

```

10 MEMORY &3FFF [758]
20 FOR adr=&4000 TO &4200 STEP 8 [1039]
30   FOR i=adr TO adr+ 7 [1011]
40     READ b$ [315]
50     byte=VAL("&" + b$) [465]
60     POKE i,byte [294]
70   NEXT i [375]
80 NEXT adr [547]
90 END [110]
100 DATA 3E,02,CD,0E,BC,21,00,C0 [1720]
110 DATA 22,32,40,3E,01,32,34,40 [969]
120 DATA 3E,01,32,35,40,21,29,40 [1023]
130 DATA 11,36,40,06,81,0E,00,CD [1417]
140 DATA D7,BC,21,0A,01,CD,75,BB [1802]
150 DATA C9,00,00,00,00,00,00,00 [1564]
160 DATA 00,00,00,00,00,00,3A,35 [1040]
170 DATA 40,47,3A,34,40,80,FE,4C [1052]
180 DATA CA,64,40,FE,00,CA,64,40 [1338]
190 DATA 32,34,40,2A,32,40,C5,F5 [1262]
200 DATA E5,AF,CD,6F,40,E1,F1,C1 [648]
210 DATA 7D,80,6F,22,32,40,3E,FF [1963]
220 DATA CD,6F,40,C9,3A,35,40,EE [683]
230 DATA FE,32,35,40,C3,36,40,06 [1238]
240 DATA 08,C5,F5,E5,06,06,77,23 [758]
250 DATA 10,FC,E1,CD,26,BC,F1,C1 [908]
260 DATA 10,EF,C9,00,00,00,00,00 [1211]
270 DATA 00,00,00,00,00,00,00,00 [1005]
280 DATA 00,00,00,00,00,00,00,00 [1005]
290 DATA 00,00,00,00,00,00,00,00 [1005]
300 DATA 00,00,00,00,00,00,00,00 [1005]
310 DATA 00,00,00,00,00,00,00,00 [1005]
320 DATA 00,00,00,00,00,00,00,00 [1005]
330 DATA 00,00,00,00,00,00,00,00 [1005]
340 DATA 00,00,00,00,00,00,00,00 [1005]
350 DATA 00,00,00,00,00,00,00,00 [1005]
360 DATA 00,00,00,00,00,00,00,00 [1005]
370 DATA 00,00,00,00,00,00,00,00 [1005]
380 DATA 00,00,00,00,00,00,00,00 [1005]
390 DATA 00,00,00,00,00,00,00,00 [1005]
400 DATA 00,00,00,00,00,00,00,00 [1005]
410 DATA 00,00,00,00,00,00,00,00 [1005]
420 DATA 21,29,40,CD,DD,BC,C9,00 [745]
430 DATA 00,00,00,00,00,00,00,00 [1005]
440 DATA 00,00,00,00,00,00,00,00 [1005]
450 DATA 00,00,00,00,00,00,00,00 [1005]
460 DATA 00,00,00,00,00,00,00,00 [1005]
470 DATA 00,00,00,00,00,00,00,00 [1005]
480 DATA 00,00,00,00,00,00,00,00 [1005]
490 DATA 00,00,00,00,00,00,00,00 [1005]
500 DATA 00,00,00,00,00,00,00,00 [1005]
510 DATA 00,00,00,00,00,00,00,00 [1005]
520 DATA 00,00,00,00,00,00,00,00 [1005]
530 DATA 00,00,00,00,00,00,00,00 [1005]
540 DATA 00,00,00,00,00,00,00,00 [1005]
550 DATA 00,00,00,00,00,00,00,00 [1005]
560 DATA 00,00,00,00,00,00,00,00 [1005]
570 DATA 00,00,00,00,00,00,00,00 [1005]
580 DATA 00,00,00,00,00,00,00,00 [1005]
590 DATA 00,00,00,00,00,00,00,00 [1005]
600 DATA 00,00,00,00,00,00,00,00 [1005]
610 DATA 00,00,00,00,00,00,00,00 [1005]
620 DATA 00,00,00,00,00,00,00,00 [1005]
630 DATA 00,00,00,00,00,00,00,00 [1005]
640 DATA 00,00,00,00,00,00,00,00 [1005]
650 DATA 00,00,00,00,00,00,00,00 [1005]
660 DATA 00,00,00,00,00,00,00,00 [1005]
670 DATA 00,00,00,00,00,00,00,00 [1005]
680 DATA 00,00,00,00,00,00,00,00 [1005]
690 DATA 00,00,00,00,00,00,00,00 [1005]
700 DATA 00,00,00,00,00,00,00,00 [1005]
710 DATA 00,00,00,00,00,00,00,00 [1005]
720 DATA 00,00,00,00,00,00,00,00 [1005]
730 DATA 00,00,00,00,00,00,00,00 [1005]
740 DATA 00,00,00,00,00,00,00,00 [1005]

