

Operációs rendszer



VIDEOTON

TV-Computer

GARAI GÉZA

TV-COMPUTER

Operációs rendszer

NOVOTRADE, 1988

Lektorálta: dr. Tagányi György
Cseh Tibor

Szerkesztette: Gergely János

A kiadásért felel RÉNYI GÁBOR, a Novotrade Rt. igazgatója
Budapest, 1987.

ISBN 963 02 5072 1

Copyright © Garai Géza, 1987.

Készült a Somogy Megyei Nyomdaipari Vállalat Kaposvári Üzemében

Felelős vezető: Mike Ferenc igazgató

TARTALOMJEGYZÉK

1. Bevezetés	9
2. Mikroprocesszor és memória	11
2.1 Mikroprocesszor	11
2.2 Memória	12
2.3 Memórialapozás	13
2.3.1 Lapozási szabályok	16
2.3.2 Lapozás SYS és EXT között	17
2.3.3 Programozási példák	17
2.4 Programmodul használata	19
2.5 Rendszer verem	20
3. Megszakításkezelés	21
3.1 Megszakításkezelő hardver	21
3.2 Megszakításkezelő szoftver	23
4. Input-output rendszer	25
4.1 Hardver struktúra	25
4.2 Funkcióhívások	28
4.2.1 Programozási példa	31
4.3 I/O hozzárendelések	33
4.3.1 Programozási példa	38
5. Rendszerváltozók	41
6. Rendszer-inicializálás	43

7. Kernel hívások	47
7.1 HI_MEM_SET	48
7.2 EXP_DIRECT	50
7.3 I/O_DIRECT	51
7.4 SLOT_NUM	52
 8. Képmegjelenítés	 53
8.1 Képmegjelenítő hardver	53
8.1.1 Video memória és képernyővezérlő	53
8.1.2 A megjelenítés folyamata	57
8.2 A 6845-ös képernyővezérlő programozása	60
8.2.1 Példák a képernyővezérlő átprogramozására	63
8.3 Video rendszerváltozók	71
8.4 Video funkciók	75
8.4.1 VID_CHOUT	75
8.4.2 VID_BKOUT	76
8.4.3 BTEXT	77
8.4.4 VMODE	77
8.4.5 CLS	78
8.4.6 SABS	79
8.4.7 BREL	80
8.4.8 BON	80
8.4.9 BOFF	81
8.4.10 FILL	81
8.4.11 DEFC	82
8.4.12 PAL	82
8.5 Programozási példák	83
 9. A billentyűzet	 85
9.1 Billentyűzet hardver	85
9.2 Billentyűzet szoftver	87
9.3 Billentyűzet funkciók	89
9.3.1 KBD_CHIN	89
9.3.2 KBD_STATUS	90
9.4 Programozási példa	90

10. Képernyőszerkesztő	93
10.1 Editor funkciók	95
10.1.1 ED_INT	95
10.1.2 ED_CHIN	95
10.1.3 ED_CHOUT	96
10.1.4 ED_BKIN	97
10.1.5 ED_BKOUT	98
10.1.6 CPOS	99
10.1.7 CFIX	99
10.2 Szerkesztési parancsok	100
10.3 Programozási példák	101
 11. Hang	 103
11.1 4 hangkezelő hardver	103
11.2 Hangkezelő szoftver	104
11.3 TONE_SET	105
 12. Nyomtatókezelés	 107
12.1 Nyomtatócsatoló hardver	107
12.2 Nyomtató szoftver	108
12.2.1 PAR_CHOUT	108
12.2.2 PAR_BKOUT	108
 13. Kazettakezelés	 109
13.1 Kazettakezelő hardver	109
13.2 Adatstruktúra a szalagon	111
13.3 Pufferelt file-ok	113
13.4 Nem pufferelt file-ok	113
13.5 Kazetta rendszerváltozók	114
13.6 Kazettafunkciók	116
13.6.1 CAS_CHIN	116
13.6.2 CAS_CHOUT	116
13.6.3 CAS_BKIN	117
13.6.4 CAS_BKOUT	117
13.6.5 CAS_OPEN	118
13.6.6 CAS_CRTE	119

13.6.7 CAS_CLOSE	119
13.6.9 CAS_VERIFY	120
13.7 Kazettahibák	121
13.8 Programozási példa	122
14. A soros vonal	125
14.1 Hardverfelépítés	125
14.2 A 8251A típusú USART programozása	127
14.3 A soros vonalat működtető szoftver	131
14.3.1 SER_CHOUT	132
14.3.2 SER_CHIN	133
14.3.3 SER_BKOUT	134
14.3.4 SER_BKIN	134
14.3.5 RS232_SET	134
14.3.6 RS232_CLOSE	135
Függelék	136

1. BEVEZETÉS

A Videoton IV-Computer BASIC-je a programozó számára lehetővé teszi a gép összes hardver egységének működtetését, de adódhatnak olyan feladatok, amelyeknél a BASIC nyelvű programozás nem ad megfelelő eredményt.

A gépi kódú programozáshoz szükséges a gép operációs rendszerének ismerete. Az operációs rendszer a felhasználó dolgát nagymértékben megkönnyíti azáltal, hogy egységes interface-t biztosít a gép fizikai és logikai eszközeit kezelő programrészekbe (eszközmeghajtókba) beépített alacsony szintű rutinokhoz.

Az operációs rendszer kezeli a megszakításokat és a memórialapozást, lehetővé teszi különböző hardver- és szoftver-eszközök hozzárendelését a logikai ki- és bemenetekhez.

Segítségével a IV-Computerhez fejlesztett interface áramkörök beilleszthetők a gép szoftver rendszerébe, lehetővé téve az új hardver-eszközök BASIC nyelvű elérését.

Ez a könyv a gépi kódú programozáshoz szükséges ismereteket tartalmazza, de a BASIC lehetőségeinek jobb kihasználásához is segítséget nyújt. Foglalkozik a memóriakiosztással, ismerteti a rendszerváltozókat, az operációs rendszer eszközmeghajtóit és azok funkcióit, az eszközök hozzárendelését a különböző I/O műveletekhez.

Annak érdekében, hogy a programozó valamennyi szükséges ismeretet egy helyen találjon meg, a könyv a hardver egységekkel is foglalkozik. Természetesen nem részletes áramköri ismertetésről van szó, elsősorban a programozáshoz szükséges tudnivalók találhatók meg ezekben a részekben.

Reméljük, hogy könyvünk segítséget nyújt a IV-Computer gépi kódú programozásához és a gép lehetőségeinek jobb kihasználásához.

A leírásban a memória és az I/O port címeket hexadecimálisan adjuk meg, de legtöbbször megadjuk a decimális értéket is. Az I/O műveletek adat-bitjeinél a 0 és 1 mellett a következő jelöléseket alkalmazzuk:

- : nem értelmezett bit, értéke közömbös,
- x: az adott kombinációban értéke közömbös,
- +: más hardver egységre vonatkozó bit, tehát a korábban kiküldött értéket ismételni kell.

Egyes esetekben a bitcsoportok bitkombinációinak hexadecimális értékét is megadjuk, ezek összeadásával is meghatározható a kiküldendő byte értéke.

2. MIKROPROCESSZOR ÉS MEMÓRIA

2.1 Mikroprocesszor

A TV-Computer Z80A típusú mikroprocesszort tartalmaz. Egy 25 MHz-es kvarc-oszcillátorból származik az összes időzítőjel:

- 12.5, 6.25 és 3.125 MHz-es rasterpont léptetőjel a háromféle grafikus üzemmódnak megfelelően
- 3.125 MHz-es CPU óra
- 3.125 MHz a hanggenerátorhoz

A 3.125 MHz-ből 320 ns-os ütemidő adódik. Lassú memória esetén lehetőség van WAIT állapot beiktatására, ez külön-külön köthető be az alábbi egységekre:

- rendszer ROM
- programmodul ROM
- az interface kártyákon található ROM

Az alkalmazott ROM áramkörök általában nem igénylik a WAIT állapot beiktatását.

A TV-Computer 64 k+ típusú továbbfejlesztett gépnél a rendszer és a programmodul ROM nem tud WAIT állapotot kérni.

A képinformációt tartalmazó videomemória kettős hozzáférésű. A képmegjelenítő áramkör meghatározott ütemben olvas a videomemóriából, ezért a CPU hozzáférést a videomemóriához szinkronizálni kell. Ez valamelyik óra-

jelfázis kinyújtásával van megoldva, tehát videomemória írás/olvasás esetén nem igaz a 320 ns-os ütemidő.

A mikroprocesszor az 1-es megszakítás módban dolgozik. A TV-Computer nem tartalmaz olyan áramköröket, amelyek más megszakítási mód használatát megengednék.

2.2 Memória

Memóriakiépítettség szempontjából a TV-Computer többféle lehet:

1. TV-Computer 32 k: A felhasználói RAM mérete 32 kbyte, ez külső modullal bővíthető 64 kbyte-ra. A videomemória 16 kbyte.
2. TV-Computer 64 k: A felhasználói RAM 64 kbyte, a videomemória 16 kbyte.
3. TV-Computer k+: Ez a változat a könyv megírásakor fejlesztés alatt lesztés alatt állt. A felhasználói RAM 64 kbyte, a videomemória is 64 kbyte, amihez 16 kbyte-os egységekben lehet hozzáférni, ez 4 kép egyidejű tárolását jelenti.

A felhasználói (USER) RAM-on és a videomemórián kívül a TV-Computer a következő memóriákat tartalmazza:

SYS	16 kbyte rendszer ROM
EXT	8 kbyte rendszer ROM kiegészítés
IOMEM	minden interface kártyán elhelyezhető 8 kbyte méretű ROM/RAM I/O memória
CART	max. 16 kbyte ROM, amely a kívülről csatlakoztatható programmodulban helyezhető el

Látható, hogy a TV-Computer a mikroprocesszor által címezhető 64 kbyte-nál jóval több memóriát kezel. A különböző memóriarészek az ún. memórialapozással érhetők el.

2.3 Memórialapozás

A mikroprocesszor címtartománya 4 db 16 kbyte-os lapra (LAP0...LAP3) van osztva.

A különböző típusú memóriák is lapszervezésűek, egy lap mérete 16 k.

A memórialapok a következők:

Az U0, U1, U2, U3 jelenti a felhasználói (USER) RAM lapjait.

A VID a 16 k-s videomemóriát, ill. a 4x16 k videomemória egy lapját jelenti.

A SYS a rendszer a ROM jelölése.

A rendszer ROM kiegészítés és a kiválasztott csatolókártya IOMEM-je egy lapnak számít, jele EXT.

A CART a programmodul ROM.

Ezeket a memóriákat a CPU különböző lapokon láthatja:

SYS: LAP0 és LAP3

U0: LAP0

U1: LAP1

U2: LAP2

U3: LAP3 és LAP0 (Az U3 → LAP0 lehetőség csak a külső modullal bővített 32 k-s gépnél és a TV-Computer 64 k+ típusú gépnél van meg.)

VID: LAP2, a TV-Computer 64 k+ típusú gépnél LAP1 is

EXT, IOMEM: LAP3
CART: LAP0, LAP3

(Az U3 → LAP0 lapozást a fejlesztés alatt álló VT-DOS operációs rendszer fogja használni. A TV-Computer 64 k típusú gépek ezt a lapozást nem tudják végrehajtani, de egy kapuáramkör utólagos beépítésével a VT-DOS rendszer futtatására alkalmassá tehetők.)

A lapozási lehetőségeket az 1. Függelék foglalja össze. A lapozás kijelölése a 2-es output port írásával lehetséges, az 1. Függelék a PORT02H adatbitjeinek értelmezését is tartalmazza.

A gép bekapcsolásakor a PORT02H valamennyi bitje 0, tehát a lapozási összetétel (LAP0...LAP3):

SYS, U1, VID, CART

A BASIC rendszer inicializálása után az alapállapot a következő:

U0, U1, U2, SYS

A 2-es port tartalma 70H.

Az ettől eltérő lapozási összetétel beállításához a 2-es portra küldendő érték az 1. Függelék segítségével határozható meg. Ehhez segítséget nyújtanak a zárójelben levő hexadecimális számok, amelyeket egyszerűen össze kell adni.

Példaként lapozzuk a videomemóriát a 2-es lapra:

LAP0:	U0	(10)
LAP1:	U1	(00)
LAP2:	VID	(00)
LAP3:	SYS	<u>(40)</u>
		50

Másik példaként rendeljük a teljes memóriához felhasználói RAM-ot:

LAP0:	U0	(10)
LAP1:	U1	(00)
LAP2:	U2	(20)
LAP3:	U3	<u>(80)</u>
		80

A memórialapozáshoz nem elég a 2-es port írása. Előfordulhat, hogy a megváltoztatott lapozási állapot alatt programmegszakítás jön létre, vagy rendszerhívást adunk ki. Mindkét esetben átíródik a lapozási összetétel. A felhasználónak a visszaállítást a P_SAVE memóriarekesz könnyíti meg (címe 3). A 2-es port írása előtt a lapozási összetételt be kell írni a P_SAVE rekeszbe. A megszakításból vagy rendszerhívásból való visszatérés esetén az operációs rendszer visszaállítja a 2-es portot a P_SAVE értéke szerint.

Ha a PORT02 a 3-as lapra az EXT-et lapozta, a 4x8 k-s IOMEM megfelelő részének kiválasztására a 3-as I/O port B7-es, B6-os bitje szolgál a megfelelő interface csatlakozóhely (CSTL0...CSTL3) kijelölésével (l. Függelék).

A 3-as port B0...B3-as bitjei más hardver egységekre vonatkoznak. A PORT03 memóriarekesz (0B11H) lehetővé teszi korábbi értékük figyelembevételét. Tartalma megegyezik a 3-as port tartalmával, ezért a 3-as port tükörképének is nevezzük.

A TV-Computer 64 k+ típusú gépnél a videomemória LAP1-re is lapozható. A PORT0FH írásával választható ki, hogy a 4 kép közül

- melyik legyen megjelenítve (B5, B4: CRT)
- melyiket lássa a CPU a 2-es lapon (B3, B2: LAP2), ha oda a PORT02 a videomemóriát jelölte ki
- melyiket lássa a CPU az 1-es lapon (B1, B0: LAP1), ha oda a PORT02 a videomemóriát jelölte ki.

A memóriakezeléssel kapcsolatban megjegyezzük, hogy a BASIC rendszer nem tudja használni az U3 memórialapot, ezért a 64 k-s gép is csak 48 k-s gépként viselkedik.

A legnagyobb használható RAM cím a HI_MEM rendszerváltozóban (0B19H) található, maximális értéke 0BFFFH (49151) a 64 k-s gépeknél, 07FFFH (32767) a 32 k-s gépeknél. A HI_MEM értékét bekapcsoláskor a memóriateszt állítja be. A felhasználó ezt kisebb értékűre is átállíthatja a későbbiekben helyfoglalás céljából.

Az U3 RAM (0C000H...0FFFFH) gépi kódú programok tárolására használható. Bekapcsoláskor ez a memóriaterület is tesztelődik, a P3RAM (0B1BH) tartalmazza a teszt eredményét (0: a memória jó).

