

SNDBD2 Source Code

```
                                ;User Interface
0400 209004      JSR $0490      ;Initialise DAC
0403 2073FE      JSR $FE73
0406 A952        LDA #$0052
0408 2075FE      JSR $FE75      ;Output "R"
040B 20FAFD      JSR $FDFA      ;Poll KB
040E A501        LDA $0001
0410 C921        CMP #$0021
0412 3006        BMI $041A      ;branch if valid
0414 2075FE      JSR $FE75
0417 4C0B04      JMP $040B      ;start again
041A A000        LDY #$0000
041C 2028FF      JSR $FF28
041F D024        BNE $0445
0421 A92C        LDA #$002C      ;output ","
0423 2075FE      JSR $FE75
0426 A513        LDA $0013
0428 290F        AND #$000F
042A 8540        STA $0040      ;store Reg
042C 20A004      JSR $04A0
042F 20B004      JSR $04B0
0432 20C004      JSR $04C0
0435 20D004      JSR $04D0      ;Read DAT
0438 A541        LDA $0041
043A 200BFF      JSR $FF0B      ;output current value
043D A92C        LDA #$002C
043F 2075FE      JSR $FE75      ;output ","
0442 4C0B04      JMP $040B
0445 C8          INY
0446 C8          INY
0447 C8          INY
0448 2028FF      JSR $FF28      ;get new value
044B 500A        BVC $0457
044D A513        LDA $0013
044F 8542        STA $0042      ;store value
0451 20A004      JSR $04A0
0454 20E004      JSR $04E0
0457 4C0004      JMP $0400      ;Start again

                                ;Routine to load registers with pre-set values (max 16).
                                ;Pointers held at $4F0-$4FF and data starting at $500.
0460 209004      JSR $0490
0463 20A004      JSR $04A0
0466 2073FE      JSR $FE73
0469 20FAFD      JSR $FDFA      ;get pre-set number
046C A501        LDA $0001
046E 38          SEC
046F E930        SBC #$0030
0471 AA          TAX
0472 BDF004      LDA $04F0,X      ;load start address offset
0475 AA          TAX
0476 BD0005      LDA $0500,X      ;load values
0479 C9FF        CMP #$00FF      ; look for end of data string.
047B F0E9        BEQ $0466
047D 8540        STA $0040      ;store register number
047F E8          INX
0480 BD0005      LDA $0500,X
0483 8542        STA $0042      ;store data
```

```

0485 20B004 JSR $04B0
0488 20E004 JSR $04E0
048B E8 INX
048C D0E8 BNE $0476

0490 A900 LDA #$0000 ;Set Port B (REG) to outputs
0492 8D2FBF STA $BF2F
0495 A9FF LDA #$00FF
0497 8D2EBF STA $BF2E ;set DDR to outputs
049A A904 LDA #$0004
049C 8D2FBF STA $BF2F ;access Output Reg
049F 60 RTS

04A0 A900 LDA #$0000 ;Set Port A (DAT) to outputs
04A2 8D2DBF STA $BF2D
04A5 A9FF LDA #$00FF
04A7 8D2CBF STA $BF2C
04AA A904 LDA #$0004
04AC 8D2DBF STA $BF2D
04AF 60 RTS

04B0 A903 LDA #$0003 ;Latch
04B2 8D2EBF STA $BF2E
04B5 A540 LDA $0040
04B7 8D2CBF STA $BF2C
04BA A900 LDA #$0000
04BC 8D2EBF STA $BF2E
04BF 60 RTS

04C0 A900 LDA #$0000 ;Set Port A (REG) to inputs
04C2 8D2DBF STA $BF2D
04C5 8D2CBF STA $BF2C
04C8 A904 LDA #$0004
04CA 8D2DBF STA $BF2D
04CD 60 RTS

04D0 A901 LDA #$0001 ;Read DAT (store in $40)
04D2 8D2EBF STA $BF2E
04D5 AD2CBF LDA $BF2C
04D8 8541 STA $0041
04DA A900 LDA #$0000
04DC 8D2EBF STA $BF2E
04DF 60 RTS

04E0 A542 LDA $0042 ;Write DAT (contents of $41)
04E2 8D2CBF STA $BF2C
04E5 A902 LDA #$0002
04E7 8D2EBF STA $BF2E
04EA A900 LDA #$0000
04EC 8D2EBF STA $BF2E
04EF 60 RTS

04F0 00 11 22 31 3C 00 00 00 ;offset pointers for data held at $500 ->
04F8 00 00 00 00 00 00 00 00
0500 01 0A 07 3E 08 10 09 00 ;REG/DAT data
0508 0A 00 0B 00 0C 04 0D 08
0510 FF 02 10 07 3D 08 00 09
0518 10 0A 00 0B 40 0C 01 0D
0520 08 FF 07 07 08 10 09 10
0528 0A 10 0B 00 0C 10 0D 09
0530 FF 04 48 07 3B 08 00 09
0538 00 0A 0F FF 08 00 09 00
0540 0A 00 FF 00 00 00 00 00

```