```

```

10 MEMORY &3FFF [758]
20 FOR adr=&4000 TO &4200 STEP 8 [1039]
30   FOR i=adr TO adr+ 7 [1011]
40     READ b$ [315]
50     byte=VAL("&" + b$) [465]
60     POKE i,byte [294]
70   NEXT i [375]
80 NEXT adr [547]
90 END [110]
100 DATA AF,32,48,40,21,49,40,11 [1267]
110 DATA 15,40,06,81,0E,00,CD,19 [1324]
120 DATA BD,CD,E0,BC,C9,3A,48,40 [1069]
130 DATA 3C,32,48,40,FE,06,C2,2C [1409]
140 DATA 40,AF,32,48,40,21,52,40 [767]
150 DATA CD,36,40,C9,FE,03,C0,21 [1453]
160 DATA 62,40,CD,36,40,C9,06,10 [1582]
170 DATA 0E,00,C5,06,7F,7E,23,ED [1581]
180 DATA 49,ED,79,C1,0C,10,F3,C9 [1824]
190 DATA 02,00,00,00,00,00,81,15 [963]
200 DATA 40,00,54,44,55,5C,58,5D [1204]
210 DATA 4C,45,4D,56,46,57,5E,40 [971]
220 DATA 5F,4E,4B,43,4A,5B,59,5A [900]
230 DATA 53,42,52,4F,47,4E,5F,40 [1011]
240 DATA 5E,57,00,00,00,00,00,00 [977]
250 DATA 00,00,00,00,00,00,00,00 [1005]
260 DATA 00,00,00,00,00,00,00,00 [1005]
270 DATA 00,00,00,00,00,00,00,00 [1005]
280 DATA 00,00,00,00,00,00,00,00 [1005]
290 DATA 00,00,00,00,00,00,00,00 [1005]
300 DATA 00,00,00,00,00,00,00,00 [1005]
310 DATA 00,00,00,00,00,00,00,00 [1005]
320 DATA 00,00,00,00,00,00,00,00 [1005]
330 DATA 00,00,00,00,00,00,00,00 [1005]
340 DATA 00,00,00,00,00,00,00,00 [1005]
350 DATA 00,00,00,00,00,00,00,00 [1005]
360 DATA 00,00,00,00,00,00,00,00 [1005]
370 DATA 00,00,00,00,00,00,00,00 [1005]
380 DATA 00,00,00,00,00,00,00,00 [1005]
390 DATA 00,00,00,00,00,00,00,00 [1005]
400 DATA 00,00,00,00,00,00,00,00 [1005]
410 DATA 00,00,00,00,00,00,00,00 [1005]
420 DATA 21,49,40,CD,E6,BC,C9,00 [1350]
430 DATA 00,00,00,00,00,00,00,00 [1005]
440 DATA 00,00,00,00,00,00,00,00 [1005]
450 DATA 00,00,00,00,00,00,00,00 [1005]
460 DATA 00,00,00,00,00,00,00,00 [1005]
470 DATA 00,00,00,00,00,00,00,00 [1005]
480 DATA 00,00,00,00,00,00,00,00 [1005]
490 DATA 00,00,00,00,00,00,00,00 [1005]
500 DATA 00,00,00,00,00,00,00,00 [1005]
510 DATA 00,00,00,00,00,00,00,00 [1005]
520 DATA 00,00,00,00,00,00,00,00 [1005]
530 DATA 00,00,00,00,00,00,00,00 [1005]
540 DATA 00,00,00,00,00,00,00,00 [1005]
550 DATA 00,00,00,00,00,00,00,00 [1005]
560 DATA 00,00,00,00,00,00,00,00 [1005]
570 DATA 00,00,00,00,00,00,00,00 [1005]
580 DATA 00,00,00,00,00,00,00,00 [1005]
590 DATA 00,00,00,00,00,00,00,00 [1005]
600 DATA 00,00,00,00,00,00,00,00 [1005]
610 DATA 00,00,00,00,00,00,00,00 [1005]
620 DATA 00,00,00,00,00,00,00,00 [1005]
630 DATA 00,00,00,00,00,00,00,00 [1005]
640 DATA 00,00,00,00,00,00,00,00 [1005]
650 DATA 00,00,00,00,00,00,00,00 [1005]
660 DATA 00,00,00,00,00,00,00,00 [1005]
670 DATA 00,00,00,00,00,00,00,00 [1005]
680 DATA 00,00,00,00,00,00,00,00 [1005]
690 DATA 00,00,00,00,00,00,00,00 [1005]
700 DATA 00,00,00,00,00,00,00,00 [1005]
710 DATA 00,00,00,00,00,00,00,00 [1005]
720 DATA 00,00,00,00,00,00,00,00 [1005]
730 DATA 00,00,00,00,00,00,00,00 [1005]
740 DATA 00,00,00,00,00,00,00,00 [1005]