A TV-Computer működőképességének az a feltétele, hogy legalább 16 k RAM (U0) és a 16 k videomemória hibátlan legyen.

2.3.1 LAPOZÁSI SZABÁLYOK

Az U0 lapon keresztül zajlik a rendszerhívás és -megszakítás, valamint itt található a rendszerváltozók és a verem. Ezért a LAP0-án lehetőség szerint hagyjuk meg az U0-t. Ha a LAP0-ra mégis mást lapozunk, előtte gondoskodjunk arról, hogy ne történhessen megszakítás. Funkcióhívást ebben a lapozási összetételben ne adjunk ki, veremművelet előtt a veremmutatót állítsuk át.

A P_SAVE írása előzze meg PORT02 írását.

A program alól ne lapozzuk ki a memóriát. Így a 2.3.3 pontban leírt első példaprogram nem lehet a 8000H...0BFFFH címtartományban (LAP2), hiszen ott a CPU videomemóriát látja. Érdekességgé megemlítjük, hogy a példaprogram középső (feldolgozás) része elhelyezhető lenne a videomemóriában is.

2.3.2 LAPOZÁS SYS ÉS EXT KÖZÖTT

Az operációs rendszer egy része a rendszer ROM kiegészítésében (EXT) található, ezért a működés során a SYS rendszer ROM-ban és az EXT kiegészítő ROM-ban a program váltakozva fut. Az átkapcsolásra mindkét memóriarész végén egy híd van kiépítve. Az átkapcsolás előtt a program a HL regiszterpárban helyezi el a folytatási címet. A programrészletek a következők:

Rendszer ROM

```
PUSH    AF
LD       A,0F0H ; U0, U1, U2, EXT
LD       (3),A
OUT      (2),A
POP      AF
JP       (HL)
```

Kiegészítő ROM

```
PUSH    AF
LD       A,70H ; U0, U1, U2, SYS
LD       (3),A
OUT      (2),A
POP      AF
JP       (HL)
```

2.3.3 PROGRAMOZÁSI PÉLDÁK

1. Egy gépi kódú rutin a videomemóriában akar dolgozni:

```
3E 50      LD    A,50H ; U0, U1, VID, SYS
32 03 00   LD    (3),A ; P_SAVE
D3 02      OUT   (2),A ; PORT02 írása
           .      ;
           .      ; feldolgozás
           .      ;
```

```

3E 70      LD      A,70H ; U0, U1, U2, SYS
32 03 00   LD      (3),A ; P_SAVE
D3 02      OUT     (2),A ; PORT02 írása
C9         RET

```

A P_SAVE beállítása miatt a feldolgozás alatti megszakítás vagy a programból kiadott rendszerhívás nem rontja el a lapozási összetételt.

2. Az 1-es csatlakozóba helyezett interface kártyán lévő memóriát akarjuk látni:

```

3E F0      LD      A,0F0H ; U0, U1, U2, EXT
32 03 00   LD      (3),A ; P_SAVE
D3 02      OUT     (2),A ; PORT02 írása
3A 11 0B   LD      A,(0B11H) ; PORT03 tükörképe
E6 3F      AND     3FH ; B7, B6 törlése
F6 40      OR      40H ; B7, B6: CSTL1
32 11 0B   LD      (0B11H),A ; PORT03 tükörkép új
                        értéke
D3 03      OUT     (3),A ; PORT03 írása
           .      ;
           .      ; feldolgozás
           .      ;

3E 70      LD      A,70H ; U0, U1, U2, SYS
32 03 00   LD      (3),A ; visszaállítása
C9         RET ;

```

2.4 Programmodul használata

A TV-Computer bekapcsolásakor lefutó inicializáló program megvizsgálja a programmodul tartalmát (előtte a programmodult) LAP3-ra lapozta, tehát a programmodul első címe 0C000H).

Normál esetben a programmodul első 4 byte-ján a "MOPS" szónak kell lennie (4DH, 4FH, 50H, 53H).

Ha a rendszer az első 4 byte-on megtalálta ezt az azonosítót, a 0C004H címre ugrik. A programmodulban levő programnak kell gondoskodnia arról, hogy a LAP0-ra a rendszer ROM (SYS) helyére RAM-ot (U0) lapozzon, természetesen a P_SAVE beállításával együtt. Programmodul használata esetén a rendszermegszakítás a szokásos módon fut le, rendszerhívások a programmodulból kiadhatók. A programmodulban tárolt programnak kell a megszakítást engedélyezni és a WARM_FLAG-et törölni.

Ha a programozó a programmodulban nem helyezte el a "MOPS" szócskát, a rendszer nem vesz tudomást a programmodulról. Ebben az esetben a RAM-ban elhelyezett gépi kódú programból lehet ugrani a programmodulba vagy programmodul rutinokat hívni.

```
Ugrás:          LD    A,30H  ; U0, U1, U2, CART
                  LD    (3),A
                  OUT   (2),A
                  JP    0CXXXH
```

.
.
.

Szubrutinhívás:

```
LD    A,30H  ; U0, U1, U2, CART
LD    (3),A
OUT   (2),A
CALL  0CXXXH
LD    A,70H  ; U0, U1, U2, SYS
```

```
LD      (3),A
OUT     (2),A
        .
        .
```

2.5 Rendszer verem

Az operációs rendszer 2 kbyte-ot foglal le a LAPD-án (16ABH → DEADH) a verem számára.

A nagyméretű veremre a képkitöltés (1.8 fejezet) miatt van szükség, bonyolult alakzatok kitöltése esetén.

A verem alsó határ az ST_LIMIT változóban található (0B17H, 2839). Tartalma 0F10H, tehát 100 byte biztonsági tartalék van a rendszer-megszakítások okozta verem mélyítésének fenntartva.

Az operációs rendszer nem működőképes, ha a verem a 2-es vagy a 3-as lapra van helyezve.

3. MEGSZAKÍTÁSKEZELÉS

3.1 Megszakításkezelő hardver

A TV-Computerben a programmegszakítás (interrupt, IT) oka háromféle lehet:

1. A hangkeltésre szolgáló programozott frekvenciaosztó kimenetének 1 → 0 átmenete, tehát a beállított frekvenciából adódó négyszögperiódus vége.
2. A képernyővezérlésre használt 6845-ös típusú képernyővezérlő CURSOR kimenete. (Ez a jel csak megszakítás előállítására szolgál. A képernyőn látható cursort szoftver állítja elő, tehát a képernyővezérlő CURSOR kimenetéhez nincs köze.)

Ennek a megszakítójelnek a pontos időzítését a képernyővezérlő programozása szabja meg. Alapbeállításban a megszakítást a megjelenített kép utolsó képpontjai váltják ki.

Mint később látni fogjuk, a TV-Computer képe 314 sorból áll, tehát a megszakítás $314 \cdot 64 \mu s = 20.096 \text{ ms}$ -onként következik be. Ez a megszakítás adja az időzítést a billentyűzet és a hangidőtartam számára.

A 20.096 ms megváltozhat a képernyővezérlő átprogramozása miatt, ill. ha a cursor helyett a hang megszakítást kapcsoljuk be (pl. a kazetta SAVE és LOAD parancsok esetén.)

3. Minden interface kártya tartalmazhat megszakítójel-előállító áramkört. Az interface kártyák a TV-Computer bal felső részén található csatlakozókba helyezhetők, számozásuk balról jobbra: CSTL3, CSTL2, CSTL1, CSTL0.

A hang és cursor megszakítás jeleket logikai VAGY kapcsolat fogja össze, a kétféle megszakítási ok szoftver úton nem különböztethető meg. Ebből következik, hogy a két megszakításforrás egyidejűleg nem használható, valamelyiket le kell tiltani.

A hang megszakítás letiltható az 5-ös I/O cím írásával (B5=0). A megszakítás újraengedélyezése hasonlóan lehetséges (B=1). A PORT05 is tartalmaz más hardver egységekhez tartozó biteket, tehát azt is a PORT05 memória-tükörkép figyelembevételével kell írni, címe: 0B12H (2834), 1. és 2. Függelék. Ha a hang megszakításánál egyszeri lefutású, pontos időzítést akarunk, az 5BH-s I/O port olvasásával indíthatjuk alap-helyzetből a frekvenciaosztót.

A cursor megszakítást többféleképpen is letilthatjuk a képernyővezérlő átprogramozásával. Egyik lehetőség a cursor kicímzése a megjelenített kép területéről. Javasolt módszer az R10-es regiszter B6, B5 bitjének 01 értékre állítása (1. a 2. Függelék, ill. a későbbiekben a képernyővezérlő programozását).

A cursor/hang megszakítás törlése (nyugtázása) a PORT07 írásával lehetséges, az írt adat értéke közömbös.

Az interface kártyák megszakításának engedélyezése a 0...3 csatlakozószámoknak megfelelően a PORT58H...PORT5BH outputok 7-es bitjének 1-be írásával történik. A megszakítások törlését (nyugtázását) az előbbi outputok B7=0 írása végzi, ez le is tiltja a megszakításokat addig, amíg B7=1-et nem írunk.

Az 59H-s PORT olvasásával megállapítható a megszakítási ok forrása (B4=0 cursor/hang, B3...B0=0 interface kártya 3...0).

Az interface kártyák csatlakozóira a Z80-as CPU busz /INT jele (a / jel a negálást jelenti) közvetlenül is ki van vezetve, tehát olyan interface kártya is készíthető, amelyik közvetlenül kér megszakítást. Ebben az esetben a TV-Computer normál megszakításkérő jeleit le kell tiltani, az operációs rendszer megszakításkezelő programját le kell cserélni (ez

elérhető a 38H-s RAM cím tartalmának átírásával). Természetesen ez súlyos következményekkel járhat.

3.2 Megszakításkezelő szoftver

A megszakításkezelő szoftver tudnivalóit a 3. Függelék foglalja össze.

A Z80-as CPU 1-es megszakítás módban dolgozik. Ez azt jelenti, hogy a megszakítás egy RST 38H-s szubrutinhívást vált ki és letiltja a további megszakításokat.

A 38H-s címen található program U0, U1, U2, SYS lapozást állít be és a rendszer ROM-ba ugrik, ahol a tényleges megszakításalprogram található.

A megszakításalprogram a 0B41H-s RAM címen (INT_EXIT) fejeződik be (felhasználói lapozás visszaállítása a P_SAVE-ből, megszakítás engedélyezése, majd visszatérés RET-tel a megszakított programba).

A felhasználónak lehetősége van az operációs rendszer megszakításalprogramja előtt speciális megszakításkezelő alprogramot végrehajtani a 38H-n található programrészek módosításával, de gondoskodni kell a saját megszakításkezelő alprogram után a visszatérésről, ill. a visszaugrásról az operációs rendszer megszakításalprogramjába (ügyelve a lapozásra és a regisztermentésekre).

Az operációs rendszer megszakításalprogramja gondoskodik arról, hogy az eszközök kiszolgálása a megszakítást kiváltó oknak megfelelő legyen.

1. A megszakításprogram az 59H-s port olvasásával megállapítja a megszakítás okát (1. 2. Függelék).

A különböző megszakítások kiszolgálási sorrendje: cursor/hang, CSTL0...CSTL3.

Ha a megszakítást interface kártya váltotta ki, a megszakításprogram az interface kártya IOMEM-jében található megszakításkiszolgáló rutint hívja meg.

2. Cursor/hang megszakítás esetén az INT_DES rendszerváltozó szabja meg, hogy mely eszközök legyenek kiszolgálva (3. Függelék). Alapbeállítás: OFCH, tehát a video és billentyűzet meghajtó kiszolgálása történik meg. A kiszolgálás sorrendje megfelel a bitsorrendnek (video → CSTL3).

Az eszközmegszakítást kiszolgáló rutinok hívása normál rendszerhívást jelent (0. funkció). A memórialapozást és a megszakításkérések nyugtázását az operációs rendszer kernel (mag) része végzi.

Az operációs rendszer megszakításprogramja a kérések törlése (nyugtázása) után az IRQ_STAT rendszerváltozó értékének megfelelően újra engedélyezi a megszakításokat. Tényleges engedélyezés/tiltás csak az interface kártyákra (CSTL0...CSTL3) történik, a cursor/hangra nem. Az IRQ_STAT alapértéke 1 (cursor).

Az operációs rendszer által kiszolgált minden egyes cursor/hang megszakítás alatt eggyel növekszik az INTINC két byte-os számláló, tehát az INTINC óráként használható. Ütemideje - a korábban tárgyalt korlátozásokkal - 20.096 ms.

Az INT_FLAG rendszerváltozó belső használatra van fenntartva.

4. INPUT-OUTPUT RENDSZER

A TV-Computer hardver-eszközeit kezelő alacsony szintű rutinok egy meghatározott interface-en keresztül érhetők el. Ezek a rutinok az ún. eszközmeghajtókba vannak szervezve. Az eszközmeghajtók az operációs rendszer hívásával, másnéven funkcióhívásokkal érhetők el.

4.1 Hardver struktúra

A TV-Computer input/output hardver-eszközei két csoportba oszthatók:

1. Az alaplókártyára felépített hardver-eszközök:

- képernyővezérlő (video)
- billentyűzet
- hanggenerátor
- nyomtató interface
- kazetta interface

2. Csatolókkártyák (külső interface-ek):

- soros vonal
- floppy disk csatoló
- általános célú párhuzamos interface
- stb.

Mint korábban már szó volt róla, az interface kártyák tartalmazhatnak ROM/RAM memóriát. Az operációs rendszer csak akkor kezeli a csato-

lőkártyát, ha azon ROM memória van, és a ROM memória meghatározott szabályok szerint tartalmazza az illető kártyát kezelő eszközmeghajtót. Kivételt képez a soros vonal, amelynek működtető programja a rendszer ROM kiegészítésben található. Az előbb felsorolt csatolókártyák közül a floppy disk csatoló tartalmaz ROM-ot, így a floppy csatoló behelyezése automatikusan átrendeli a kazettás ki-bemenetet a floppy diskre (1. 6. Rendszer-inicializálás fejezet).

Minden csatolókártya számára le van foglalva egy I/O címtartomány (4. Függelék). A már korábban megismert I/O memória lapozáson és a megszakításkezelő input/outputkon kívül még az 5AH-s inputról kell beszélni. Lehetőség van minden csatolókártyán két bites azonosító bekötésére, ezek az azonosítók olvashatók be az 5AH-s inputról (4. Függelék).

A kártyaazonosítók:

- 00: Soros vonal (RS232).
- 01: Eredetileg ez a videojáték modul (Video Game Board, VGB) számára volt lefoglalva. A VGB nem fog ROM-ot tartalmazni, ezért azonosítóbitjeit azonosítatlan kártya értékre (11) kell állítani.
- 10: Floppy disk csatoló.
- 11: Üres kártyahely vagy azonosítatlan kártya.

A rendszer-inicializálás során az inicializáló program megvizsgálja a kártyaazonosítókat. Ha azonosított kártyát talál, feltölti a 004H...00FFH címeken található csatolókártya táblázatot a kártyákat azonosító nevekkal, és meghívja a kártyák ROM-jában található inicializáló rutint. (Kivétel az RS232, amelynek inicializáló rutinja a rendszer ROM kiegészítésében található.) Ezért a megfelelő ROM tartalom nélküli azonosított kártyák behelyezése a rendszer működőképtelenségét eredményezi.

