

```

10 ' KEY_DEF v 2.02
20 '
30 ' Dick Ruck - Feb 1985
40 '
50 '
60 ' remove symbol buffer
70 '
80 SYMBOL AFTER 256
90 '
100 ' allocate 1024 bytes for function k
ey buffer (&a800-&abff)
110 ' plus 256 bytes for m/code routines
(&a700-&a7ff)
120 ' note if disable CTRL-SHIFT-ESC opt
ion used then &a7fb-&a7ff should NOT be
used for anything else!
130 '
140 MEMORY &A6FF
150 '
160 ' this routine allocates 1k memory t
o keyboard buffer
170 ' uses KM_EXP_BUFFER - jumpblock &bb
15
180 '
190 RESTORE 300
200 destination=&A700
210 FOR offset=0 TO 9
220 READ byte
230 POKE destination + offset,byte
240 NEXT
250 '
260 ' de = start address of buffer,
hl = length
270 '
280 ' ld de,&a800 : ld hl,&0400:
call &bb15 :ret
290 '
300 DATA &11, &0,&a8, &21, &0,&4,
&cd, &15,&bb,&c9
310 '
320 CALL destination
330 '
340 'set up expansion strings and key de
finitions for keywords
350 '
360 RESTORE 460
370 FOR expkey=128 TO 147
380 READ f$,k,l,u
390 KEY expkey,f$
400 KEY DEF k,l,l,u,expkey
410 NEXT
420 '
430 ' keywords generated by CTRL-key of
initial letter
440 ' except sYmbol, l\st and goto>
450 '
460 DATA "asc(",69,&61,&41
470 DATA "border ",54,&62,&42
480 DATA "chr$(",62,&63,&43
490 DATA "draw ",61,&64,&44
500 DATA "else ",58,&65,&45
510 DATA "for ",53,&66,&46
520 DATA "gosub ",52,&67,&47
530 DATA "goto ",31,&2e,&3e
540 DATA "hex$(",44,&68,&48
550 DATA "left$(",36,&6c,&4c
560 DATA "list ",22,&5c,&60
570 DATA "mid$(",38,&6d,&4d
580 DATA "next",46,&6e,&4e
590 DATA "paper£",27,&70,&50
600 DATA "right$(",50,&72,&52
610 DATA "sound ",60,&73,&53
620 DATA "symbol ",43,&79,&59
630 DATA "then ",51,&74,&54
640 DATA "upper$(",42,&75,&55

```

```

650 DATA "windowf",59,&77,&57
660 ?
670 ? set up top row numeric keys as fun
ctions
680 ?
690 RESTORE 760
700 FOR j=21 TO 30
710 READ f$,k,l,u
720 KEY 127+j,f#+CHR$(13)
730 KEY DEF k,0,l,u,127+j
740 NEXT
750 ?
760 DATA "mode 1",64,&31,&21
770 DATA "mode 2",65,&32,&22
780 DATA "ink 0,0:ink 1,13:border 0:pape
r 0:pen 1:mode 2",57,&33,&23
790 DATA "list",56,&34,&24
800 DATA "?int(time/18000)",49,&35,&25
810 DATA "?hex$(himem),hex$(fre(0)),hime
m-fre(0)-370",48,&36,&26
820 DATA "for j=0 to 255:? j chr$(32)chr
$(1)chr$(j),:next",41,&37,&27
830 DATA "run",33,&39,&29
840 DATA "mode 0",32,&30,&5f
850 DATA "cls",16,16,16
860 ?
870 ? following prints contents of funct
ion keys
880 ?
890 progf$="a=&a800:FOR k=1 TO 32:l=PEEK
(a):PRINT k+127,,:FOR c=a+1 TO a+l:PRINT
CHR$(1)CHR$(PEEK(c)):NEXT:PRINT:a=a+l+
1:NEXT"
900 ?
910 ? following prints real time clock
920 ?
930 progf$="t=time:h=t/108e4:hh=int(h):m
=(h-hh)*60:mm=int(m):ss=int((m-mm)*60):c
ls:? hh chr$(58) mm chr$(58) ss"+CHR$(13)
)
940 ?
950 KEY 158,"run"+CHR$(&22)+CHR$(13)
960 KEY DEF 40,0,&38,&28,158
970 KEY 159,progf#+CHR$(13)
980 KEY DEF 25,0,&2D,&3D,159
990 KEY 152,progf$
1000 ?
1010 ?redefine keypad
1020 ?
1030 RESTORE 1090
1040 FOR j=1 TO 11
1050 READ k,m
1060 KEY DEF k,1,m,m,m
1070 NEXT
1080 ?
1090 DATA 13,&31
1100 DATA 14,&32
1110 DATA 5,&33
1120 DATA 20,&34
1130 DATA 12,&35
1140 DATA 4,&36
1150 DATA 10,&37
1160 DATA 11,&38
1170 DATA 3,&39
1180 DATA 15,&30
1190 DATA 7,&2e
1200 ?
1210 ? replace RUN" on small ENTER key
1220 ?
1230 KEY DEF 6,1,&D,&D,158
1240 ?
1250 ? short routine to set TIME to valu
es in de hl
1260 ? uses KL_TIME_SET - jumpblock &bd1