```

```

10 MEMORY &3FFF [758]
20 FOR adr=&4000 TO &4200 STEP 8 [1039]
30 FOR i=adr TO adr+ 7 [1011]
40 READ b$ [315]
50 byte=VAL("&"+b$) [465]
60 POKE i,byte [294]
70 NEXT i [375]
80 NEXT adr [547]
90 END [110]
100 DATA AF,32,15,40,21,16,40,11 [1242]
110 DATA 1F,40,06,81,0E,00,CD,19 [1374]
120 DATA BD,CD,E0,BC,C9,00,00,00 [1033]
130 DATA 00,00,00,00,00,00,00,3A [918]
140 DATA 15,40,3C,32,15,40,FE,06 [1883]
150 DATA C2,35,40,AF,32,15,40,3E [835]
160 DATA 01,CD,1C,BD,C9,FE,03,C0 [976]
170 DATA 3E,02,CD,1C,BD,C9,00,00 [1806]
180 DATA 00,00,00,00,00,00,00,00 [1005]
190 DATA 00,00,00,00,00,00,00,00 [1005]
200 DATA 00,00,00,00,00,00,00,00 [1005]
210 DATA 00,00,00,00,00,00,00,00 [1005]
220 DATA 00,00,00,00,00,00,00,00 [1005]
230 DATA 00,00,00,00,00,00,00,00 [1005]
240 DATA 00,00,00,00,00,00,00,00 [1005]
250 DATA 00,00,00,00,00,00,00,00 [1005]
260 DATA 00,00,00,00,00,00,00,00 [1005]
270 DATA 00,00,00,00,00,00,00,00 [1005]
280 DATA 00,00,00,00,00,00,00,00 [1005]
290 DATA 00,00,00,00,00,00,00,00 [1005]
300 DATA 00,00,00,00,00,00,00,00 [1005]
310 DATA 00,00,00,00,00,00,00,00 [1005]
320 DATA 00,00,00,00,00,00,00,00 [1005]
330 DATA 00,00,00,00,00,00,00,00 [1005]
340 DATA 00,00,00,00,00,00,00,00 [1005]
350 DATA 00,00,00,00,00,00,00,00 [1005]
360 DATA 00,00,00,00,00,00,00,00 [1005]
370 DATA 00,00,00,00,00,00,00,00 [1005]
380 DATA 00,00,00,00,00,00,00,00 [1005]
390 DATA 00,00,00,00,00,00,00,00 [1005]
400 DATA 00,00,00,00,00,00,00,00 [1005]
410 DATA 00,00,00,00,00,00,00,00 [1005]
420 DATA 21,16,40,CD,E6,BC,C9,00 [1400]
430 DATA 00,00,00,00,00,00,00,00 [1005]
440 DATA 00,00,00,00,00,00,00,00 [1005]
450 DATA 00,00,00,00,00,00,00,00 [1005]
460 DATA 00,00,00,00,00,00,00,00 [1005]
470 DATA 00,00,00,00,00,00,00,00 [1005]
480 DATA 00,00,00,00,00,00,00,00 [1005]
490 DATA 00,00,00,00,00,00,00,00 [1005]
500 DATA 00,00,00,00,00,00,00,00 [1005]
510 DATA 00,00,00,00,00,00,00,00 [1005]
520 DATA 00,00,00,00,00,00,00,00 [1005]
530 DATA 00,00,00,00,00,00,00,00 [1005]
540 DATA 00,00,00,00,00,00,00,00 [1005]
550 DATA 00,00,00,00,00,00,00,00 [1005]
560 DATA 00,00,00,00,00,00,00,00 [1005]
570 DATA 00,00,00,00,00,00,00,00 [1005]
580 DATA 00,00,00,00,00,00,00,00 [1005]
590 DATA 00,00,00,00,00,00,00,00 [1005]
600 DATA 00,00,00,00,00,00,00,00 [1005]
610 DATA 00,00,00,00,00,00,00,00 [1005]
620 DATA 00,00,00,00,00,00,00,00 [1005]
630 DATA 00,00,00,00,00,00,00,00 [1005]
640 DATA 00,00,00,00,00,00,00,00 [1005]
650 DATA 00,00,00,00,00,00,00,00 [1005]
660 DATA 00,00,00,00,00,00,00,00 [1005]
670 DATA 00,00,00,00,00,00,00,00 [1005]
680 DATA 00,00,00,00,00,00,00,00 [1005]
690 DATA 00,00,00,00,00,00,00,00 [1005]
700 DATA 00,00,00,00,00,00,00,00 [1005]
710 DATA 00,00,00,00,00,00,00,00 [1005]
720 DATA 00,00,00,00,00,00,00,00 [1005]
730 DATA 00,00,00,00,00,00,00,00 [1005]
740 DATA 00,00,00,00,00,00,00,00 [1005]