A kártyák azonosíthatók a ROM tartalommal is, a ROM tartalomnak meg kell felelnie a később leírt formátumnak.

A 0040H kezdőcímen található csatolókártya táblázat felépítése az 5. Függelékben látható.

A 0, 1, 2, 3-as csatlakozónak megfelelően 40H, 70H, DA0H és 000H címen kezdődik az adott kártyára vonatkozó táblázat:

- 0. byte: azonosítónév hossza
- 1-6. byte: azonosító (max. 6 karakter)
- 7. byte: egységszám

Az azonosító string után 40 byte-os puffer van lefoglalva az egyes kártyák számára. Csatolókárttyára vonatkozó funkcióhívás esetén az IX regiszter erre a pufferre mutat.

Az azonosító 7. byte-ján található egységszám normál esetben 0. Eltérő (1, 2, 3) értéket akkor vesz fel, ha több azonos típusú kártya (pl. soros vonal) van a TV-Computerbe csatlakoztatva.

Az eddig lefoglalt azonosítószavak:

- RS232 - soros vonal
- DISK - floppy disk csatoló

4.2 Funkcióhívások

A fizikai és logikai eszközök (logikai eszköz a képernyőszerkesztő - screen editor -, amely lehetővé teszi a programgépélést) funkciói egységesen az RST 30H utasítással hívhatók. Az RST 30H (0F7H, 247) utasítást követő byte-on kell elhelyezni a funkciókódot. Pl. a printerre 1 byte-ot kiküldő hívás a következő alakú:

```
RST 30H
DEFB 41H ; funkciókód
```

(A kiküldendő byte-ot előzetesen el kell helyezni a C regiszterben.)

A funkcióhívások osztályokba vannak sorolva. Ezeket az osztályokat a funkciókód B6, B5, B4 bitje jelöli ki:

0	(000)	:	video osztály
1	(001)	:	billentyűzet osztály
2	(010)	:	editor osztály
3	(011)	:	hang osztály
4	(100)	:	nyomtató osztály
5	(101)	:	kazetta osztály
6	(110)	:	csatolókérttya osztály
7	(111)	:	kernel osztály

Az osztályok a sorszámmal (0...7) vagy megnevezésükkel azonosíthatók. A megnevezés (pl. nyomtató osztály) azt jelenti, hogy normál esetben az adott osztályra kiadott I/O műveletek a nevezett eszközön kerülnek végrehajtásra. Tehát a 4-es osztályra küldött adat alapesetben a nyomtatóra kerül.

Az átvitel irányát a funkciókód B7 bitje határozza meg: B7=0 esetén output (kimeneti), B7=1 esetén input (bemeneti) műveletről van szó.

A B3, B2, B1, B0 bitek jelölik a ténylegesen végrehajtandó funkciót. Az első 3 funkció minden eszközre kötelezően azonos:

- 0 (0000) : Megszakítást kiszolgáló rutin. Ezt a rutint hívja meg az operációs rendszer a megszakítás kiszolgálása céljából. Ezt a funkciót nem tanácsos a felhasználónak hívnia.
- 1 (0001) : Karakter (byte) I/O. A B7 bit jelöli ki az átvitel irányát. A kiküldendő karaktert a "C" regiszterben kell elhelyezni, ill. a beolvasott karakter a C regiszterbe kerül.
- 2 (0010) : Blokk I/O. Itt is a B7 bit mutatja meg, hogy inputról szól vagy outputról van-e szó. Az átvitel a memória egy kijelölt részének átvitelét jelenti. A DE regiszterpár tartalmazza a kiviendő memóriarész, ill. input esetén az input adatokkal feltöltendő memóriarész kezdőcímét. Az átvendő blokk hosszát a BC regiszterpár tartalmazza. Bármekkora memóriatartomány átvitele lehetséges a 0, 1, 2 memórialapról (0000...BFFF), de HI_MEM értéke nem léphető túl. (A HI_MEM korlát nem minden eszközre áll fenn. Az 1.3 verziótól kezdve a HI_MEM átlépés ellenőrzése elmarad.) A blokk I/O a karakter I/O ismételt hívását jelenti.

A BASIC programbetöltő és -kimentő utasításai (LOAD, SAVE) a karakter- és blokk I/O-n kívül OPEN, CREATE (3-as funkció) és CLOSE (4-es funkció) hívásokat is kiadnak az operációs rendszer felé, ezért a LOAD és SAVE csak olyan eszközre adható ki, amely a fenti hívásokat tartalmazza (kazetta, soros vonal, floppy disk).

Az eszközmeghajtók a funkcióhívásokat ROM-ban tárolt ugrási táblák alapján hajtják végre.

A rendszer ROM-ban, ill. kiegészítésében található eszközök (video, billentyűzet, editor, hang, nyomtató, kazetta, soros vonal) ugrási táblája a következő:

```
DEPB  [az eszközhöz tartozó funkciók száma]
DEFW  [megszakításkiszolgáló rutin címe]
DEFW  [karakter I/O funkciók címe]
DEFW  [blokk I/O funkciók címe]
DEFW  ....
DEFW  ....
DEFW  ....
[initializáló rutin]
```

Ha a csatolókartya ROM-ot tartalmaz, akkor az ugrási tábla ebben a ROM-ban helyezkedik el. Az IOMEM a 3-as lapra (0C000H) lapozható. A ROM első byte-jain található a kártyaazonosító és az ugrótábla:

```
C000  DEFM  'MOPS'
C004  DEFB  [az azonosító név hossza]
C005  DEFM  [azonosítónév, max. 6 karakter]
C008  DEFW  [initializáló rutin címe]
C00D  DEFB  [funkciók száma]
C00E  DEFW  [megszakításkiszolgáló rutin címe]
C010  DEFW  [karakter I/O funkciók címe]
C012  DEFW  [blokk I/O funkciók címe]
C014  DEFW  ....
C016  DEFW  ....
C018  DEFW  ....
```

A funkcióhívások során a paraméterek átadása a BC és DE regisztereken keresztül történik, a végrehajtáskor észlelt hiba kódját az A regiszterben, a hívás eredményét a DE, BC regiszterekben kapja meg a hívó. A HL, IX, IY, AF' és a második regiszterkészlet tartalma megőrződik. Hibátlan lefutás esetén A=0. A DE és a BC regiszterek tartalma csak akkor értelmezhető, ha A=0 (nincs hiba). Az A-ban visszaadott hibakód DFFH (255), ha nemlétező funkciókóddal hívta meg az operációs rendszert az alkalmazói program.

Az operációs rendszer a funkciókódokat az A regiszterben adja át az eszközmeghajtónak, az előjel (SIGN) jelző megfelel B7-nek (input vagy output művelet).

A funkcióhívások kezdő és befejező programrésze a RAM-ban, a LAP0-án található (l. 5. Függelék).

4.2.1 PROGRAMOZÁSI PÉLDA

A funkcióhívások BASIC-ből történő kiadására, esetleg a funkciók kipróbálására javasolunk egy rövid gépi kódú rutint. A rutin számára LOMEM-mel foglalunk helyet, a VLOMEM átírásával biztosítjuk, hogy a gépi kódú program NEW és LOAD hatására se vesszen el.

A rutin EXT0-val hívható:

EXT0, funkciókód, DE, BC

A hívás utáni regiszterértékek a memóriából PEEK-kel olvashatók:

E regiszter	:	19F0H	(6640)
D regiszter	:	19F1H	(6641)
C regiszter	:	19F2H	(6642)
B regiszter	:	19F3H	(6643)
A regiszter	:	19F4H	(6644), itt található az esetleges hibakód.

A gépi kódú program:

1A00 70	LD	A,L	
1A01 32 05 1A	LD	(FC),A	; A funkciókód lerakása.
1A04 F7	RST	30H	
	FC:		
1A05	DEFS	1	; A funkciókód helye (L)
1A06 ED 53 F0 19	LD	(19F0H),DE	; DE → 19F0H

1A0A ED 43 F2 19	LD	(19F2H),BC ; BC → 19F2H
1A0E 32 F4 19	LD	(19F4H),A ; A → 19F4H
1A11 C9	RET	

A gépi kódot betöltő BASIC program:

```

10 LOMEM 6674:POKE 5920,18:POKE 5921,26 ! VLOMEM
20 FOR I=6656 TO 6673:READ A:POKE I,A:NEXT
30 POKE 33,0:POKE 34,26 ! EXT0
100 DATA 125,50,5,26,247,0,237,83,240,25
110 DATA 237,67,242,25,50,244,25,201

```

RUN után ez a program akár ki is törölhető. Ezután az EXT0-val bármelyik operációs rendszer funkció meghívható.

4.3 I/O hozzárendelések

A TV-Computer fizikai eszközeivel az adatforgalom az operációs rendszer karakter és blokk I/O funkcióinak segítségével bonyolódik le. Ezeknek a funkcióhívásoknak az átirányításával (a hívások másik eszközhöz történő hozzárendelésével) elérhető, hogy a műveletet más eszköz hajtsa végre. Az átirányításra két példát mutatunk be.

1. A felhasználó egy olyan BASIC programot fejleszt, amelyik a futás eredményét nyomtatóra fogja kiírni, ehhez az LPRINT utasítást használja. Ha a programfejlesztés során nem akar sok papírt elhasználni vagy éppen nincs nyomtatója, valamennyi LPRINT utasítást át kellene írni PRINT utasítássá, majd később visszaírni. Ehelyett elegendő a 4-es (nyomtató) osztályhoz az editor outputot rendelni, és az LPRINT már a képernyőre dolgozik. A későbbiekben látni fogjuk, hogy az átrendeléshez elegendő a következő BASIC utasítás:

POKE 2828,2

Visszarendelés:

POKE 2828,4

2. A TV-Computer tulajdonosnak soros interface-szel rendelkező nyomtatója van, ezért vásárolt egy soros vonalat. A

POKE 2828,6

utasítás eredményeként az LPRINT, LLIST, PRINT#4: utasítások a soros vonalra fognak dolgozni.

Az I/O hozzárendelés két táblázat segítségével történik, a táblázatok az operációs rendszer munkaterületén találhatók a 0B00H és 0B08H címen (l. 6. Függelék).

0800H : IN_TABLE ; input hozzárendelések
0808H : OUT_TABLE ; output hozzárendelések

A 7-es funkcióosztály (kernel) nem irányítható át, ezért a táblázatok a 0...6-os osztályok átrendelését teszik lehetővé.

A táblázatok felépítése a következő:

- a kezdőcímmre vonatkoztatott relatív cím jelöli ki a funkcióosztályt,
- az adott cím tartalma jelenti az osztályhoz rendelt eszközt.

Alap-hozzárendelésben az osztálynak megfelelő eszköz van kijelölve, azaz

- a 2-es (editor) osztályhoz a 2-es eszköz (editor),
- az 5-ös (kazetta) osztályhoz az 5-ös eszköz (kazetta) stb.

A nem értelmezett hozzárendeléseknél (pl. nyomtató input, billentyűzet output) a táblázat OFFH-t tartalmaz. (A nem értelmezett megjegyzés a karakteres és blokk input/outputra vonatkozik, más funkciók ezekben az esetekben is értelmezettek.)

A táblázat 8. (7-es relatív című) eleme azt jelöli ki, hogy a 6-os (csatolókártya) osztály melyik konkrét kártyára vonatkozik. Értéke lehet 0, 1, 2, 3 a négy interface csatlakozónak megfelelően, ill. OFFH, ha nincs csatolókártya vagy nincs kijelölve. (Inicializáláskor a TV-Computer automatikusan kijelöli a soros vonalat a 6-os funkcióhoz, ha van soros vonali interface.)

A hozzárendelésnek 3 típusa van:

1. I/O hozzárendelés, pl. a 4-es osztályhoz (nyomtató) hozzárendeljük a 2-es eszközt (editor). Ez a hozzárendelés az 1-es és 2-es funkcióra (karakter- és blokk I/O) vonatkozik.

Ha az átvitel során hiba lép fel, automatikusan visszaáll az alap-hozzárendelés.

2. I/O hozzárendelés csatolókárttyához

Ez a 6-os osztály hívását jelenti, a 6-os osztályhoz a táblázat utolsó byte-ja segítségével rendeljük hozzá a kívánt csatolókárttyát. (A 6-os osztály hívása egy másik osztály átrendeléséből is adódhat.)

3. Közvetlen hozzárendelés az interface csatlakozóhoz.

A hozzárendelési táblába a csatlakozószámot kell beírni úgy, hogy 87=1 legyen, tehát a 0, 1, 2, 3-as csatlakozónak 80H, 81H, 82H, 83H felel meg (decimálisan 128, 129, 130, 131). Ez a hozzárendelés az összes funkciót átrendeli a csatlakozóhoz, hiba esetén sem áll vissza az alap-hozzárendelés.

(Mint már említettük a csatlakozók számozása balról jobbra: 3, 2, 1, 0.)

Nem lehet a hozzárendeléseket láncolni, nem fog működni a következő hozzárendelés:

4. (nyomtató) osztály, 2828: 05 (kazetta)

5. (kazetta) osztály, 2829: 131 (a 3-as csatlakozóban lévő floppy csatoló)

Ebben az esetben a nyomtató osztályhoz ténylegesen a kazetta fog hozzárendelődni.

Működőképes floppy hozzárendelés a printer osztályhoz:

4. (nyomtató) osztály, 2828: 131

vagy

4. (nyomtató) osztály, 2828: 06 (csatolókártya)
Csatolókártya-kijelölés, 2831: 03

Ha a hozzárendelési táblázat valamelyik osztályához tartozó címén OFFH van, az 1-es 2-es funkciók hibát jeleznek és a 3...15-ös funkciók az osztályhoz tartozó eszközön lesznek végrehajtva.

A funkcióhívások során a kernel a hozzárendelést a következő lépésekben hajtja végre:

1. A felhasználói memórialapozás és a megőrzött regiszterek mentése, az U0, U1, U2, SYS lapozás beállítása és a funkciókód beolvasása.
2. A B7 bitnek (input/output) megfelelően a kernel megcímzi a megfelelő hozzárendelési táblát (IN_TABLE, OUT_TABLE).
3. A funkcióosztály és a funkciószám kiolvasása a funkciókódból.
4. Ha 7-es osztály (kernel hívás) van, akkor nincs átirányítás, a program a rendszer RDM ugrótábla kezelőjére ugrik. (10)

A zárójelben levő 10-es azt jelenti, hogy a program a 10. pontnál folytatódik.

5. A 0...6-os osztály esetén a hozzárendelési táblázat megfelelő elemének kiolvasása. Ha értéke OFFH, és 0, 1, 2-es a funkciószám, akkor nem megengedett a hozzárendelés, visszatérés hibakóddal. (12)

A 3...15-ös funkciókód és OFFH esetén nincs átrendelés. (10)

6. Ha az eszközsám értéke (a B7-es bitet figyelmen kívül hagyva) nagyobb 6-nál, akkor visszatérés hibakóddal. (12)
7. Ha a kiolvasott eszközszámban B7=1, akkor közvetlen hozzárendelés van a csatlakozóhoz, a rendszer ROM ugrótábla kezelője megállapítja a kijelölt csatlakozóba elhelyezett kártya típusát.