```

```

1270 ?
1280 RESTORE 1380
1290 FOR offset=0 TO 9
1300 READ byte
1310 POKE destination + offset,byte
1320 NEXT
1330 ?
1340 ? dehl contain real time in 1/300 s
ec d=msb l=lsb
1350 ? need to get time from keyboard an
d poke to de hl
1360 ?
1370 ? ld de,&???? : ld hl,&????:
call &bd10
1380 DATA &11, &0,&0, &21, &0,&0,
&cd, &10,&bd,&c9
1390 ?
1400 INPUT "Enter: HOURS";hh
1410 INPUT " MINS ";mm
1420 INPUT " SECS ";ss
1430 ticks=300*(ss+60*mm+3600*hh)
1440 ?
1450 ? MOD division with large real numb
ers results in overflow
1460 ? therefore define a Modulus Divisi
on function
1470 ?
1480 DEF FNmd(x,y)=x-INT(x/y)*y
1490 b0 = FNmd(ticks,256)
1500 b1 = FNmd(INT(ticks/256),256)
1510 b2 = FNmd(INT(ticks/(256*256)),256)
1520 b3 = INT(ticks/(256*256*256))
1530 POKE destination + 2,b3
1540 POKE destination + 1,b2
1550 POKE destination + 5,b1
1560 POKE destination + 4,b0
1570 CALL destination
1580 ?
1590 ? disable CTRL-SHIFT-ESC
1600 ?
1610 INPUT "Do you want to disable CTRL-
SHIFT-ESC";a$
1620 IF LEFT$(UPPER$(a$),1)<>"Y" GOTO 19
10
1630 ?
1640 ? this routine intercepts calls to
KM_TEST_BREAK - jumpblock &bdee
1650 ? register C contains state of SHIF
T & CTRL keys - this resets C to 0
1660 ? thus system sees ESC key only pre
ssed.
1670 ?
1680 destination=&A7FB
1690 RESTORE 1760
1700 FOR offset=0 TO 4
1710 READ byte
1720 POKE destination + offset,byte
1730 NEXT
1740 ?
1750 ? ld c ,&0 : jp &????
1760 DATA &e ,&0 , &c3 ,&0 ,&0
1770 ?
1780 ? patch routine to KM_TEST_BREAK
1790 ?
1800 POKE destination+3,PEEK(&BDEF)
1810 POKE destination+4,PEEK(&BDF0)
1820 ?
1830 ? patch jumpblock to this routine
1840 ?
1850 POKE &BDEF,&FB
1860 POKE &BDF0,&A7
1870 ?

```

1880 ?  
1890 ? recover default symbols  
1900 ?  
1910 SYMBOL AFTER 240  
1920 ?  
1930 ? set cassette speed to fast saves  
1940 ?  
1950 SPEED WRITE 1  
1960 ?  
1970 ? favourite edit mode (White on black)  
1980 ?  
1990 INK 0,0 : INK 1,13 : MODE 2 : BORDER 0 : PAPER 0 : PEN 1  
2000 ?  
2010 ? erase program ?  
2020 ?  
2030 ? new  
2040 ?  
2050 END