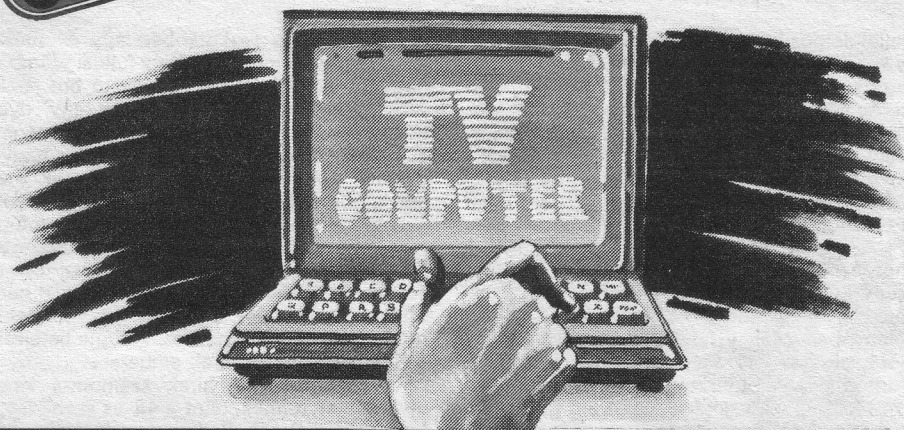




Mostanában egyre több, a TV Computerrel kapcsolatos kérdést kapunk. Némi fej-
törés után – vajon ki vála-
szolhatna ezekre? Úgy dön-
töttünk, hogy megkeressük
a legilletékesebbeket. Így
hát a gép fejlesztésében részt
vett fiatal mérnökökkel „kö-
töttünk egyezséget”, misze-
rint mostantól kezdve ellátják
olvasóinkat információkkal.