```

Listing: Mode setzen

```

10 MEMORY &3FFF [758]
20 FOR adr=&4000 TO &4200 STEP 8 [1039]
30 FOR i=adr TO adr+ 7 [1011]
40 READ b$ [315]
50 byte=VAL("&"+b$) [465]

```

```

60 POKE i,byte [294]
70 NEXT i [375]
80 NEXT adr [547]
90 END [110]
100 DATA 3E,0A,32,AA,40,3E,3B,32 [1116]
110 DATA AB,40,3E,38,32,AC,40,21 [534]
120 DATA B3,40,11,29,40,06,81,0E [1321]
130 DATA 00,CD,EF,BC,21,AD,40,11 [1767]
140 DATA 32,00,01,32,00,CD,E9,BC [1235]
150 DATA C9,CD,7F,40,CD,78,BB,E5 [1036]
160 DATA CD,93,BB,F5,3E,01,CD,90 [1952]
170 DATA BB,21,01,01,CD,75,BB,3A [1141]
180 DATA AA,40,CD,64,40,3E,3A,CD [1163]
190 DATA 5A,BB,3A,AB,40,CD,64,40 [1776]
200 DATA 3E,3A,CD,5A,BB,3A,AC,40 [1285]
210 DATA CD,64,40,F1,CD,90,BB,E1 [1128]
220 DATA CD,75,BB,C9,01,00,0A,FE [1253]
230 DATA 0A,FA,71,40,0C,90,C3,67 [1490]
240 DATA 40,F5,79,C6,30,CD,5A,BB [1355]
250 DATA F1,C6,30,CD,5A,BB,C9,3A [1616]
260 DATA AC,40,3C,32,AC,40,FE,3C [1186]
270 DATA F8,AF,32,AC,40,3A,AB,40 [1654]
280 DATA 3C,32,AB,40,FE,3C,F8,AF [1102]
290 DATA 32,AB,40,3A,AA,40,3C,32 [892]
300 DATA AA,40,FE,18,F8,AF,32,AA [643]
310 DATA 40,C9,0B,01,13,00,00,1B [678]
320 DATA 00,32,00,00,00,00,81,29 [1317]
330 DATA 40,00,00,00,00,00,00,00 [62]
340 DATA 00,00,00,00,00,00,00,00 [1005]
350 DATA 00,00,00,00,00,00,00,00 [1005]
360 DATA 00,00,00,00,00,00,00,00 [1005]
370 DATA 00,00,00,00,00,00,00,00 [1005]
380 DATA 00,00,00,00,00,00,00,00 [1005]
390 DATA 00,00,00,00,00,00,00,00 [1005]
400 DATA 00,00,00,00,00,00,00,00 [1005]
410 DATA 00,00,00,00,00,00,00,00 [1005]
420 DATA 21,AD,40,CD,EC,BC,C9,00 [827]
430 DATA 00,00,00,00,00,00,00,00 [1005]
440 DATA 00,00,00,00,00,00,00,00 [1005]
450 DATA 00,00,00,00,00,00,00,00 [1005]
460 DATA 00,00,00,00,00,00,00,00 [1005]
470 DATA 00,00,00,00,00,00,00,00 [1005]
480 DATA 00,00,00,00,00,00,00,00 [1005]
490 DATA 00,00,00,00,00,00,00,00 [1005]
500 DATA 00,00,00,00,00,00,00,00 [1005]
510 DATA 00,00,00,00,00,00,00,00 [1005]
520 DATA 00,00,00,00,00,00,00,00 [1005]
530 DATA 00,00,00,00,00,00,00,00 [1005]
540 DATA 00,00,00,00,00,00,00,00 [1005]
550 DATA 00,00,00,00,00,00,00,00 [1005]
560 DATA 00,00,00,00,00,00,00,00 [1005]
570 DATA 00,00,00,00,00,00,00,00 [1005]
580 DATA 00,00,00,00,00,00,00,00 [1005]
