

```

10 REM *** (c) 1985 S.POTTER ***
20 REM *** LOUD BOB SOFTWARE ***
30 MODE 1: PEN 1: MEMORY 39999
40 PRINT* Please Wait - poking...
50 FOR T= 40000 TO 40500 STEP 8
60 READ ADDR
70 FOR S=0 TO 7: READ DAT
80 SUM=SUM+DAT:POKE ADDR+S,DAT:NEXT
90 READ CKSUM:IF CKSUM<>SUM THEN GOTO 150
100 SUM=0:NEXT:PRINT"DATA CORRECT!":MODE 2
110 PRINT:PRINT"Type CALL 40000 [enter] to initiate"
120 PRINT:PRINT"Type {CLOCKON,[ hour ,minute,[second]]}:PRINT"no parameters t
o display clock":PRINT"hours,minutes, optional seconds to reset time and disp
lay clock":PRINT" {CLOCKOFF to hide clock display (still working!)"
130 PRINT:PRINT" {ALARMON,[hours,minutes] to switch alarm on and set time
for alarm *:PRINT" {ALARKOFF to switch alarm off*:PRINT"Press CTRL (P) t
o stop the noise!"
140 END
150 PRINT "ERROR IN LINE - *;ADDR:STOP
40000 DATA 40000, 62, 201, 50, 64, 156, 175, 50, 0, 758
40008 DATA 40008, 158, 50, 255, 157, 50, 254, 157, 62, 1143
40016 DATA 40016, 15, 33, 20, 158, 205, 191, 188, 1, 811
40024 DATA 40024, 122, 156, 33, 250, 157, 205, 209, 188, 1320
40032 DATA 40032, 33, 243, 157, 6, 129, 14, 0, 17, 599
40040 DATA 40040, 179, 156, 205, 239, 188, 33, 237, 157, 1393
40048 DATA 40048, 17, 50, 0, 1, 50, 0, 205, 233, 556
40056 DATA 40056, 188, 201, 139, 156, 195, 74, 157, 195, 1305
40064 DATA 40064, 63, 157, 195, 133, 157, 195, 128, 157, 1185
40072 DATA 40072, 195, 213, 157, 67, 76, 79, 67, 75, 929
40080 DATA 40080, 79, 206, 67, 76, 79, 67, 75, 79, 728
40088 DATA 40088, 70, 198, 65, 76, 65, 82, 77, 79, 712
40096 DATA 40096, 206, 65, 76, 65, 82, 77, 79, 70, 720
40104 DATA 40104, 198, 67, 76, 79, 67, 75, 65, 68, 695
40112 DATA 40112, 202, 0, 243, 33, 254, 157, 17, 5, 911
40120 DATA 40120, 158, 6, 3, 52, 26, 190, 48, 5, 488
40128 DATA 40128, 54, 0, 35, 52, 43, 19, 35, 16, 254
40136 DATA 40136, 243, 205, 26, 157, 205, 209, 156, 251, 1452
40144 DATA 40144, 201, 58, 3, 158, 254, 0, 200, 62, 936
40152 DATA 40152, 7, 205, 180, 187, 50, 9, 158, 62, 858
40160 DATA 40160, 30, 205, 90, 187, 58, 0, 158, 205, 933
40168 DATA 40168, 7, 157, 62, 58, 205, 90, 187, 58, 824

```

```

40176 DATA 40176, 255, 157, 205, 7, 157, 62, 58, 205, 1106
40184 DATA 40184, 90, 187, 58, 254, 157, 205, 7, 157, 1115
40192 DATA 40192, 58, 9, 158, 205, 180, 187, 201, 14, 1012
40200 DATA 40200, 47, 12, 214, 10, 48, 251, 198, 58, 838
40208 DATA 40208, 71, 121, 205, 90, 187, 120, 205, 90, 1089
40216 DATA 40216, 187, 201, 58, 4, 158, 254, 0, 200, 1062
40224 DATA 40224, 58, 254, 157, 254, 0, 192, 58, 255, 1228
40232 DATA 40232, 157, 71, 58, 2, 158, 184, 192, 58, 880
40240 DATA 40240, 0, 158, 71, 58, 1, 158, 184, 192, 822
40248 DATA 40248, 33, 10, 158, 205, 170, 188, 201, 243, 1208
40256 DATA 40256, 175, 50, 3, 158, 205, 181, 157, 195, 1124
40264 DATA 40264, 207, 156, 243, 254, 0, 40, 38, 254, 1192
40272 DATA 40272, 3, 202, 91, 157, 254, 2, 194, 166, 1069
40280 DATA 40280, 157, 24, 10, 221, 126, 0, 50, 254, 842