Ha a típus RS232, akkor a funkciószám szerinti ugrótábla a rendszer ROM kiegészítésben van, egyébként a kijelölt csatlókártyán. (10)
8. Ha a funkciószám nagyobb 2-nél, nincs átrendelés. (10)
9. A funkciókód osztályszámának kicserélése a táblázatból a felvett értékkel.
10. A kiválasztott ugrótábla szerint a funkciószám valódiságának ellenőrzése. Nem létező funkciószám esetén visszatérés hibakóddal. (12)
11. Elugrás a funkciószám szerint és a tényleges funkció végrehajtása.
12. A jelzők beállítása a hibakód szerint.
13. Hiba esetén az alap-hozzárendelés visszaállítása (kivéve a közvetlen csatlakozóhoz rendelést).
14. Visszatérés a hívóhoz (felhasználói memórialapozás visszaállítása, megőrzött regiszterek visszaállítása).

Hibakezelés:

Ha az I/O hozzárendelés vagy a funkciók végrehajtása során a program hibát észlel, az alap-hozzárendelés áll vissza.

Az alapérték visszaállítása nem történik meg közvetlen csatlakozóhoz rendelés esetén.

A rendszer nem vizsgálja a hozzárendelések során a rekurzív funkcióhívásokat. Kivétel ez alól az editor input hozzárendelése a billentyűzet osztályhoz (a billentyűzet meghívja az editort, amely billentyűzet hívást tartalmaz stb.). Ebben az esetben visszaáll az alap-hozzárendelés.

4.3.1 PROGRAMOZÁSI PÉLDA

A hozzárendelések alkalmazását két BASIC program összefésülésén mutatjuk be.

Előfordulhat, hogy egy BASIC programba be kellene szűrni egy másik BASIC programrészt. Mivel MERGE utasítás nincs, ez közvetlenül nem oldható meg.

Az összefésülésre mégis van lehetőség:

ASCII formában írt listát készítünk kazettára az egyik (célszerűen a rövidebb) programból úgy, hogy a programlistát az 5-ös osztályra (kazetta) küldjük:

```
OPEN OUTPUT "NÉV": LIST *5:: CLOSE OUTPUT
```

(Ez nem jár a hozzárendelési tábla átírásával, mert a LIST ilyenkor közvetlenül a kazettakezelőt hívja.)

Mivel a felírás 256 byte-os blokkokban történik, ez a kiírás hosszabb, mint a normál SAVE.

Töltsük be ezután a másik programot a LOAD paranccsal, majd az editor inputtal olvassuk be az ASCII formátumú szalagot:

```
OPEN "NÉV": POKE 2818,5
```

Az editor - mintha most gépelnénk - beolvassa a programlistát úgy, hogy az azonos sorszámu sorokkal felülírja a régi sorokat, az eltérő sorszámu sorokat közbeszúrja.

Nagyobb programok összefésülésénél a közbeszűrés időigényes, ezért ilyenkor a blokkok után rövid időre célszerű leállítani a pillanat-stop (pause) kapcsolóval a magnót (nincs gond, ha olyan magnónk van, amely tudja fogadni a motorvezérlő jelet).

Amikor a teljes program beolvasása befejeződik, a kazettakezelő file-vég hibát fog jelezni. A hibajelzésre visszaáll az alap-hozzárendelés (a képernyőn két ok látszik). Ezután ki kell még adni egy CLOSE parancsot.

Floppy disk esetén a kiírás ugyanúgy történhet, a beolvasás viszont eltérő:

```
OPEN "NÉV": POKE 2818,128+ [floppy csatoló kártyahelye]
```

Mivel ez közvetlen hozzárendelés, a filevég hibára nem áll vissza az alap-hozzárendelés. Ahhoz, hogy a billentyűzetről folytatni lehessen a munkát, a RESET gombot kell egyszer megnyomni.

Persze lehetséges nem közvetlen hozzárendelés is:

```
POKE 2823, [floppy csatoló kártyahelye]  
OPEN "NÉV": POKE 2818,6
```

A beolvasás után ne felejtsük el a CLOSE-t kiadni.

5. RENDSZERVÁLTOZÓK

A TV-Computer rendszerváltozói a 0-ás memórialapra helyezkednek el.

Az egyes rendszerváltozók értelmezése a vonatkozó hardver- és szoftver-eszközök tárgyalásánál található meg, de ettől függetlenül a 23. Függelékben egy összefoglaló táblázatban is felsoroljuk az összes változót. Most csak globális áttekintést adunk a LAP0 felhasználásáról. A felsorolásban megadott értékek: cím hexadecimálisan, utána zárójelben a decimális cím és a decimális hossz.

1. 0000H (0,48) : A BASIC által használt terület, ahol pl. a hibakezelő rutin és az EXT utasítás címtáblája található. Kivétel a 3-as cím, ahol a P_SAVE változó van.

Vigyázat! A TV-Computerhez vásárolható programmodul debugger az RST 28H utasítást használja a töréspontok kezelésére, ezért a debugger alkalmazása esetén az EXT3, EXT4 nem használható. BASIC hiba miatt a SOUND utasítás elrontja az EXT5, EXT6 utasítások címeit.

2. 0030H (48,8) : Az operációs rendszer hívások (funkcióhívások) belépési pontja.
3. 0038H (56,8) : A megszakítások belépési pontja.
4. 0040H (64,192) : A csatolókarttyák azonosító táblázata és munkaterülete.
5. 0100H (256,1600) : A képernyőn tárolt szöveg ASCII kódú képe az editor műveletekhez. Hossza 25 sor + 64 karakter = 1600, a 25. sor nincs használva.

6. 0740H (1856,960) : A 128...223 kódú definiálható karakterek pontmátrixa (96 karakter + 10 byte).
7. 0B00H (2816,114) : Az operációs rendszer változói.
8. 0B72H (2930,15) : Video munkaterület.
9. 0B81H (2945,100) : I/O munkaterület.
10. 0BE5H (3045,10) : Billentyűzet munkaterület.
11. 0BEFH (3055,1) : Hang munkaterület.
12. 0BFDH (3056,600) : Kazetta munkaterület.
13. 0E48H (3656,80) : Editor munkaterület.
14. 0E98H (3736,20) : Programazonosító név (csak 1.3 verziótól).
15. 0EACH (3756,2048) : Rendszer verem.
16. 16ACH (5804,835) : BASIC munkaterület.
17. 19EFH (6639) : BASIC programok kezdete.

Az 1.3 verziótól kezdődően a 0E98H címen egy 20 byte-os terület van lefoglalva a futó program azonosítására. Az azonosító 0-val van lezárva. Reset hatására (meleg reset hatására is) az azonosító törlődik, majd a BASIC a "TV COMPUTER BASIC" azonosítót írja be.

Megjegyezzük, hogy a rendszerváltozó területen található lyukakat (pl. az ASCII képernyő 25. sorát) csak korlátozott mértékben célszerű gépi kódú rutinok tárolására használni, mivel ezeket a szabad címeket a gép későbbi bővítései felhasználhatják.

6. RENDSZER-INICIALIZÁLÁS

A TV-Computer mikroprocesszorát és hardver regisztereit alaphelyzetbe állító ún. reset impulzus kétféleképpen keletkezhet:

- Bekapcsoláskor a +5 V-os tápfeszültség felfutásából, késleltetéssel;
- A reset nyomógomb megnyomásából, szinkronizálva a processzor /M1 jeléhez. (A szinkronizálással a memóriatartalom esetleges megsérülését lehet elkerülni. Előfordulhat, hogy erős külső, pl. a hálózaton érkező zavar a CPU-t olyan állapotba viszi, amelyben nincs utasításle hívás, tehát nincs /M1 jel. Ebben az esetben a reset nyomógomb hatástalan, csak a ki- bekapcsolás segít.)

A reset impulzus háromféle programállapotban keletkezhet:

- Bekapcsoláskor: ebben az esetben a RAM tartalma határozatlan, a különböző tesztelések miatt hideg reset fog végrehajtódni.
- Programfutas közben: ilyenkor a rendszerváltozók értékétől függően hajtódik végre hideg vagy meleg reset.
- Inicializálás alatt: ez a reset gomb ismételt megnyomását jelenti, tehát a meleg resetből hideg reset lesz.

Ezek figyelembevételével az inicializálási folyamat a következő:

1. A reset impulzus letiltja a CPU megszakításokat, a programszámlálót 0-ra állítja. A TV-Computer meghatározott regisztereit alaphelyzetbe állítja, pl. a PORT02 lapozási regiszter 00 értéket vesz fel, ami SYS, UI, VID, CART lapozást jelent. Emiatt a programfutas a 0 címről indul, ahol a rendszer ROM található.

2. A processzor inicializálja a képernyővezérlőt.
3. A COLD_FLAG (0B22H,2850) és a WARM_FLAG (0B21H,2849) rendszerváltozók vizsgálata:

Ha a COLD_FLAG értéke OFFH, a "meleg" RESET le van tiltva, a program a WARM_FLAG-be 0-át ír és "hideg" RESET hajtódik végre (7).

Ha a WARM_FLAG értéke nem 0, akkor meleg reset van végrehajtás alatt (a reset nyomógombot rövid időn belül másodszor is megnyomták), tehát hideg resetet kell végrehajtani. A WARM_FLAG 0 értéket vesz fel. A program a 7. pontnál folytatódik.

4. Az inicializáló program megvizsgálja, hogy a korábbi inicializáláskor a ROM-ból RAM-ba töltött program megsérült-e. Ha a RAM és ROM tartalom nem azonos, akkor hideg reset lesz és a program a 7. pontnál folytatódik. (Előfordulhat, hogy a vizsgált memóriatartalom ép, de a RAM tartalom valahol másutt megsérült. Ilyenkor a meleg reset végrehajtása közben a rendszer beragadhat, azaz csak a kettős megnyomással kiváltott hideg reset segít.)
5. Meleg reset hajtódik végre, a program a WARM_FLAG-et OFFH-ra írja át. Az eszközmeghajtó inicializáló rutinok a WARM_FLAG-et tesztelhetik, ennek megfelelően hajtják végre az eszközinitializálást.
6. Az inicializáló program fél másodpercig vár, ezzel lehetővé teszi a reset nyomógomb kettős megnyomását. A várakozásra azért van szükség, mert a meleg reset nem hajt végre memóriatesztet és memóriatörlést. A gyorsabb lefutás miatt nem lehetne a reset gombot másodszor megnyomni az inicializálás alatt.

A HI_MEM változó értéke a meleg reset alatt megőrződik.

A meleg reset a memóriatesztet átugorja, ezért folytatás a 9. pontnál.

7. Az inicializáló program a rendszer ROM-ot a harmas lapra lapozza és teszteli a 0-ás lapon az U0 memóriát: 55H írása/olvasása, 0AAH írása/olvasása, olvasás előtt memóriadekrementálás (csökkentés).

A jó memóriarekeszeket a tesztelés után törli a program. Ha az U0 bármely címe hibás, a képkeret (BORDER) villogni fog és a rendszer-inicializálás nem folytatódik.

Ezután a video RAM-ot teszteli a program. (A TVC 64 k+ típusnál csak az első 16 kbyte-ot.) Hiba esetén néhány másodpercre vörös lesz a képkeret és az inicializálás leáll.

8. A program teszteli az U1, U2, U3 RAM memóriát. Az U3 tesz eredménye beíródik P3RAM változóba (0: jó, FF: hibás az U3). Az U1, U2 tesz eredményétől függően beállítódik a HI_MEM rendszerváltozó. Hibátlan 32 k-s gép esetén értéke 7FFFH, 64 k-s gép esetén 0BFFFH.

9. A rendszer-kiegészítő ROM a 3-as lapra lapozódik és megtörténik a 0-ás RAM lap feltöltése:

- A 32 újradefiniálható karakter pontmátrixa átmásolódik a ROM-ból a RAM-ba, a fennmaradó 64 definiálható karakter betűközzel töltődik fel.
- A RESTART utasításokhoz tartozó programrészek és a további, RAM-ban tárolt programrészek átmásolódnak.
- A rendszerváltozókat, az I/O hozzárendelési táblázatokat a program feltölti a kezdőértékekkel.

10. A PORT5AH-t beolvasva az inicializáló program megvizsgálja a hardver-azonosítók alapján, hogy az egyes interface kártyahelyek milyen bővítő kártyákat tartalmaznak.

Ha a hardverazonosító üres kártyahelyet, ill. az ezzel egyenértékű ismeretlen kártyát jelez, az inicializáló program teszteli az inter-

face kártyákon található IOMEM-et ("MOPS" azonosító, kártya-azonosító). A tesztek eredményét elhelyezi az operációs rendszer munkaterületén.

11. A rendszer meghívja az összes eszköz inicializáló rutinját, beleértve az interface kártyák inicializáló rutinait is. Az inicializáló rutinok a WARM_FLAG figyelembevételével futnak le.

Ha floppy disk interface-t tartalmaz a rendszer, a kazetta hívások átirányítódnak a floppy csatolóhoz (közvetlen hozzárendelés a csatlakozóhelyhez, azaz valamennyi kazetta funkció a floppy diskre fog vonatkozni).

12. A 0-as lapra a rendszer ROM, a 3-as lapra a programmodul (CART) lapozódik. Az inicializáló program megvizsgálja, hogy a programmodul elején megtalálható-e a MOPS azonosító. Ha igen, a vezérlés átadódik a programmodulnak.

Ha nincs programmodul ROM, ill. nincs MOPS azonosító, a vezérlést a BASIC kapja meg.

A programmodulban tárolt program ill. a BASIC a WARM_FLAG-ból megállapíthatja, hogy induláskor milyen inicializálást kell végrehajtania. (Pl. meleg resetnél a BASIC nem hajt végre automatikus NEW-t.)

A WARM_FLAG törlését és a megszakítás engedélyezését a BASIC végzi, ill. a programmodulban tárolt programnak kell végrehajtania.

7. KERNEL HÍVÁSOK

A felhasználó által hívható kernel funkciók input/output hozzárendelési és memóriahelyfoglalási műveleteket jelentenek. A kernel funkciók csak a 7-es osztály hívásával érhetők el, átrendelésre nincs lehetőség, ezért itt az 1-es és 2-es funkciók sem a szokásos karakteres, ill. blokkátvitelt jelentik. Nincs megkülönböztetve az átvitel iránya sem, a funkciókódban a 87 bit értéke közömbös.

Megjegyezzük, hogy a kernel hívások kiadása helyett általában elegendő az operációs rendszer változóinak átírása. Programokból mégis célszerűbb a funkcióhívásokat használni a változók átírása helyett, mert

- a funkcióhívások a megfelelő ellenőrzések révén csökkentik a hibás kijelölések, hibás tárfoglalások valószínűségét;
- hozzárendeléseknél nem szükséges a csatlakozóhelyet megadni, a csatoló nevére lehet hivatkozni (tehát mindegy, hogy hova van dugva a csatoló-kártya, a program működni fog).