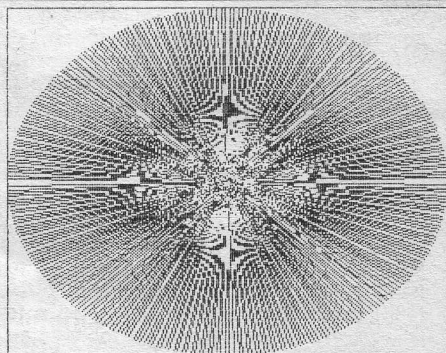
ELSŐ KÉZBŐL

A TV COMPUTER RŐL

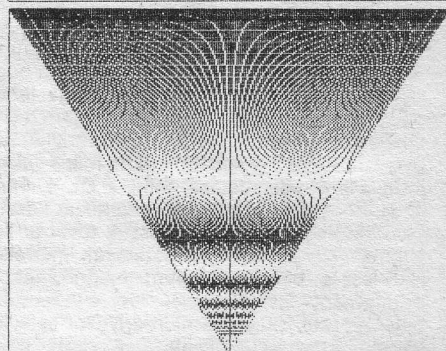
```

10 POKE 2842,127:RUN 20      ! Csak 64kB-os TVC-hez kell a 10.sor !
20 ! ----- !
30 !
40 !   TV-COMPUTER   1.sz. program !
50 !
60 !   Képernyőtartalom elmentése !
70 !       és visszaállítása.    !
80 ! ----- !
90 !
100 !
110 DATA 243      !   DI
120 DATA 62,80    !   LD   A,50h   ;page=UUVS
130 DATA 50,3,0   !   LD   (3),A
140 DATA 211,2    !   OUT (2),A
150 DATA 251      !   EI
160 DATA 201      !   RET
170 !
180 DATA 33,0,128 !   LD   HL,8000h
190 DATA 17,0,26  !   LD   DE,1A00h
200 DATA 1,0,60   !   LD   BC,3C00h
210 DATA 237,176 !   LDIR
220 DATA 201      !   RET
230 !
240 DATA 33,0,26  !   LD*  HL,1A00h
250 DATA 17,0,128 !   LD   DE,8000h
260 DATA 1,0,60   !   LD   BC,3C00h
270 DATA 237,176 !   LDIR
280 DATA 201      !   RET
300 !
310 LOMEM 86*256      ! <--- kép elmentéséhez hely !
320 GRAPHICS 4:SET MODE 3:T=0
330 FOR I=1 TO 10:READ X:VID_ON$(I)=CHR$(X):NEXT I
340 VID=2+VARPTR(VID_ON$)
350 X=USR(VID)        ! <--- VIDEO RAM bekapcsolása !
360 FOR I=1 TO 12:READ X:K_SAVE$(I)=CHR$(X):NEXT I
370 K_S=2+VARPTR(K_SAVE$) ! <--- képelmentés kezdőcíme !
380 FOR I=1 TO 12:READ X:K_LOAD$(I)=CHR$(X):NEXT I
390 K_L=2+VARPTR(K_LOAD$) ! <--- kép visszaállítás kezdőcíme !
400 !
410 SET INK 3:MODE 3
420 FOR I=0 TO PI STEP 0.02
430 RX=511*COS(I):RY=479*SIN(I)
440 PLOT 511-RX,479-RY:511+RX,479+RY
450 NEXT I
460 X=USR(K_S)        ! <--- kép elmentése !
500 !
510 T=NOT T:SET PAPER 3+T:INK 1:CLS:PRINT AT 5,1:;
520 LIST 500-.SET INK 2+T
530 FOR I=1 TO 2000:NEXT I ! <--- időzítés !
540 X=USR(K_L)        ! <--- kép visszaállítása !
550 FOR I=0 TO 1020 STEP 4
560 PLOT 512,956+T*956:I,-T*956
570 NEXT I
580 FOR I=1 TO 1000:NEXT I ! <--- időzítés !
590 IF INKEY$<>CHR$(13) THEN 510

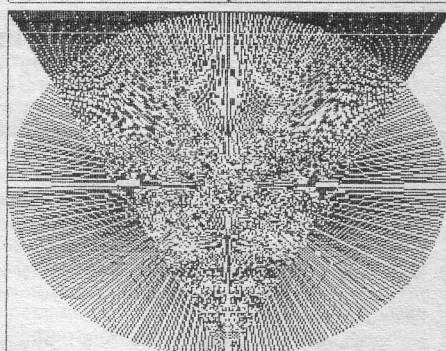
```



1. ábra



2. ábra



3. ábra



Aki már próbálta, tapasztalhatta, hogy a TV-Computer emberközeli számítógép. A Kezelési Útmutató alapján hamar megtanulható az egész képernyős programszerkesztés (full screen editor), a gép BASIC nyelvének tömör ismertetése pedig a BASIC Programozási Segédletben található. Mindez azonban kevésnek bizonyul, ha gépi kódban szeretnénk programozni. Cikksorozatunkban ismertetni fogjuk az alapvető ismereteket a TV-Computer gépi kódú programozásához.

A TV-Computer (továbbiakban: TVC) beépített programja két részre bontható, ebből az egyik rész a BASIC interpreter. A másik rész az operációs rendszer, amely az egyes logikai eszközöket kezelő, a konkrét input-output műveleteket végrehajtó rutinsomagból és az ehhez tartozó hozzáférési struktúrából áll. Ezt a második részt a gépi kódú programok is használhatják. Következő cikkben ismertetjük az operációs rendszer rutinjainak hívási módját, a rutinok rövid leírását, a hozzájuk tartozó rendszerváltozókat, az input-output perifériacímek (portok) kiosztását és funkcióit, a TVC memóriakezelését, valamint a gépi kódú programok BASIC-ba ágyazását.

Itt jegyezzük meg, hogy a közeljövőben jelennek meg a TVC hardvert és az operációs rendszert részletesen ismertető könyvek.

A TVC memóriái

A TVC-ben Z80 típusú 8 bites mikroprocesszor van, ami közvetlenül csak 64 kB memóriát tud címezni. A TVC már alapkiépítésben is ennél több memóriát tartalmaz (32 kB felhasználói RAM, 16 kB video RAM, 20 kB rendszer ROM), emellett lehetőség van további memóriák használatára is. A több memória elérése a memórialapozás (paging) segítségével történik. A hardver a processzor címtartományát négy, egyenként 16 kB-os részre osztja, ezek lesznek a memórialapok:

0. lap (PAGE0): 0-16383 azaz 0h-3FFFh.