590 DATA 00,00,00,00,00,00,00,00 [1005]
600 DATA 00,00,00,00,00,00,00,00 [1005]
610 DATA 00,00,00,00,00,00,00,00 [1005]
620 DATA 00,00,00,00,00,00,00,00 [1005]
630 DATA 00,00,00,00,00,00,00,00 [1005]
640 DATA 00,00,00,00,00,00,00,00 [1005]
650 DATA 00,00,00,00,00,00,00,00 [1005]
660 DATA 00,00,00,00,00,00,00,00 [1005]
670 DATA 00,00,00,00,00,00,00,00 [1005]
680 DATA 00,00,00,00,00,00,00,00 [1005]
690 DATA 00,00,00,00,00,00,00,00 [1005]
700 DATA 00,00,00,00,00,00,00,00 [1005]
710 DATA 00,00,00,00,00,00,00,00 [1005]
720 DATA 00,00,00,00,00,00,00,00 [1005]
730 DATA 00,00,00,00,00,00,00,00 [1005]
740 DATA 00,00,00,00,00,00,00,00 [1005]

```

Listing: Uhr

#### Listing: Sprite-Bewegung

```

ORG &4000:
LD a,2:
call &bc0e : Mode 2
LD hl,&c000 : Position

```

```

LD (adr),hl      : d. Sprite
LD a,1           :
LD (zaehler),a   :
LD a,1           :
LD (richtung),a  :
LD hl,block       : Adresse d.
LD de,move        : Unterprgs.
LD b,&81          : Multif.
LD c,&00          : ROM-Select
call &bcd7        : Int. einh.
LD hl,&010a       :
call &bb75        :
RET              :

block ds 9        :
adr ds 2          :
zaehler db 0      :
richtung db 0     :

move LD a,(richtung) : Unterprg.
LD b,a           : (Bewegen)
LD a,(zaehler)   :
add a,b          :
cp 76            :
jp z,wechsel     :
cp 0             :
jp z,wechsel     :
weiter LD (zaehler),a :
LD hl,(adr)      :
push bc          :
push af          :
push hl          :
xor a            :
call malen       : Sprite weg
pop hl           :
pop af           :
pop bc           :
LD a,1           :
add a,b          :
LD l,a           :
LD (adr),hl      :
LD a,255         :
call malen       : Sprite hin
ret              :
wechsel LD a,(richtung) :
xor &fe          :
LD (richtung),a  :
jp move          :

malen LD b,8      : Y-Sprite
sch3 push bc     : Unterprg.
push af          : (Malen)
push hl          :
LD b,6           :
sch4 LD (hl),a   :
inc hl           :
djnz sch4        :
pop hl           :
call &bc26        : next Line
pop af           :
pop bc           :

```

```

djnz sch3        :
ret              :

org &4100         : Interrupt
LD hl,block       : abschalten
call &bcd4        :
RET              :