A funkcióhívások ismertetésénél megadjuk a funkció nevét, kódját, a hívási paramétereket, a hívás után visszaadott értékeket, a lehetséges hibakódokat. A kernel funkcióknál adjuk meg az általánosan előforduló hibakódokat (hibás blokk I/O paraméterek, STOP). Ezeket a konkrét eszköz-meghajtóknál nem mindig tüntetjük fel.

7.1 HI_MEM_SET

Memória foglalás a legmagasabb használható RAM-cím csökkentésével.

Kód: 71H vagy 0F1H (113 vagy 241)

Input: DE = a lefoglalni kívánt memóriaterület mérete

Output: DE = a lefoglalt puffer kezdőcíme

Hibakód: 0FBH (251): nincs elég memória

A HI_MEM_SET rutin csökkenti a HI_MEM rendszerváltozó értékét a DE-ben megadott értékkel. DE-ben visszaadja a lefoglalt memóriapuffer kezdőcímét. Hibát jelez a rutin akkor, ha a lefoglalás után nem marad legalább 2 kbyte memória a rendszer verem után. (Tehát a HI_MEM_SET-tel nem csökkenthető a HI_MEM értéke 1EACH, 7852 alá.) A DE=0 nem változtatja a HI_MEM értékét.

Itt jegyezzük meg, hogy a HI_MEM nem BASIC, hanem operációs rendszer változó, értékét a BASIC csak bizonyos parancsoknál (pl. NEW, RUN) veszi figyelembe. Ezért a BASIC programból kiadott HI_MEM_SET hívást vagy a HI_MEM átírását célszerű egy RUN sorral folytatni.

Példa:

A 7000H (32000) címtől akarjuk lefoglalni a memóriát, azaz HI_MEM-et 7CFFH-ra kell állítani.

POKE 2841,255 : POKE 2842,124

A 2841-es címet általában nem kell átírni, hiszen jó memóriánál 255-öt tartalmaz (7FFFH, ill. 8FFFH az alapérték).

A fenti sorok után HI_MEM már felvette az új értéket, erről meg is győződhetünk:

```
PRINT PEEK(2841)+256*PEEK(2842)
31999
```

A FREE változó azonban még a régi értéket mutatja, NEW vagy RUN hatására fog a csökkentett érték beállítódni. Ezért a BASIC programot a következőképpen célszerű kezdeni:

```
1 POKE 2841,255 : POKE 2842,124
2 RUN 10
10 .....
```

A 10-es sortól a BASIC program a 7D00H-val kezdődő memóriarészt már szabadon használhatja. A memória feltölthető gépi kódú rutinnal, vagy más tetszőleges tartalommal.

7.2 EXP_DIRECT

Csatolókártya kijelölése a 6-os osztályhoz.

Kód: 72H vagy 0F2H (114 vagy 242)

Input: B = 0FFH : input kártya kijelölés
<> 0FFH : output kártya kijelölés

C = egységszám (több azonos típusú kártya megkülönböztetése,
egy kártya esetén 0)

DE = a kártyaazonosító memóriacíme

Hibakód: 0FDH (253) : nemlétező kártya

Bármelyik szabad memóriaterületen elhelyezhető a kijelölendő kártya azonosítója hossz+név formában, DE-nek az azonosítóra kell mutatnia. Ha a megadott azonosítójú kártya nem létezik, a rutin hibakóddal tér vissza.

Az EXP_DIRECT hívása helyett átírható az IN_TABLE, ill. az OUT_TABLE 7. eleme is. Természetesen ebben az esetben tudni kell, hogy a kijelölendő kártya melyik csatlakozóba van dugva (0...3).

7.3 I/O_DIRECT

I/O hozzárendelés

Kód: 73H vagy 0F3H (115 vagy 243)

Input: D = 0FFH : input hozzárendelés
 <>0FFH : output hozzárendelés

 B = a funkcióosztály száma

 C = az eszköz száma

Hibakód: 0FEH (254) : hozzárendelési hiba

A D regiszter értéke szerint a rutin átírja az IN_TABLE vagy az OUT_TABLE megfelelő elemét, a B regiszterben kijelölt osztályhoz hozzárendelve a C regiszterben megadott sorszámú eszközt.

Ha az osztály vagy az eszközsorszám nagyobb 6-nál, vagy a megfelelő táblaelem értéke 0FFH, a rutin hibával tér vissza.

Példa:

Ha az I/O_DIRECT hívása előtt D=00, B=04, C=02 értékre állítjuk a regisztereket, a rutin a 4-es output (nyomtató) osztályhoz hozzárendeli a 2-es eszközt (editor):

```
.  
.   
.   
LD     D,0  
LD     B,4  
LD     C,2  
RST    30H
```

DEFB 73H

.
.
.

Ez a hívás helyettesíthető a következő BASIC utasítással:

POKE 2828,4

7.4 SLOT_NUM

Csatolókártya pozíciójának meghatározása.

Kód: 74H vagy 0F4H (116 vagy 244)

Input: DE = a kártyazonosító memóriacíme
C = egységszám

Output: C = csatlakozószám

Hibakód: 0FDH (253) : nemlétező kártya

A kártyaazonosítót és az egységszámot ugyanúgy kell megadni, mint az EXP_DIRECT-nél.

Ez a rutin a C regiszterben visszaadja a keresett kártya csatlakozószámát.

8. KÉPMEGJELENÍTÉS

A képmegjelenítésre szolgáló hardver és az azt működtető szoftver, a video eszközmeghajtó a TV-Computer legfontosabb és egyben legbonyolultabb része. A video eszközmeghajtóra épül a BASIC kényelmes, jól kezelhető, egyszerű, de mégis hatékony grafikája. A BASIC-ben használható grafikus funkciók gépi kódból is hívhatók, ezért gépi kódban sem nehéz látványos grafikát tartalmazó programokat írni. A video eszközmeghajtó megismerése a BASIC programozás lehetőségeit is kibővíti.

Szokásunkhoz híven a video eszközmeghajtó ismertetése előtt áttekintjük a képmegjelenítő hardvert.

8.1 Képmegjelenítő hardver

8.1.1 VIDEO MEMÓRIA ÉS KÉPERNYŐVEZÉRLŐ

A képmegjelenítő hardver programozási szempontból a következő fontosabb egységeket tartalmazza:

- 16 k vagy 4*16 k (TV-Computer 64 k+) videomemória
- 6845 típusú képernyővezérlő
- képpont léptető shift regiszter
- paletta regiszterek
- border (keret szín) regiszter

Megjegyzés: A képernyővezérlő a számítástechnikai szakirodalomban a következő szinonímákkal is szerepel: CRT controller (ahol a CRT = Cathode RAY Tube, magyarul katódsugárcső, képernyő), CRT vezérlő, display vezérlő.

A TV-Computer az ábrázolt grafikus vagy szöveges információt a tv technikából ismert módon jeleníti meg: a képernyőn soronként fut végig az elektronsugár, a sugár intenzitásának vezérlésével (színes televízió esetén a három elektronsugár intenzitásának vezérlésével) állíthatók elő a különböző színű, különböző fényességű képpontok.

Az elektronsugár egyszeri végigfutásának nyomát nevezzük a továbbiakban TV sornak vagy rastersornak. Egy-egy betű vagy írásjel felrajzolásához több TV sorra van szükség.

A sorközöket is beszámítva, az egy betű- (karakter-) sorhoz tartozó tv sorokat együtt karaktersornak nevezzük. A karakter a betűk, számok, írásjelek, szimbólumok gyűjtőneve. A TV-Computernél 24 karaktersor fér el a képernyőn, egy karaktersor 10 tv sorból áll, tehát a képernyő hasznos felülete 240 sort tartalmaz.

A képmegjelenítést tekintve szokás grafikus és alfanumerikus megjelenítésről beszélni. (Alfanumerikus: betűk és számok megjelenítésére szolgáló eszköz, berendezés.) Alfa numerikus megjelenítés esetén a képmemóriában a megjelenített karakterek kódja van tárolva, pl. egy A betű a memóriában 1 byte-ot foglal el, a byte tartalma 41H (65). Az A betűhöz tartozó képpontokat egy külön memória, az ún. karaktergenerátor tartalmazza. A karaktergenerátor címbemeneteire rá kell adni a megjelenített karakter kódját és azt, hogy hányadik tv sornál tart a megjelenítés. Ezek hatására a karaktergenerátor kimenetén megjelennek a kiválasztott karakter megfelelő tv sorához tartozó képpontok.

Grafikus megjelenítés esetén a képmemóriában a képernyőn megjelenő képpontokat tároljuk. A TV-Computerben grafikus megjelenítés van, egy A betű 10, 20 vagy 40 byte-ot foglal el a képmemóriában a grafikus üzemmódtól függően.

A grafikus megjelenítőkben a szoftver végzi a karaktergenerálást: a memóriában bárhol eltárolt karaktergenerátor alapján a szoftver tölti fel a képmemóriát az "A" betű képpontjaival.

Megjegyezzük, hogy az előbbi rövid fejtegetés korántsem nyújt teljes áttekintést a megjelenítő rendszerekről, léteznek egészen más elven működő be-
rendezések is.

A TV-Computerben a megjelenítést egy 6845-ös típusú képernyővezérlő irányítja, ez az áramkör alapvetően alfanumerikus megjelenítésre szolgál.

A 6845-ös fontosabb funkciói:

- Az egy tv soron belüli karaktermegjelenítési időt meghatározó órajelből (a TV-Computernél ez 3.125 MHz, azaz 640 ns) előállítja a tv eltérítő áramköreit vezérlő vízszintes és függőleges szinkronjelet (HS, VS), valamint a képernyőn a hasznos képet kijelölő DE (más néven DISPTMG) jelet.
- Kiadja a karakterek kódjait tartalmazó memória címét (MA0...MA13) lehetővé téve az éppen megjelenített karakter kiolvasását.
- Kiadja a karaktergenerátor vezérléséhez a karaktersoron belüli tv sor sorszámát (RA0...RA4).
- Előállít egy cursormegjelenítő jelet, amelyet a TV-Computer csak programmegszakításra használ.

Egy alfanumerikus megjelenítő memóriájában a karakterek kódjai folyamatosan vannak tárolva. Az alfanumerikus megjelenítési rendszert végig-gondolva adódik, hogy az egy karaktersorhoz tartozó karakterek kódjait annyiszor kell ismételtlen kiolvasni, ahány tv sorból áll a karaktersor.

Egy karaktersor kiolvasása alatt tv soronként nő a képernyővezérlő által kiadott tv sor számláló érték (RA0...RA4).

A 6845-ös képernyővezérlőt grafikus megjelenítésre használva a kézenfekvő megoldás az lenne, hogy a "karaktersor" 1 tv sorból álljon. Ebben az esetben ugyanis a grafikus megjelenítésnek megfelelően a képmemória kiolvasása

folyamatos lenne, egy byte csak egyszer kerülne kiolvasásra. Sajnos ezt a 6845-ös korlátai nem teszik lehetővé, ugyanis nincs lehetőség 240 hasznos "karaktersor" programozására. (A "karaktersort" most azért tesszük idézőjelbe, mert nem a képernyőn látható tényleges, a TV-Computeren 10 rastersorból álló karaktersorról van szó. Itt a "karaktersor" azt a rastersor mennyiséget jelenti, amelyen belül ugyanaz a címtartomány ismételtelen van kiadva.)

A TV-Computerben alkalmazott megoldás a következő:

- A 6845-ös képernyővezérlő 4 tv sorból álló logikai "karakssorral" dolgozik;
- A videomemória címzésére az MA jeleken kívül az RA rastersor-kijelölő vezetékeket is felhasználjuk úgy, hogy a képmemória címvezetékei növekvő sorrendben:

MA0...MA5 : kijelölik az egy tv soron belüli logikai "karaktereket"

RA0...RA1 : kijelölik az egy "karaktersorhoz" tartozó tv sorokat

MA6...MA11: kijelölnek 60 logikai "karaktersort"

Mint ismeretes, a TV-Computer 3 grafikus üzemmódban dolgozhat. Kétszínű üzemmódban 1 bit, négyszínű módban 2 bit, tizenhatszínű módban 4 bit tartozik egy képponthoz a képmemóriában. Ebből az is következik, hogy egy tv sorhoz mindig 64 byte tartozik:

$512 \text{ pont} * 1 \text{ bit} = 64 \text{ byte}$

$256 \text{ pont} * 2 \text{ bit} = 64 \text{ byte}$

$128 \text{ pont} * 4 \text{ bit} = 64 \text{ byte}$

A képmemóriában tárolt 1 byte a képernyőn 8, 4 vagy 2 képpontot jelent, ezt az 1 byte-ot a képernyővezérlő 1 "karakternek" érzi, ezért beszélünk logikai karakterről. A 8. Függelékben látható, hogy a LAP2-re lapozva a videomemóriát, a logikai "karakterek" és "karaktersorok" milyen címeken

érhetők el. Látható, hogy az utolsó használt cím 0B8FFH, tehát a 0B000H-tól a 0B8FFH-ig tartó 1 kbyte nincs kihasználva.

8.1.2 A MEGJELENÍTÉS FOLYAMATA

A megjelenítés folyamata a következő:

- A képernyővezérlő végigfut az MA0...MA5 címeken, kiolvasva ezáltal 64 kbyte-ot ("karaktert").
- RA0, RA1 eggyel nő, így MA0...MA5 újbóli végigfutása már a következő 64 byte olvasását jelenti.
- Ez ismétlődik amíg RA0, RA1 sorra föl nem veszi a 0, 1, 2, 3 értéket.
- Ezután MA6...MA11 értéke eggyel nő, kezdődik a következő karaktersor.
- Ez ismétlődik, amíg MA6...MA11 végig nem fut mind a 60 "karaktersoron".

A memóriából kiolvasott byte-ok egy shift regiszterbe íródnak, majd grafikus üzemmódtól függően:

- Kétszínű üzemmódban a shift regiszterből 12.5 MHz-es órával (80 ns) lépnek ki a bitek, az először kilépő bit a B7, tehát az 1 byte-hoz tartozó 8 képpont közül a balszélsőt a B7, a jobbszélsőt a B0 bit értéke határozza meg. (Az elektronsugár balról jobbra halad.)

A kilépő bitek értéküktől függően a 0-ás, ill. 1-es palettaregisztert választják ki. Tehát a 0 értékű bit esetén (háttér) a 0-ás palettaregiszter, 1 értékű bit esetén az 1-es palettaregiszter kapcsolódik egy pillanatra a videojelet előállító szín kódolóra. Így a képpont a megfelelő palettaregiszterben tárolt színű lesz.

- Négyszínű üzemmódban a shift regiszter 8. bitjén kívül a 4. bit is a palettaregiszterek címzésére van felhasználva. A léptető óra 6.25 MHz (160 ns), az egy byte-hoz tartozó 640 ns alatt csak 4 léptetés (4 képpont) van. Ebből következik, hogy B7 és B3 meghatározza a 4 pont közül a balszélső színét, B4 és B0 a jobbszélső színét (l. 8. Függelék).