1. lap (PAGE1): 16384-32767 azaz 4000h-7FFFh.

2. lap (PAGE2): 32768-49151 azaz 8000h-0BFFFh.

3. lap (PAGE3): 49152-65535 azaz 0C000h-0FFFFh.

Egy-egy lapon többféle memóriát is ki lehet választani. A kiválasztást egy hardver regiszter („page” regiszter) végzi, a megfelelő értéket a regiszterbe a 2. perifériacímre (port2) keresztül kell beírni. A 0. lapon négy választási lehetőségünk van (két bit), az 1. lapon egy (nincs bit), a 2. lapon kettő (egy bit), a 3. lapon pedig négy (két bit). A kiküldött byte („paging” byte) ezek alapján öt értékes bitet tartalmaz, a 0., 1. és 2. bit értéke közömbös.

Az egyes bitekkel kiválasztható memóriák az 1. táblázatban találhatók. Bővítőkártyán levő memória eléréséhez még külön ki kell választani a bővítőkártyát tartalmazó csatlakozót egy hardverregiszterrel. Ebbe a regiszterbe a 3. perifériacímre (port 3) kiküldött byte 6. és 7. bitje kerül. A kiküldendő bitekhez hozzá kell maszkolni a PORT03 rendszerváltozó 0-5. bitjét (címe: 2833=0B11h), és a rendszerváltozóba az új értéket vissza kell írni.

Alaphelyzetben, azaz a BASIC futása alatt a memória négy lapjára az alábbiak vannak kiválasztva:

PAGE0:	PAGE1:	PAGE2:	PAGE3:
USER RAM	USER RAM	USER RAM	SYSTEM ROM
Röviden: UUUS.			

Ezt a kiosztást a felhasználó által írt programok módosíthatják, sőt bizonyos esetekben módosítaniuk kell. Ilyen pl. az az eset, amikor a P3 alatti USER RAM területen USR rutinokat akarunk futtatni (ezt a RAM-területet a BASIC soha nem használja!), vagy a P2 területen levő VIDEO RAM tartalmát akarjuk kiolvasni vagy módosítani.

A memória lapozása a következő utasításokkal történik:

243	F3	DI
62.X	3E X	LD A,X
50,3,0	32 03 00	LD (3),A
211,2	D3 02	OUT (2),A
251	FB	EI

ahol X a „page” regiszter új értéke. Az új memóriaállapot beállítása az OUT utasítás hatására fog megtörténni. Vigyázni kell arra, nehogy a program „kilapozza maga alól a memóriát”, vagy ha igen, akkor ez kiszámítottan történjen.

X lehetséges értéke megállapítható az 1. táblázat alapján is, de az egyszerűség kedvéért a 2. táblázatban közöljük a lehetséges értékeket a megfelelő memóriakiválasztás rövidített jelölésével. A közömbös biteket 0 értékűnek tekintettük.

Az LD (3),A utasítás a memória 3 című rekeszébe, azaz a P SAVE rendszerváltozóba elteszi a „page” regiszter aktuális értékét. Ennek célja, hogy az operációs rendszer rutinjainak meghívása, vagy megszakítás kiszolgálása után a hívás, illetve megszakítás előtti érték visszaállítható legyen.