40288 DATA 40288, 157, 221, 35, 221, 35, 221, 126, 0, 1016
40296 DATA 40296, 50, 255, 157, 221, 35, 221, 35, 221, 1195
40304 DATA 40304, 126, 0, 50, 0, 158, 62, 7, 50, 453
40312 DATA 40312, 3, 158, 205, 90, 187, 195, 207, 156, 1201
40320 DATA 40320, 175, 50, 4, 158, 201, 243, 254, 0, 1085
40328 DATA 40328, 40, 17, 254, 2, 194, 166, 157, 221, 1051
40336 DATA 40336, 126, 2, 50, 1, 158, 221, 126, 0, 684
40344 DATA 40344, 50, 2, 158, 62, 7, 50, 4, 158, 491
40352 DATA 40352, 205, 90, 187, 195, 207, 156, 33, 28, 1101
40360 DATA 40360, 158, 126, 35, 205, 90, 187, 254, 10, 1065
40368 DATA 40368, 32, 247, 195, 207, 156, 62, 7, 205, 1111
40376 DATA 40376, 180, 187, 50, 9, 158, 62, 30, 205, 881
40384 DATA 40384, 90, 187, 33, 42, 158, 126, 35, 205, 876
40392 DATA 40392, 90, 187, 254, 10, 32, 247, 58, 9, 887
40400 DATA 40400, 158, 205, 180, 187, 201, 243, 254, 1, 1429
40408 DATA 40408, 194, 166, 157, 33, 237, 157, 221, 126, 1291
40416 DATA 40416, 0, 95, 22, 0, 79, 6, 0, 205, 407
40424 DATA 40424, 233, 188, 195, 207, 156, 0, 0, 0, 979
40432 DATA 40432, 0, 0, 0, 0, 0, 0, 0, 0, 0
40440 DATA 40440, 0, 0, 0, 0, 0, 0, 0, 0, 0
40448 DATA 40448, 0, 0, 0, 0, 0, 59, 59, 23, 141
40456 DATA 40456, 0, 0, 7, 0, 15, 60, 0, 0, 82
40464 DATA 40464, 15, 0, 8, 0, 130, 10, 5, 1, 169
40472 DATA 40472, 10, 251, 1, 0, 13, 66, 97, 100, 538
40480 DATA 40480, 32, 67, 111, 109, 109, 97, 110, 100, 735
40488 DATA 40488, 13, 10, 32, 32, 32, 32, 32, 32, 215
40496 DATA 40496, 32, 32, 10, 0, 0, 0, 0, 0, 74

```



```

10 org 40000
20 ent $
30 entry: ld a,#c9
40 ld (entry),a ; stop re initialise
50 xor a
60 ld (hour),a ; reset time
70 ld (min),a
80 ld (sec),a
90 ld a,15
100 ld hl,ent
110 call #bcbf ; init tone envelope
120 ;
130 ld bc,contab ; address of routine table
140 ld hl,wkspc ; address of workspace
150 call #bcd1 ; init new commands
160 ;
170 event: ld hl,tikblk+6 ; address of workspace
180 ld b,129 ; event class
190 ld c,0 ; rom select
200 ld de,clock ; address of interrupt routine
210 call #bcef ; init event
220 ;
230 addtk: ld hl,tikblk ; address of workspace
240 ld de,50
250 ld bc,50
260 call #bce9 ; add to interrupt path
270 ret
280 ;
290 contab: defw names ; rsx routines
300 jp ckon
310 jp ckoff
320 jp alon
330 jp aloff
340 jp adj
350 ;
360 names: defw "CLOCK0" ; command name table
370 defb "N"+#00
380 defw "CLOCKOF"
390 defb "F"+#00
400 defw "ALARM0"
410 defb "N"+#00
420 defw "ALARMOF"
430 defb "F"+#00
440 defw "CLOCKRD"
450 defb "J"+#00
460 defb 0
470 ;
480 clock: equ $ ; interrupt driven routine
490 di
500 ld hl,sec
510 ld de,constants
520 ld b,#03
530 inc (hl)
540 counta: ld a,(de)
550 cp (hl)
560 jr nc,countb
570 ld (hl),00
580 inc hl
590 inc (hl)
600 dec hl
610 countb: inc de
620 inc hl
630 djnz counta ; increment variables
640 call atest ; test for alarm
650 call Print ; and Print them
660 finish: ei
670 ret ; back to whence we came
680 ;
690 Print: equ $
700 ld a,(to99le)
710 cp 0 ; is clock on?