Az egy képponthoz tartozó 2-2 bit értékétől (0...3) függően van kiválasztva a képpont idejére a 4 palettaregiszter egyike.

- Tizenhatszínű üzemmódban a léptetőfrekvencia 3.125 MHz (320 ns), a shift regiszter 8., 6., 4. és 2. bitje van felhasználva. Egy byte-hoz csak 2 léptetés tartozik. Az egy képponthoz tartozó bitek nem a palettaregisztert címzik, hanem közvetlenül a színkódoló áramkörre jutnak, tehát magát a színt határozzák meg (l. a 8. Függelék). I jelenti a fényes/sötét szín kiválasztást (intenzitás), a G, R és B magát a színt jelenti (G: zöld, R: vörös, B: kék).

Az előbbieken említett palettaregiszterek egy 4*4 bites regisztermezőt jelentenek. A 4 regiszter közül íráskor a CPU választ, olvasáskor a shift regiszterből kilépő bitek (a 8. Függelékben C0, C1) választanak.

A CPU a regisztereket a következő I/O címeken éri el:

PAL0: 60H (96)

PAL1: 61H (97)

PAL2: 62H (98)

PAL3: 63H (99)

A regiszterekbe a CPU B6, B4, B2, B0-ás adatbitjei íródnak:

B6: I

B4: G

B2: R

B0: B

A képheret (border) színét a borderregiszter határozza meg, írási címe 0.

A CPU B7, B5, B3, B1-es adatbitje íródik a borderregiszterbe:

B7: I

B5: G

B3: R

B1: B

Megjegyezzük, hogy a rendszer (cursor) megszakításprogram mindig beírja a BORDER (0B4FH, 2895) rendszerváltozó értékét a borderregiszterbe. Ezért a PORT0 megírása csak egy pillanatra villantja fel a képkeretet (a megírás pillanatától a képvisszafutásig). Ha a képkeretet változtatni akarjuk, akkor a BORDER változót kell átírnunk.

Az I, G, R, B értékétől (ami 0 vagy 1 lehet) függően a képpontokhoz tartozó alapszínek intenzitása a következőképpen számítható ki:

zöld : $G \cdot (4/7 + I \cdot 3/7)$

vörös: $R \cdot (4/7 + I \cdot 3/7)$

kék : $B \cdot (4/7 + I \cdot 3/7)$

A képponthoz tartozó világosságérték (fekete-fehér tv-n a képpont fényessége):

$$Y = (4/7 + I \cdot 3/7) \cdot (0.59 \cdot G + 0.3 \cdot R + 0.11 \cdot B)$$

A video üzemmódot a 6-os I/O port B1, B0 bitje jelöli ki:

00: GRAPHICS 2

01: GRAPHICS 4

1X: GRAPHICS 16

A programozó a beállított video módot a PORT06 rendszerváltozóból tudhatja meg (0B13H, 2835), ez a 6-os port tükörképe.

A képmegjelenítő hardver regisztereit a 9. Függelék foglalja össze.

A TV-Computer hátoldalán található színekikapcsoló egyrészt letiltja a színsegédvívót (színes tv-n is fekete-fehér lesz a kép), másrészt állapotát a program is érzékeli az 59H-s port 86-os bitjén. Állapotát a VMODE grafikus üzemmód beállító rendszerfunkció veszi figyelembe.

8.2 A 6845-ös képernyővezérlő programozása

A TV-Computeren a kép felépítését az eddig elmondottak szerint a képernyővezérlő végzi az inicializálás során megadott jellemzők alapján. A kép tulajdonságainak beállítására, az információcserére a képernyővezérlőben 18 regiszter található, számozásuk: R0...R17. A CPU és a CRT kontroller regiszterei közötti adatcseréhez először ki kell választani a regisztert (70H, 112-es I/O cím), majd a 71H-s (113-as) I/O cím írásával/olvasásával végrehajtható az adatcsere (10. Függelék).

Az R5-ös regiszterbe a következő BASIC utasításokkal lehet például 1-et írni:

```
OUT 112,5:OUT 113,1
```

Gépi kódban:

```
LD    A,5
OUT   (70H),A
LD    A,1
OUT   (71H),A
```

Van néhány olyan funkciója a 6845-ös képernyővezérlőnek, amelyeket bizonyos szállítók bizonyos típusváltozata nem tartalmaz, ezeket a 10. Függelékben * -gal jelöltük. A TV-Computerbe épített 6845-ösök általában az összes funkciót tudják, de a programozás során a * -gal jelölt funkciók nincsenek kihasználva.

A 10. Függelék a regiszterekbe inicializáláskor írt alapértékeket hexadecimálisan és decimálisan is tartalmazza.

A következőkben áttekintjük az egyes regiszterek funkcióit:

R0 : Megszabja, hogy a teljes, tehát a sorvisszafutást is tartalmazó tv sor hány "karakter" ütemnyi legyen. A beprogramozott érték 99. Mivel a kívánt érték ---1 a beírandó adat, a TV-Computerben a tv soridő 100 karakterütemet jelent. A korábban elmondottak szerint a karakterütem 640 ns, tehát a soridő $100 \cdot 0.64 = 64 \mu s$. Ez megfelel a tv-technikában szabványos 15625 Hz-nek.

R1 : Értéke megszabja, hogy mennyi "karakter" látható soronként. (Logikai "karakter"-ről van szó.) A beírt érték 64. Ebből következően a jelmegjelenítés ideje $40.96 \mu s$.

R2 : A vízszintes szinkronjel pozícióját szabja meg. A TV-Computer nem közvetlenül a képernyővezérlő vízszintes szinkronjelét használja fel a tv vezérlésére. A tényleges szinkronjel pozíciója a programozott szinkronjel-pozíció és a programozott szinkronjel szélesség összegéből adódik. Az R2-be programozott érték $76---1$.

R3 : A két szinkronjel szélességét szabja meg. A valóságban a vízszintes szinkronjel szélessége a TV-Computernél a szinkronjel előtti sor-kioltás időtartamát határozza meg. A függőleges szinkronjel szélességét monostabil multivibrátor állítja be, tehát a programozott érték (3) közömbös.

R4 : Az egy képperiódusra jutó "karaktersorok" számát határozza meg, a beírt érték $78---1$.

R5 : Kiegészítő tv sorok száma. A beírt érték 2. Ez az R4-gyel és az R9-cel együtt azt jelenti, hogy a TV-Computernél 1 kép alatt a tv sorok száma:

$$78 \text{ karaktorsor} \cdot 4 \text{ tv sor} + 2 \text{ tv sor} = 314 \text{ tv sor}$$

(A szabványos tv kép két, 312.5 tv sorból álló félképet tartalmaz, ez jelenti az 50 Hz-es félképfrekvenciát. A TV-Computer ettől kismértékben eltér, a képidő $314 \cdot 64 \mu s = 20.096 \text{ ms}$, ami 49.76 Hz-nek felel meg.)

R6 : A megjelenített "karaktersorok" száma. A beírt érték 60, tehát $60 \cdot 4 = 240$ tv sor a hasznos képfelület a TV-Computernél.

R7 : Függőleges szinkron pozíció, a beírt érték 67---1.

R8 : A cursor elcsúsztatására, a hasznos képet kijelölő DISPTMG elcsúsztatására és a váltottsoros letapogatásra ad lehetőséget. Nem részletezzük a beállítható lehetőségeket, csupán a váltottsoros leképzéshez adjuk meg a beírandó értéket: 0 helyett 1. (A váltottsoros leképzés azt jelenti, hogy a tv képet alkotó tv sorok száma nem egész szám, hanem pl. 312.5, emiatt minden második képszinkronjel a tv sor közepére fog esni, az egymást követő két kép fél tv sorral elcsúszik. Így minden második kép alatt más információt lehet megjeleníteni, tehát kétszeresére növelhető a tv sorok száma, persze a képminőség romlása árán: váltottsoros leképzés esetén a kép villog. TV-Computernél a két kép tartalma ugyanaz, a váltottsoros leképzés csak sorremegést okoz.)

R9 : Megszabja, hogy a logikai "karaktersor" hány tv sorból álljon. A beírt érték 4---1, tehát a "karaktersor" 4 tv sorból áll.

R10: Cursorra vonatkozó információk. B6, B5 engedélyezi ill. letiltja a cursort (00, 01), de lehetne villogtatni is (10, 11). Mivel itt a cursor megszakítás kérésre szolgál, villogtatni nem nagyon célszerű. A B4...B0 jelöli ki, hogy a karaktersor melyik tv sorában kezdődjön a cursor. A beírt érték 3.

R11: A cursor befejező tv sorának kijelölése a karaktersorban. Annak érdekében, hogy a képernyővezérlő képenként csak egyszer kérjen megszakítást, R10-be és R11-be azonos értéket kell írunk.

R12: A képhez kezdő memóriacím H része, a beírt érték 0.

R13: A képhez kezdő memóriacím L része, a beírt érték 0.

R14: A cursorpozíció H része, a beírt érték 14.

R15: A cursorpozíció L része, értéke 255. A cursorpozíció a "karakterek" címével (MA0...MA11) jelent egyezőséget. Az utolsó "karakter" címe:

$$60 \cdot 64 - 1 = 3839$$

A cursorpozíció:

$$14 \cdot 256 + 255 = 3839.$$

Mivel R9 és R10 értéke 3, az utolsó tv sor utolsó "karaktere" váltja ki a megszakítást.

R16: Fényceruzapozíció H rész.

R17: Fényceruzapozíció L rész.

Ezek csak olvasható regiszterek. Az "LPEN" buszcsatlakozópontra impulzust adva, a pillanatnyi MA0...MA11 érték R16-ba és R17-be íródik, innen kiolvasható. (Tehát, ha lenne fényceruza a TV-Computerhez, "karakter" felbontásban lehetne képpontot kijelölni, ez 60 sort és 64 oszlopot jelent.)

8.2.1 PÉLDÁK A KÉPERNYŐVEZÉRLŐ ÁTPROGRAMOZÁSÁRA

1. A képenkénti tv sorok számának átállítása 314-ről 312-re.

Enhez a kiegészítő tv sorszámot (R5) kell átírni:

OUT 112,5:OUT 113,0

Hatására a képernyőn apró mozgó pontok formájában látható színsegédvívó mozgása leáll, a színes karakterek függőleges vonalai összetörnek.

Visszaállítás:

OUT 112,5:OUT 113,2

2. Szabványos, 312.5 sorból álló kép előállítás:

Át kell írni a kiegészítő tv sorszámot (R5) és a váltottsoros leképzést (R8):

OUT 112,5:OUT 113,0

OUT 112,8:OUT 113,1

Hatására a vízszintes vonalak enyhén remegnek, de a képfrekvencia pontosan 50 Hz lesz.

3. Cursor (megszakítás) pozíció átírás:

Legyen a kívánt tv sor sorszáma X (0...239), a kívánt oszloppozíció Y (0...63).

Először meg kell határozni a karaktersorszámot:

$$KS = \text{INT} (X/4)$$

A karaktersoron belüli tv sorszám:

$$TVS = X - 4 * KS$$

A cursorhoz tartozó memóriacím (MA0...MA11):

$$CP = KS * 64 + Y$$

Ezt H és L részre kell bontani:

$$CPH = INT(CP/256)$$

$$CPL = CP - 256 * CPH$$

A regiszterekbe írandó értékek:

R14: CPH

R15: CPL

R10: TVS

R11: TVS

A 239. rasztersor 63. oszlopára az ellenőrző számítást elvégezve:

$$KS = INT(239/4) = 59$$

$$TVS = 239 - 4 * 59 = 3$$

$$CP = 59 * 64 + 63 = 3839$$

$$CPH = INT(3839/256) = 14$$

$$CPL = 3839 - 256 * 14 = 255$$

Soha ne programozzuk a cursort a karaktersor első pozíciójába (Y ne legyen 0, azaz az R15-ös regiszter ne legyen 0, 64, 128, 192). Ilyenkor ugyanis a CRT vezérlő cursor kimenetén "tüske" jelentkezhet, ami kétszeri megszakításkérést jelentene.

4. A teljes 16 k videomemória megjelenítése:

Mint már említettük, a 16 k videomemóriából csak 15 k van kihasználva. A képernyővezérlő átprogramozásával megjeleníthető az utolsó 1 k-hoz tartozó 16 rasztersor is. Ehhez át kell írni az R6-os regisztert (meg-

jelenített karaktersork száma) és a függőleges szinkron pozíciót (R7). Utóbbi azért, hogy a kép függőleges irányban középre kerüljön. Célszerű valamilyen kerettel (BORDER) láthatóvá tenni a hasznos képmezőt. Tehát:

```
SET BORDER 20
OUT 112,6:OUT 113,64
OUT 112,7:OUT 113,69
```

Az R7-es regiszterbe a tv-készüléktől függően kell 68-at vagy 69-et írni, megkeresve a legkedvezőbb függőleges képhelyzetet. A tv-készüléken beállított függőleges képmérettől függ, hogy nem csúszik-e egy kicsit ki a kép alsó vagy felső széle. Normál esetben azért van 240 sor megjelenítve a lehetséges 256 sor helyett, hogy képkicsúszás ne álljon elő.

Az R6-os átírása után a kép lefelé kinyúlik, az R7-es átírása után középre áll. A képernyő utolsó TV sorának jobb szélén néhány színes pont látható, ez annak az eredménye, hogy a bekapcsolás utáni memóriatesztelés alatt a verem egy ideig a videomemóriában volt.

Természetesen írni vagy rajzolni továbbra sem lehet az utolsó sorba. Hasznos információt a legegyszerűbben úgy lehet itt megjeleníteni, ha azt a képernyő más területéről másoljuk át.

Ez megtehető BASIC programmal (egy kicsit lassú) vagy gépi kódú programmal (LDIR utasítás). A következő BASIC program bemutatja a 25 (pontosabban 25.6) soros képmegjelenítést:

```
10 GRAPHICS 4
20 SET BORDER 20
30 SET PALETTE 0, 84, 68, 81
40 OUT 112,6:OUT 113,64
50 OUT 112,7:OUT 113,69
100 SET INK 2
110 PRINT AT 1,1:"25. sor";
120 SET INK 3
130 PLOT 256,956-6*4
```

```

140 PRINT #0:"Ez még lejjebb van."
150 SET INK 1
200 KEZDŐCÍM=49152
210 HOSSZ=16*64
230 TAV=240*64
250 FOR I=KEZDŐCÍM TO KEZDŐCÍM+HOSSZ---1
260 POKE I+TAV,PEEK(I)
270 NEXT
300 CLS
310 FOR I=1 TO 24
320 PRINT AT I,1;;
330 PRINT USING "##":I;:PRINT ". sor";
340 NEXT
400 GET:PRINT

```

A képen látható 6 rasztorsoros szövegelcsúsztatást a video eszközmeghajtón keresztüli kiírással értük el (PRINT #0:), előtte a szöveg bal felső sarkát PLOT-tal pozicionáltuk. Ezekről később lesz még szó.

A továbbiakban egy USR rutint ismertetünk. Ez a rutin a következőképpen hívható:

```
X = USR(3760,VARPTR(A$))
```

Hatása:

- elmenti a képmemória első 12 rasztersornyi tartalmát, azaz 12*64=768 byte-ot
- ezt a területet törli
- az AS-t kiírja a képernyő első sorába
- az első 12 sort átmásolja a 240...251. rasztersorba
- visszaállítja az első 12 sort.

