    :EE73        Run next block
                  *
                  * RESTART ENVELOPE:
                  *
                  * Gets 1st volume of envelope.
                  *
                  * Entry: DE: Points to 2nd byte of envelope pointer
                  *            of a SCB.
                  * Exit:  A:  Volume.
                  *        HL: Address volume field in envelope.
                  *        BCDEF preserved.
                  *
EF93 D5           L1E267  PUSH   D            Save pntr
EF94 13                   INX    D
EF95 1A                   LDAX   D            Get lobyte env pntr
EF96 6F                   MOV    L,A          in L
EF97 13                   INX    D
EF98 1A                   LDAX   D            Get hibyte env pntr
EF99 67                   MOV    H,A          in H
EF9A D1                   POP    D            Restore pntr
EF9B 7E                   MOV    A,M          Get volume in A
EF9C C9                   RET
                  *
                  * IF SOUND OF BLOCK IS 'OFF': GOTO NEXT BLOCK:
                  *
EF9D 0C           L1E268  INR    C            Update SCB count
EF9E 79                   MOV    A,C
EF9F FE04                 CPI    :04          All blocks done ?
EFA1 C28BEF               JNZ    :EF8B        Next block if not
EFA4 E1           L1E269  POP    H
EFA5 C9                   RET
                  *
                  ***************
                  * BANK RETURN *
                  ***************
                  *
                  * Goto ROM bank 0.
                  *
                  * Entry: A: Current POROM.
                  *
EFA6 C1           BRET    POP    B
```

```
EFA7 E63F                  ANI    :3F        Select ROM bank 0
EFA9 C308D8                JMP    :D808      Load POR0/POR0M
                   *
EFAC FF                    DATA   :FF
EFAD FF                    DATA   :FF
EFAE FF                    DATA   :FF
EFAF FF                    DATA   :FF
EFB0 FF                    DATA   :FF
EFB1 FF                    DATA   :FF
EFB2 FF                    DATA   :FF
EFB3 FF                    DATA   :FF
EFB4 FF                    DATA   :FF
                   *
                   ***************************************
                   * LOADA: READ VARIABLE TYPE FROM TAPE *
                   ***************************************
                   *
                   * Reads 1 byte from tape.
                   *
                   * Exit: A: Variable type.
                   *         DEHL preserved.
                   *
EFB5 E5            R1BB    PUSH   H
EFB6 D5                    PUSH   D
EFB7 213E01                LXI    H,:013E    Startaddr EBUF
EFBA 113F01                LXI    D,:013F    Next addr
EFBD CDD102                CALL   :02D1      Read type from tape
EFC0 D2B3D2                JNC    :D2B3      Evt run 'LOADING ERROR ..'
EFC3 3A3E01                LDA    :013E      Get var.type in A
EFC6 D1                    POP    D
EFC7 E1                    POP    H
EFC8 C9                    RET
                   *
EFC9 FF                    DATA   :FF
EFCA FF                    DATA   :FF
EFCB FF                    DATA   :FF
EFCC FF                    DATA   :FF
EFCD FF                    DATA   :FF
EFCE FF                    DATA   :FF
EFCF FF                    DATA   :FF
EFD0 FF                    DATA   :FF
EFD1 FF                    DATA   :FF
EFD2 FF                    DATA   :FF
EFD3 FF                    DATA   :FF
EFD4 FF                    DATA   :FF
EFD5 FF                    DATA   :FF
EFD6 FF                    DATA   :FF
EFD7 FF                    DATA   :FF
EFD8 FF                    DATA   :FF
EFD9 FF                    DATA   :FF
EFDA FF                    DATA   :FF
EFDB FF                    DATA   :FF
EFDC FF                    DATA   :FF
EFDD FF                    DATA   :FF
EFDE FF                    DATA   :FF
EFDF FF                    DATA   :FF
EFE0 FF                    DATA   :FF
EFE1 FF                    DATA   :FF
EFE2 FF                    DATA   :FF
EFE3 FF                    DATA   :FF
EFE4 FF                    DATA   :FF
EFE5 FF                    DATA   :FF
```

```
EFE6 FF                 DATA  :FF
EFE7 FF                 DATA  :FF
EFE8 FF                 DATA  :FF
EFE9 FF                 DATA  :FF
EFEA FF                 DATA  :FF
EFEB FF                 DATA  :FF
EFEC FF                 DATA  :FF
EFED FF                 DATA  :FF
EFEE FF                 DATA  :FF
EFEF FF                 DATA  :FF
EFF0 FF                 DATA  :FF
EFF1 FF                 DATA  :FF
EFF2 FF                 DATA  :FF
EFF3 FF                 DATA  :FF
EFF4 FF                 DATA  :FF
EFF5 FF                 DATA  :FF
EFF6 FF                 DATA  :FF
EFF7 FF                 DATA  :FF
EFF8 FF                 DATA  :FF
                *
                *************************
                * part of 'EXP' (1E667) *
                *************************
                *
EFF9 B4         L1E272  ORA   H
EFFA F2B8E6             JP    :E6B8
EFFD C3ADE6             JMP   :E6AD
                *
                *
                *
F000                    END
```

```
***************************
* S Y M B O L   T A B L E *
***************************

BRET    EFA6   L1E245 EE0B   L1E246 EE2F   L1E247 EE38
L1E248 EE4C   L1E250 EE73   L1E251 EE7E   L1E252 EEA9
L1E253 EEB1   L1E254 EEB8   L1E255 EEBD   L1E256 EEDC
L1E257 EEEB   L1E258 EF12   L1E259 EF18   L1E260 EF3B
L1E261 EF44   L1E262 EF4D   L1E263 EF61   L1E264 EF6D
L1E265 EF75   L1E266 EF8B   L1E267 EF93   L1E268 EF9D
L1E269 EFA4   L1E272 EFF9   L1E273 EE0F   R1BB    EFB5
TEMPO   EE6E
```