720 ret z ; return if not
730 ld a,7
740 call #bbb4 ; swap text streams
750 ld (stream),a ; save old stream
760 ld a,30
770 call #bb5a
780 ;
790 ld a,(hour)
800 call decimal ; Print in decimal
810 ld a,#3a
820 call #bb5a
830 ld a,(min)
840 call decimal
850 ld a,#3a
860 call #bb5a
870 ld a,(sec)
880 call decimal
890 ;
900 ld a,(stream) ; return text stream
910 call #bbb4
920 ret
930 ;
940 decima: ld c,47
950 split: inc c
960 sub 10
970 jr nc,split
980 add a,50
990 ld b,a
1000 ld a,c
1010 call #bb5a
1020 ld a,b
1030 call #bb5a
1040 ret
1050 ;
1060 atest: ld a,(ato99le) ; is alarm on?
1070 cp 0
1080 ret z ; return if not
1090 ld a,(sec)
1100 cp 0
1110 ret nz
1120 ld a,(min)
1130 ld b,a
1140 ld a,(amin)
1150 cp b
1160 ret nz
1170 ld a,(hour)
1180 ld b,a
1190 ld a,(ahour)
1200 cp b
1210 ret nz
1220 sound: ld hl,wowow ; sound alarm!!!

```

```

1230 call #bcaa
1240 ret
1250 ;
1260 ckoff: di ; switch clock off
1270 xor a
1280 ld (to99le),a
1290 call blank
1300 jp finish
1310 ;
1320 ckon: di ; time set + on
1330 cp 0
1340 jr z,con
1350 cp 3
1360 jp z,setsec
1370 cp 2
1380 jp nz,error ; syntax error
1390 jr sethm
1400 setsec: ld a,(ix)
1410 ld (sec),a
1420 inc ix
1430 inc ix
1440 sethm: ld a,(ix)
1450 ld (min),a
1460 inc ix
1470 inc ix
1480 ld a,(ix)
1490 ld (hour),a
1500 con: ld a,7
1510 ld (to99le),a ; beep and switch on
1520 call #bb5a
1530 jp finish
1540 ;
1550 aloff: xor a ; switch alarm off
1560 ld (ato99le),a
1570 ret
1580 ;
1590 alon: di ; alarm time set
1600 cp 0
1610 jr z,aon
1620 cp 2
1630 jp nz,error ; syntax error
1640 ld a,(ix+2) ; set the alarm time
1650 ld (ahour),a
1660 ld a,(ix+0)
1670 ld (amin),a
1680 aon: ld a,7
1690 ld (ato99le),a ; beep and switch on
1700 call #bb5a
1710 jp finish
1720 ;
1730 error: ld hl,message ; Print syntax error
1740 erPrint: ld a,(hl)
1750 inc hl
1760 call #bb5a
1770 cp 10
1780 jr nz,erPrint
1790 jp finish
1800 ;
1810 blank: ld a,7 ; swap streams
1820 call #bbb4
1830 ld (stream),a
1840 ld a,30
1850 call #bb5a
1860 ld hl,space
1870 blPrint: ld a,(hl) ; blank out last clock
1880 inc hl
1890 call #bb5a
1900 cp 10
1910 jr nz,blPrint
1920 ld a,(stream) ; swap streams
1930 call #bbb4
1940 ret
1950 ;
1960 adj: di ; adjust timekeeping
1970 cp 1
1980 jp nz,error
1990 ld hl,tikblk
2000 ld a,(ix+0)
2010 ld e,a
2020 ld d,0
2030 ld c,a
2040 ld b,0
2050 call #bce9 ; reinit new interrupt
2060 jp finish
2070 ;
2080 tikblk: defs 13 ; space for interrupt block
2090 wkspc: defs 4 ; space for rsx block
2100 sec: defb 0 ; time variables
2110 min: defb 0
2120 hour: defb 0
2130 ahour: defb 0
2140 amin: defb 0
2150 to99le: defb 0 ; on/off switches
2160 ato99le: defb 0
2170 const: defb #3b ; time max values
2180 defb #3b
2190 defb #17
2200 defb 0
2210 stream: defb 0 ; text stream
2220 wowow: defb #7 ; sound command
2230 defb 0
2240 defb 15
2250 defw 60
2260 defb 0
2270 defb #f
2280 defw #000
2290 defb 0
2300 ent: defb 130 ; tone envelope
2310 defb 10
2320 defb 5
2330 defb 1
2340 defb 10
2350 defb -5
2360 defb 1
2370 defb 0
2380 messag: defb 13 ; syntax error message
2390 defw "Bad Command"
2400 defb 13,10
2410 space: defw " "
2420 defb 10

```