Természetesen AS helyett tetszőleges stringváltozót használhatunk.

A rutin a grafikus üzemmódnak megfelelően AS-ból csak 16, 32 vagy 64 karaktert ír ki (ha AS hosszabb lenne). Üres string törli a 240...251. rastersort.

A gépi kódú program:

```

0EB0 E5      PUSH HL
0EB1 3E 50    LD  A,50H      ; U0,U1,VID,SYS
0EB3 32 03 00 LD  (3),A
0EB6 D3 02    OUT (2),A
0EB8 21 00 80 LD  HL,8000H  ; 12 sor mentés
0EBB 11 20 0F LD  DE,0F20H
0EBE 01 00 03 LD  BC,300H
0EC1 ED B0    LDIR
0EC3 21 00 80 LD  HL,8000H  ; 12 sor törlés
0EC6 11 01 80 LD  DE,8001H
0EC9 01 FF 02 LD  BC,2FFH
0ECC 36 00 LD   (HL),0
0ECE ED B0    LDIR
0ED0 06 01    LD  B,1        ; BTEXT: pozicionálás
0ED2 0E 01    LD  C,1
0ED4 F7       RST 30H
0ED5 03       DEFB 3
0ED6 E1       POP HL
0ED7 23       INC HL
0ED8 4E       LD  C,(HL)
0ED9 06 41    LD  B,65
0EDB 3A 13 0B LD  A,(0B13H) ; PORT6, VMODE
0EDE E6 03    AND 3
0EE0 28 07    JR  Z,COMP     ; 2-es mód, max. 64
0EE2 06 21    LD  B,33
0EE4 3D       DEC A
0EE5 28 02    JR  Z,COMP     ; 4-es mód, max. 32
0EE7 06 11    LD  B,17      ; 16-os mód, max. 16
      COMP:
0EE9 79       LD  A,C

```

```

0EEA B8      CP    B          ; Nagyobb-e?
0EEB 3B 02    JR    C,JO
0EED 05      DEC    B
0EEE 4B      LD     C,B
              JO:
0EEF 06 00    LD     B,0
0EF1 B7      OR     A          ; 0-e a hossz?
0EF2 2B 04    JR    Z,NIR
0EF4 23      INC    HL
0EF5 EB      EX     DE,HL
0EF6 F7      RST    30H        ; VID_BKOUT
0EF7 02      DEFB    2
              NIR:
0EFB 21 00 B0 LD     HL,8000H ; Másolás
0EFB 11 00 BC LD     DE,0BC00H
0EFE 01 00 03 LD     BC,300H
0F01 ED B0    LDIR
0F03 21 20 0F LD     HL,0F20H ; 12 sor visszaállítás
0F06 11 00 80 LD     DE,8000H
0F09 01 00 03 LD     BC,300H
0F0C ED B0    LDIR
0F0E 3E 70    LD     A,70H     ; U0,U1,U2,SYS
0F10 32 03 00 LD     (3),A
0F13 03 02    OUT    (2),A
0F15 C9      RET

```

A programban található rendszerhívásokról később lesz szó.

A gépi kód a következő BASIC program segítségével vihető be:

```

10 POKE 2839,192 ! ST_LIMIT 0F10H → 0F00H
20 FOR I=3760 TO 3863
30 READ A:POKE I,A
40 NEXT
50 OUT 112,6:OUT 113,63
60 OUT 112,7:OUT 113,68

```

```

100 END
1000 DATA 229,62,80,50,3,0,211,2
1010 DATA 33,0,128,17,32,15,1,0
1020 DATA 3,237,176,33,0,128,17,1
1030 DATA 128,1,255,2,54,0,237,176
1040 DATA 6,1,14,1,247,3,225,35
1050 DATA 78,6,65 58,19,11,230,3
1060 DATA 40,7,6,33,61,40,2,6
1070 DATA 17,121,184,56,2,5,72,6
1080 DATA 0,183,40,4,35,235,247,2
1090 DATA 33,0,128,17,0,188,1,0
1100 DATA 3,237,176,33,32,15,17,0
1110 DATA 128,1,0,3,237,176,62,112
1120 DATA 50,3,0,211,2,201,0,0

```

Ez a BASIC program a képernyővezérlőt is átírja úgy, hogy a 240...251. rasztersor látható legyen. A BASIC program a lefuttatás után akár ki is törölhető. A 3760-as címen levő USR rutin hívásával tetszőleges üzenetet tudunk elhelyezni a 25. sorban. A képernyőtartalom első sora nem romlik el, csak egy rövid időtartamra villan fel a kép tetején a 25. sorba írandó szöveg. "Meleg" RESET után elég az 50-es és a 60-as sor utasításait begépelni, hogy a 25. sor újra látható legyen.

A programot a CPU számára lefoglalt verem alján helyeztük el, ezért az ST_LIMIT változót átírtuk. Az első 12 sort is ide mentjük el. Mivel a kifrás alatt nincs képkitöltés, biztos, hogy a verem nem mélyül le veszélyes mértékben, hiszen még közel 1 kbyte verem marad meg.

Megjegyzés: Az 5. fejezetben (rendszer-inicializálás) elmondottak szerint a meleg reset is inicializálja a képernyővezérlőt, ezért bármit változtattunk, reset hatására visszaáll az alapállapot. Ha sikerülne a cursort letiltanunk, vagy a szinkronjeleket úgy elállítani, hogy a kép fut, és emiatt nem tudunk gépelni, "meleg" RESET-tel még megmenthetjük a beírt programot.

8.3 Video rendszerváltozók

A video eszközmeghajtó a következő, a felhasználó számára is hozzáférhető rendszerváltozókat tartalmazza:

L_MODE (0B4BH, 2891)

A vonalkeresztelési módot szabja meg, összhangban a BASIC-ből ismert "SET MODE" utasítással:

- 0: felülírás
- 1: OR
- 2: AND
- 3: XOR

A logikai műveletek a videomemória tartalmára vonatkoznak, tehát kétszínű és négy színű üzemmódban a palettasorszámra, 16 színű üzemmódban magukra a színekre. (l. a "Kezelési útmutató" 4.3.4 pontját.)

L_STYLE (0B4CH, 2892)

A vonaltípust jelenti, l. a "Kezelési útmutató" 4.3.3 pontját. A vonaltípusok ismétlődő elemei a 11. Függelékben is megtalálhatók.

INK (0B4DH, 2893)

Az aktuális tintaszínt adja meg. Tartalma két- és négy színű módban a palettaregiszter sorszámot, 16 színű módban magát a színt jelenti. (A színekre vonatkozóan l. a "Kezelési útmutató"-t és a 9. Függelékét.)

PAPER (0B4EH, 2894)

Az aktuális papírszínt jelenti, az INK-hez hasonlóan.

BORDER (0B4FH, 2895)

Az operációs rendszer megszakításprogram a 0-ás portot (border-regisztert) a BORDER változónak megfelelően állítja be, ezért a keretszín megváltoztatásához ezt a változót kell átírni.

Bitjei:

B7: I

B5: G

B3: R

B1: B

V_FLAG (0B50H, 2896)

Karakter felülírási flag.

A video eszközmeghajtón keresztül történő nyomtatás (tehát PRINT#0:, ill. a 0-ás osztályra vonatkozó 1-es és 2-es funkcióhívás) esetén hatásos.

A B1 bit vonatkozik a karaktermező háttérének (papír) felülírására, B1=1 esetén nincs felülírás (mintha egy a képre helyezett átlátszó papírra írnánk).

A B0 bit vonatkozik a karaktermező hasznos pontjainak (tinta) felülírására, B0=1 esetén nincs felülírás. (Szintelen tintánk van.)

Tehát egy karaktert írva, B1, B0 értékétől függően:

00 (0) esetén a karaktermező teljes 8*10-es pontmátrixa felülíródik a papír és tintaszínnek megfelelően;

01 (1) esetén nem változnak azok a képpontok, ahová a tinta színével kijelölt pontok esnének. (A tinta szintelen.)

10 (2) esetén nem változnak az új karakter háttéréhez tartozó pontok, tehát ebben az esetben két karaktert egymásra lehet írni. (A papír "átlátszó".)

11 (3) esetén nem változik az írt karaktermező. ("Színtelen" tinta, átlátszó" papír.)

Ha a V_FLAG értelmezése nem egészen érthető, írjuk be a következő BASIC programot:

```
10 GRAPHICS 16:SET INK 12
20 PLOT 320,480
30 SET PAPER 9:PRINT #0:" ";
40 SET PAPER 10
50 PLOT 320,480
60 FOR I=0 TO 3
70 POKE 2896,I ! V_FLAG
80 PRINT #0:"T";
90 NEXT
100 SET PAPER 0
```

A 10...30-as sorok eredményeként a képernyő közepén megjelenik egy 4 betűhelynyi kék mező. Ezután vörös papírszínt jelölünk ki (40), a tinta zöld (10-es sor). Az 50-es sor a sugarat a 4 betűhely első betűjére pozicionálja.

A 60...90-es sorokból álló ciklus a V_FLAG-be rendre 0,1,2,3-at, a kijelölt 4 karakterhely 1...4 karakterébe egy T betűt ír.

A futtatás után:

- Az 1. karakter vörös háttéren zöld T, tehát a karaktermező teljesen felülíródott.
- A 2. karakter vörös háttéren kék T betű, tehát ahová a T betű pontjai estek volna, a mező színe nem változott (a tinta "színtelen".)

- A 3. karakter kék háttéren zöld T betű, tehát csak a T betűhöz tartozó pontok változtak (a papír "átlátszó").
- A 4. karakternél megmaradt a kék háttér, tehát "átlátszó" papírra "színtelen" tintával írtunk.

A video rendszerváltozóknál a nem értelmezett biteket nem veszi figyelembe a rendszer, tehát pl. GRAPHICS 2-ben a 3-as szín 1-nek, a 2-es szín 0-nak felel meg. Ez nem teljesen igaz. Pl.: parancsállapotba visszatéréskor nem lehet azonos a tinta és papírszín, a rendszer ilyenkor automatikusan növeli a tintaszín sorszámát, tehát a

SET PAPER 2:SET INK 2

eredménye 2-es papír és 3-as tintaszín lesz (csak a BASIC parancsmódjában!). De a

SET PAPER 2:SET INK 6

eredménye vörös papírra vörös tinta, tehát nem olvasható írás.

8.4 Video funkciók

A video eszközmeghajtó 240 rastersorban 512, 256 vagy 128 fizikai képpont ábrázolását teszi lehetővé, ami a grafikus üzemmódok egységes kezelése érdekében 960 függőleges és 1024 vízszintes logikai pontnak felel meg. A fizikai koordináták meghatározására végzett osztás során az 1.3 verziójú rendszer kerekít, az 1.2 verzió elhagyja a maradékot.

A video eszközmeghajtó egy képzeletbeli sugárral vagy tollal dolgozik. A toll lehet letett állapotban (bekapcsolt sugár) vagy felemelt állapotban (kikapcsolt sugár). A toll letett állapotában mozgása közben vonalat húz.

A sugarat vagy tollat tetszőleges pontba mozgatva, oda karaktereket lehet írni. Egy karakter 10 rastersorban 8 fizikai pontot foglal el.

A video funkciókat a 12. Függelék foglalja össze.

8.4.1 VID_CHOUT

Karakterkiírás a képernyőre

Kód: 1H (1)

Input: C = a kiírandó karakter kódja

Hibakód: -

A VID_CHOUT kirajzol egy karaktert az aktuális sugár (toll) pozícióba, ez a pozíció a karaktermező bal felső képpontját jelöli ki. (BASIC-ből a kijelölés a PLOT utasítás segítségével történhet.)

Nemcsak a megszokott karakterpozíciókat lehet kijelölni, hanem tetszőleges képpontból lehet a karaktert indítani. (Lehetséges például index vagy ki-tevő írása.)

Ha a kijelölt pozíció túl közel van a képernyő jobb széléhez (nincs 8 raszterpont), akkor a karakter 10 rasztersorral lejjebb, a képernyő bal szélére íródik.

Ha a pozíció túl közel van a kép alsó széléhez (nincs 10 tv sor), nem íródik ki a karakter. Ezért CLS után a "PRINT #0:" hatástalan (az aktuális pozíció 0,0). Előbb PLOT-tal a sugarat pozicionálni kell.

A video eszközmeghajtó a 20H (32)-nél kisebb és a 0DFH (233)-nál nagyobb karaktereket nem veszi figyelembe. Kivétel a RETURN (0DH, 13) és a LINE FEED (0AH, 10) karakter. A RETURN hatására a sugár ugyanabban a sorban a képernyő bal szélére pozicionálódik. A LINE FEED hatására a sugár 10 rasztersorral lejjebb mozdul, de hatástalan a képernyő alján.

A VID_CHOUT az L_MODE, az INK, a PAPER és a V_FLAG változók figyelembevételével rajzolja ki a karaktert a képernyőre. A karaktergenerátor "1" értékű bitjei tinta színnel, a "0" értékű bitek papír színnel rajzolódnak.

Képernyő rollozás (felfelé mozgás) VID_CHOUT hatására nem történik.

A video rutincsomaggal kirajzolt karakterek a képernyőszerkesztő (editor) számára nem látszanak, az így beírt szöveg nem olvasható be az editor segítségével. (A karakterek kódja nem kerül be az ASCII képmemóriába, erről a képernyőszerkesztő leírásánál lesz szó.)

8.4.2 VID_BKOUT

Szövegírás a képernyőre.

Kód: 2H (2)

Input: DE = a kiírandó memóriaterület kezdőcíme
BC = a kiírandó karakterek száma

Hibakód: -

A `VID_BKOUT` a memóriából sorra olvassa a karaktereket és a karakteres output segítségével kiírja a képernyőre.

8.4.3 BTEXT

Sugárpozicionálás normál karakterpozícióba.

Kód: 3H (3)

Input: B = oszlop pozíció (1 ... 16, 32, 64)
C = karaktorsor pozíció (1 ... 24)

Hibakód: 0F9H (249): nem megengedett pozíció

A BTEXT a BASIC-ből ismert PRINT AT-hez hasonlóan a sugarat a kijelölt karakterpozícióba mozgatja. A kezdő (bal felső) pozíció 1.1. Ha híváskor 0 értékű koordinátát adunk meg, a megfelelő koordináta nem változik. Ez a funkció soha nem húz vonalat.

8.4.4 VMODE

Grafikus üzemmód beállítás.

Kód: 4H (4)

Input: C = grafikus mód (0 = kétszínű mód)
(1 = négyszínű mód)
(2 = tizenhatszínű mód)

Hibakód: 0F7H (247): nemlétező grafikus mód

A VMODE hívás beállítja a kívánt grafikus üzemmódot. Törli a képernyőt, a video eszközmeghajtót és a képernyőszerkesztőt alapállapotba hozza. Két- és négy színű módban beállítja a színpalettát a színikapcsoló állásától függően:

Kétszínű mód, fekete-fehér állás: fekete, fehér

Kétszínű mód, színes állás: fekete, zöld

Négy színű mód, fekete-fehér állás: fekete, fehér, zöld, vörös

Négy színű mód, színes állás: fekete, zöld, vörös, kék

Két- és négy színű módban a papírszín 0-ra, a tintaszín 1-re állítódik.