Az 1. sz. program rövid példát mutat a lapozásra. A 100-300 sorok tartalmazzák a három gépi kódú rutint (az adatokat egyetlen DATA után is felsorolhatjuk). A 350-es sorban a 2. lapon bekapcsoljuk a VIDEO RAM-ot. A 32 kB-os alapgépben a 2. lapon nincs USER RAM, ezért többet nem kell lapozni. A 410-460-es sorokban készítettünk egy háttérrel a későbbi rajzok számára (1. ábra), és elmentjük a BASIC program elé. Ehhez a helyet a 310-es sorban biztosítottuk. Kilistázzunk egy programkiszáradat (510-530), ezután elővesszük az elmentett háttérrel a rajzhoz (540-es sor), és rárajzoljuk a 2. ábrán látható rajzot. Az eredmény a 3. ábrán látható (550-580-as sor). Ismétlés az 510-es sortól, ha nem nyomtunk RETURN-t (590-es sor). A gyakorlatban ritkán kell az egész képernyőt elmenteni és visszaállítani, elegendő a képernyő egy részletével ezt megtenni. Ne feledjük, hogy az egész képernyő számára 15 kB helyet kell biztosítani elmentés előtt! 64 kB-os TVC-ben érdemesebb a képet a 3. lapon levő RAM-ba menteni, mert azt a BASIC interpreter soha nem használja.

Cseh Tibor

1. táblázat: A „paging” bajt bitjeinek jelentése.

"paging" bajt 4. bit 3. bit		0. lapon címezhető memória (PAGE 0)	rövid jelölés
0	0	rendszer ROM fő része (SYSTEM ROM)	S
0	1	a TVC oldalán bedugható ROM-modul (CARTRIDGE ROM)	C
1	0	felhasználói RAM első 16 kB (USER RAM)	U
1	1	felhasználói RAM negyedik 16 kB (USER RAM +)	U+
"paging" bajt		1. lapon címezhető memória (PAGE 1)	rövid jelölés
-	-	felhasználói RAM második 16 kB (USER RAM)	U
"paging" bajt 5. bit		2. lapon címezhető memória (PAGE 2)	rövid jelölés
0	0	képernyő RAM (VIDEO RAM)	V
1	1	felhasználói RAM harmadik 16 kB (USER RAM)	U
"paging" bajt 7. bit 6. bit		3. lapon címezhető memória (PAGE 3)	rövid jelölés
0	0	a TVC oldalán bedugható ROM-modul (CARTRIDGE ROM)	C
0	1	rendszer ROM fő része (SYSTEM ROM)	S
1	0	felhasználói RAM negyedik 16 kB (USER RAM +)	U+
1	1	rendszer ROM bővítés része (első 8 kB, ebből 4kB üres) (EXTENSION SYSTEM ROM) és bővítőkártyán levő RAM/ROM (alsó 8 kB) (EXPANSION CARD RAM/ROM)*	E

* Bővítőkártya kiválasztása a 3. port-ra kiírt bajt 6. és 7. bitjével történik.

2. táblázat: A memória-kiosztásokhoz tartozó „paging” bajtok.

"paging" bajt		memória-kiosztás			
decimális	hexa	P0	P1	P2	P3
0	00	S	U	V	C
64	40	S	U	V	S
128	80	S	U	V	U+
192	C0	S	U	V	E
32	20	S	U	U	C
96	60	S	U	U	S
160	A0	S	U	U	U+
224	E0	S	U	U	E
8	08	C	U	V	C
72	48	C	U	V	S
136	88	C	U	V	U+
200	C8	C	U	V	E
40	28	C	U	U	C
104	68	C	U	U	S
168	A8	C	U	U	U+
232	E8	C	U	U	E
16	10	U	U	V	C
80	50	U	U	V	S
144	90	U	U	V	U+
208	D0	U	U	V	E
48	30	U	U	U	C
112	70	*****	U	U	S
176	B0	U	U	U	U+
240	F0	U	U	U	E
24	18	U+	U	V	C
88	58	U+	U	V	S
152	98	U+	U	V	U+
216	D8	U+	U	V	E
56	38	U+	U	U	C
120	78	U+	U	U	S
184	B8	U+	U	U	U+
248	F8	U+	U	U	E

***** Aktuális memória-kiosztás BASIC alatt

Rövidítések: S - rendszer ROM fő része
C - képernyő RAM
U - felhasználói RAM
U+ - felhasználói RAM a 3. lapon
C - oldalt bedugható ROM-modul
E - rendszer ROM bővítés és bővítőkártya RAM/ROM
P0, P1, P2, P3 - 16 kB-os memórialapok