```

#### Listing: Fast Ticker:

```

ORG &4000         :Split an!
xor a            :
LD (zaehler),a   :
LD hl,block       :
LD de,split       :Adr. Hauptprg.
LD b,&81          :
LD c,&00          :
call &bd19        :Framefly warten
call &bce0        :Interr. an
ret              :
:
SPLIT LD a,(zaehler) :Hauptprg.
inc a            :
LD (zaehler),a   :
cp 6             :
jp nz,weiter     :
xor a            :
LD (zaehler),a   :
LD hl,farb1       :Tab. 1.Farben
call setzen      :Farben setzen
ret              :
weiter cp 3       :Mitte setzen
ret nz           :
LD hl,farb2       :Tab. 2.Farben
call setzen      :
ret              :
:
setzen LD b,16    :Farben von 0
LD c,0           :bis 15 setzen
sch1 push bc     :
LD b,&7f          :
LD a,(hl)        :
inc hl           :
out (c),c        :Farbregister
out (c),a        :setzen
pop bc           :
inc c            :
djnz sch1        :
ret              :

zaehler db 0      :
block ds 9        :

farb1 db &54,&44,&55 :1. F-Tabelle
db &5c,&58,&5d,&4c   :
db &45,&4d,&56,&46   :
db &57,&5e,&40,&5f   :
db &4e              :

farb2 db &4b,&43,&4a :2. F-Tabelle

```

```

db &5b,&59,&5a,&53 :
db &42,&52,&4f,&47 :
db &4e,&5f,&40,&5e :
db &57 :

ORG &4100 :Split aus!
LD hl,block :
call &bce6 :Interr. aus
ret :

```

#### Listing: Setzen

```

SPLIT LD a,(zaehler) : Hauptprg.
inc a :
LD (zaehler),a :
cp 6 :
jp nz,weiter :
xor a :
LD (zaehler),a :
LD a,1 : Mode 1
call &bd1c : setzen
ret : weiter
cp 3 :Mitte setzen
ret nz :
LD a,2 : Mode 2
call &bd1c : setzen
RET : Rcksprung

```

#### Listing: Uhr

```

ORG &4000 :Uhr an!
LD a,10 :Stunde...
LD (stunden),a :
LD a,59 :Minute...
LD (minuten),a :
LD a,56 :Sekunde...
LD (sekunden),a :...stellen
LD hl,block+6 :
LD de,uhr :de,Adresse
LD b,&81 :Hauptprogr.
LD c,0 :
call &bcef :Block einr.
LD hl,block :
LD de,50 :Aufruf jede
LD bc,50 :Sekunde
call &bce9 :Los!
RET :

```

```

uhr call stellen :Uhr +1 Sek.
call &bb78 :Cursorpos.
push hl :merken
call &bb93 :Pen merken
push af :
LD a,1 :Pen 1
call &bb90 :
LD hl,&0101 :Cursor
call &bb75 :setzen
LD a,(stunden) :Stunden
call ziffern :ausgeben
LD a,58 :Doppel-

```

```

call &bb5a :punkt
LD a,(minuten) :
call ziffern :s.o.
LD a,58 :
call &bb5a :s.o.
LD a,(sekunden) :
call ziffern :s.o.
pop af :alten Pen
call &bb90 :setzen
pop hl :Cursor(alt)
call &bb75 :setzen
ret :

```

```

ziffern LD bc,&0a00 :Unterprog.

```

```

schl cp &a : (2-Ziffern)
jp m,weiter2 :
inc c :
sub b :
jp schl :
weiter2 push af :
LD a,c :
add a,48 :
call &bb5a :Print
pop af :
add a,48 :
call &bb5a :Print
ret :

```

```

stellen LD a,(sekunden) :Unterprog.
inc a : (+1 Sek.)
LD (sekunden),a :
cp 60 :
ret m :
xor a :
LD (sekunden),a :
LD a,(minuten) :
inc a :
LD (minuten),a :
cp 60 :
ret m :
xor a :
LD (minuten),a :
LD a,(stunden) :
inc a :
LD (stunden),a :
cp 24 :
ret m :
xor a :
LD (stunden),a :
ret :

```

```

stunden db 0 :Variablen
minuten db 0 :
sekunden db 0 :
block ds 13 :

```

```

org &4100 :Uhr aus!
LD hl,block :
call &bcec :
RET :

```