GRAPHICS 16-ban a papírszín 0, a tintaszín a színikapcsolótól függően fehér (15) vagy zöld (12).

8.4.5 CLS

Képernyőtörlés

Kód: 5H (5)

Input: -

Hibakód: -

A képernyő kitöltődik az aktuális papírszínnel, a grafikus koordináta (sugár, toll) a 0,0 pozícióra áll (a képernyő bal alsó sarka), a sugár kikapcsolt állapotba kerül (felemelt toll). Az ASCII képmemória kitöltődik, az editor cursor az 1.1 pozícióba áll (bal felső sarok).

8.4.6 BA8S

Sugármozgatás a megadott koordinátákba

Kód: 6H (6)

Input: BC = X koordináta
DE = Y koordináta

Hibakód: 0F9H (249): nem megengedett pozíció

Az X, Y koordinátákkal megadott abszolút pozícióba mozgatja a rajzoló sugarat.

X = 0 ... 1023

Y = 0 ... 959

A bal alsó sarok koordinátái: 0,0.

Ha a sugár be van kapcsolva (a toll letéve), akkor mozgás közben vonalat húz a korábbi sugárpozícióból az új pozícióba. Tehát adott pontból adott pontba a következőképpen lehet vonalat húzni:

- sugárkikapcsolás
- pozicionálás a kezdőpontba
- sugárbekapcsolás
- pozicionálás a végpontba

Több szakaszból álló törtvonal rajzolása:

- sugárkikapcsolás
- pozicionálás a kezdőpontba
- sugárbekapcsolás
- pozicionálás az első töréspontba
- pozicionálás a második töréspontba
- stb.

A vonalhúzás az L_MODE, az L_STYLE az INK és a PAPER változókat veszi figyelembe. (A PAPER változónak a nem folyamatos vonalaknál van szerepe, a "kihagyott" vonaldarabkák a papír színével rajzolódnak.)

8.4.7 BREL

Sugárpozicionálás relatív elmozdulásokkal

Kód: 7H (7)

Input: BC = X elmozdulás
DE = Y elmozdulás

Hibakód: 0F9H (249): az elmozdulás eredménye nem megengedett pozíció

A BREL hatása hasonló a BABS-hoz. Különbség: BC és DE nem abszolút koordinátákat jelentenek, hanem az aktuális ponthoz képest relatív elmozdulást.

8.4.8 BON

Sugárbekapcsolás

Kód: 8H (8)

Input: -

Hibakód: -

A sugarat bekapcsolt állapotba hozza (leteszi a tollat) és rajzol egy pontot az INK és az L_MODE változók figyelembevételével az aktuális pozícióba. Ezután a BABS, BREL hívások vonalhúzást eredményeznek.

8.4.9 BOFF

Sugárkikapcsolás

Kód: 9H (9)

Input: -

Hibakód: -

Kikapcsolja a sugarat. Ezután a BABS, BREL nem fog vonalat húzni.

8.4.10 FILL

Képkitöltés

Kód: 0AH (10)

Input: -

Hibakód: -

A FILL az aktuális sugárpozícióból kiindulva az aktuális tintaszínnel átfesti a képernyő azon pontjait, amelyeknek színe megegyezik az aktuális sugárpozíció színével mindaddig, amíg más színű pontokba nem ütközik. Tehát kikapcsolt állapotban a sugarat egy zárt görbe belsejébe pozicionálva, FILL hívással a zárt alakzat kifesthető. A kitöltés "özönvíz" algoritmussal történik. Maga a kitöltő rutin rekurzív, ezért bonyolult alakzatoknál erősen mélyíti a vermet. A rutin a veremmélyítés előtt ellenőrzi, hogy elérte-e az ST_LIMIT által megszabott határt, nehogy az operációs rendszer változói, adatterülete megsérüljön. Ha a verem nem elég, nem garantált a hibátlan kifestés. A képernyőn lefelé haladó képkitöltés kevesebb vermet használ, mint a felfelé haladó.

8.4.11 DEFC

Karakterdefiniálás

Kód: 0BH (11)

Input: D = karakterkód
DE = a karakter mátrixot meghatározó 10 byte címe

Hibakód: 0F8H (248): nem definiálható karakter

A definiálható, ill. újradefiniálható karakterek definiálására szolgál. Definiálás után a megadott kódú karaktert nyomtatva az új karakter jelenik meg a képernyőn, de nem változnak a képernyőre korábban kiírt, azonos kódú, más formájú karakterek.

Nem definiálhatók a 80H-nál (128)-nál kisebb és a 0DFH-nál (223)-nál nagyobb kódú karakterek.

Ne feledjük el, hogy a definiálható karaktereket a meleg reset is újra tölti, tehát az általunk definiált karakter meleg reset hatására átíródik.

A DEFC hívás helyett elegendő a megfelelő RAM terület átírása is (a 740H-s, azaz 1856-os címtől kezdődő 96*10 byte).

8.4.12 PAL

Színpaletta-definiálás

Kód: 0CH (12)

Input: DE = a palettaszíneket tartalmazó 4 byte-os string
címe

Hibakód: -

Két- és négy színű módban lehet a 0,1 ill. a 0,1,2,3 szín sorszámkhoz tartozó színeket definiálni. Hívása helyett közvetlenül is írhatók a 60H...63H-s portok.

8.5 Programozási példák

1. Folyamatos szövegmozgatás

Az eddigiekből kiderült, hogy a video eszközmeghajtóval tetszőleges képpontpozícióba lehet írni. Ezzel folyamatosan mozgathatunk szöveget, BASIC programból is:

```
10 GRAPHICS 4
20 FOR I=1 TO 916 STEP 4
30 PLOT I,36+I
40 PRINT #0:"TVC"
50 NEXT
60 GET:CLS
```

2. Képpontpozíció meghatározása karakterrajzoláshoz

Ha a képernyő Y. sorába és X. oszlopába egy B betűt akarunk írni, akkor a PRINT AT-et használjuk:

```
PRINT AT Y,X:"B";
```

Ha az X karaktersorhoz képest DY rásztersorral lejjebb, az X. oszlophoz képest DX ponttal jobbra akarjuk a karaktert rajzolni, akkor a video eszközmeghajtón keresztül kell nyomtatni:

```
PX = 4*((X---1)*8+DX):PY=999---4*(10+Y+DY)
PLOT PX,PY: PRINT #0:"B"
```

PX-nél a 4-es szorzó csak GRAPHICS 4 esetén igaz, GRAPHICS 2-ben 2, tizenhatszínű módban 8 a szorzószám.

Ha csak függőleges irányú eltolást akarunk (GRAPHICS 4-ben):

```
PX = 32*X---32:PY=999---40*Y---4*DY  
PLOT PX,PY:PRINT #0:"B"
```

Nyomtassunk tört kifejezést vízszintes törtvonallal (pl. a $C = A/B-t$):

```
10 PRINT AT 2,1:"C=";CHR$(155)  
20 PX=32*3---32: PY=999---40*2+4*6  
30 PLOT PX,PY: PRINT #0:"A"  
40 PY=PY---4*13  
50 PLOT PX,PY: PRINT #0:"B"  
60 PRINT
```

9. A BILLENTYŰZET

9.1 Billentyűzet hardver

A TV-Computer billentyűzetéhez funkcionálisan hozzátartozik a beépített botkormány és kívülről csatlakoztatható két külső botkormány is.

A billentyűzet nyomógombjai mátrixba vannak rendezve, a mátrix (beleértve a botkormányokat is) 10 sorból és 8 oszlopból áll. A működtetett nyomógomb összeköti azt a sor- és oszlopvezetékét, amelyek keresztezési pontjában a gomb található.

A sorvezetékek egy 10 kimenetű dekóderre vannak kötve, az oszlopvezetékek jelei buszmeghajtón keresztül a CPU adatbuszára kapcsolódnak. A PORT03 alsó 4 bitje vezérli a tízkimenetű dekódert, ezért a PORT03-ba írt 0...9 számnak megfelelően valamelyik sorvezeték logikai 0 állapotba kerül. Ha lenyomunk egy, a kiválasztott sorban levő nyomógombot, a megfelelő oszlopvezetéken is 0 lesz a jelszint. Ezért a CPU a 0FFH-től eltérő értéket fog beolvasni. A billentyűzet oszlopvezetékeinek állapota az 5BH-s portról olvasható be.

A billentyűzet mátrix elrendezése a 13. Függelékben látható. A 8-as számú sorvezeték a beépített botkormányt és az ezzel párhuzamosan kapcsolódó jobb oldali külső botkormányt választja ki. (Ez az elülső csatlakozó, a "Kezelési útmutató"-ban 13-as számmal van jelölve.) A 9-es sor választja ki a másik külső botkormányt.

A Videoton gyártmányú botkormányon a 4 irány kijelölésén kívül még az alábbi vezetékek találhatók:

- normál tűz (FIRE): a kar tetején levő és a vele egyező színű további 2 tűzgombra van kötve.
- második tűz: a maradék két tűzgombra van kötve. A botkormány oldalán található kapcsolóval ezek a tűzgombok összeköthetők egy ún. gyorsító gyűrűvel. Ebben az állásban a második tüzet a botkormány bármilyen irányú erősebb nyomása is kiváltja, tehát ez a jel az irányított figura gyorsabb mozgatására használható. A továbbiakban ez a vezeték az ACC (gyorsítás) néven szerepel.

Felhívjuk a figyelmet, hogy az egyes nyomógombok nincsenek diódával szétválasztva, ezért 2-nél több nyomógomb egyidejű lenyomása hibás működést válthat ki:

- nyomjuk le az "A"-t, eredménye: "a";
- miközben "A"-t lenyomva tartjuk, nyomjuk le a "D"-t is, eredménye: "d";
- nyomjuk még le az "E"-t is, eredménye: "e" és "q".

(Célszerű előtte SET DELAY 255-öt begépelni.)

Magyarázat: az "A" és "D" egy sorban (4), a "D" és "E" egy oszlopban (B1) van. Amikor a processzor a 2-es sort vitte 0-ba, a B1 oszlopvezetéken keresztül az "E" és "D" össze volt kötve, a 2-vel együtt a 4-es sorvezeték is 0 lett, ezért a lenyomott "A" betűn keresztül a B6-os bit is 0 állapotba került. Tehát a processzor a 2-es vezeték aktiválásakor B1-et és B6-ot érezte 0-nak, ez pedig az "e" és "q" betűnek felel meg. Kedvezőbb elrendezésben több gomb egyidejű lenyomása sem fog hibát eredményezni.

9.2 Billentyűzet szoftver

A billentyűzet szoftver legnagyobb része a rendszer megszakítás-alprogramban fut. Minden 20 ms-ban letapogatja a teljes billentyűzet mátrixot, a letapogatás eredménye a PICTURE nevű, 10 byte-os változóba kerül. A 10 byte jelenti a billentyűzet 10 sorát. A lenyomott állapotban levő nyomógombokhoz tartozó bitek értéke 1 lesz. Pl. az "E" betűt lenyomva, a 2-es sor 81-es oszlopának megfelelően a 0B53H-s cím tartalma 2 lesz. A kódváltásra szolgáló ALT, SHIFT, CTRL és LOCK nyomógombok nem állítanak be bitet a PICTURE-be, hatásukat más változókon keresztül érvényesítik.

A billentyűzet szoftver minden nyomógombhoz hozzárendel egy kódot, amely kód a SHIFT, CTRL és az ALT nyomógombtól, ill. a LOCK állapottól függ. Az egyes nyomógombokhoz tartozó kódokat a 14. Függelék tartalmazza. Tartós nagybetű vagy ALT karakter gépelésére szolgálnak a LOCK állapotok:

- SHIFT+LOCK benyomása: SHIFT LOCK állapot, inverz "S" cursorral. SHIFT LOCK-ban a kisbetűk helyett a nagybetűk, számok helyett az írásjelek kódját állítja elő a billentyűzet.
- CTRL+LOCK benyomása: CAPS LOCK állapot, inverz "C" cursor. Hasonló a hatása a SHIFT LOCK-hoz, de csak kisbetű-nagybetű váltás következik be, szám-írásjel váltás nem.
- ALT+LOCK benyomása: ALT LOCK állapot, inverz "A" cursor. A definiálható (160-nál nagyobb) kódú karakterek megjelenítésére szolgál.

LOCK állapotból a LOCK nyomógomb önmagában történő benyomásával lehet kihozni a billentyűzetet.

SHIFT és CAPS LOCK-ban a SHIFT nyomógomb, ALT LOCK-ban az ALT nyomógomb visszaváltó hatású. Ez azt jelenti, hogy a SHIFT nyomógombot benyomva pl. a nagy "A" betű helyett a kis "a" betű áll elő SHIFT vagy CAPS LOCK-ban.

A CTRL kódok előállítása a LOCK állapotoktól függetlenül helyesen történik.

Speciális szerepe van CTRL+P-nek, ill. a CTRL+ESC-nek.

CTRL+P: Ha a HOLD mód engedélyezve van (HOLD_DIS=0), akkor a CTRL+P az ún. HOLD módba viszi a gépet. Ez azt jelenti, hogy a gép a billentyűzet megszakításprogramba "beragad" addig, amíg valamilyen gombot meg nem nyomnak, tehát leáll pl. a kiírás. Ilyenkor még az INTINC számláló értéke sem növekszik.

CTRL+ESC: A két billentyű egyidejű lenyomása a STOP_FLAG nevű változót OFFH-ba írja, ill. egyes operációs rendszer funkciók STOP hibajelzéssel (0F5H, 245) térnek vissza. A CTRL+ESC a BASIC programok megszakítására szolgál, de gépi kódú programok is figyelhetik a STOP_FLAG változót, ill. a STOP hibajelzést.

Figyelem! Az OS funkciók mindaddig visszaadják a STOP hibajelzést amíg a STOP_FLAG változót nem töröltük. A törlést a BASIC elvégzi, de gépi kódú programok esetén erről a programozónak kell gondoskodnia.

A billentyűzet rendszerváltozóit a 15. Függelék foglalja össze. Eddig már volt szó a PICTURE, a STOP_FLAG és a HOLD_DIS nevű változókról.

Az OLDPICT a megelőző billentyűzetmátrix letapogatás eredményét tárolja, ezt az AUTO-REPEAT-hoz használja a szoftver. Az AUTO-REPEAT funkció lényege: egy nyomógombot tartósan lenyomva, bizonyos késleltetés után a billentyűzet folyamatosan adja a lenyomott gomb kódját. A DELAY_KEY nevű változó szabja meg az AUTO-REPEAT indulási késleltetését, a RATE_KEY az ismétlés szaporaságát, mindkettő 20 ms-os egységekben programozható. Az AUTO-REPEAT speciálisan működik a botkormányoknál. Amíg normál esetben két gomb lenyomásakor az utolsó kódja ismétlődik, a botkormányoknál az irányokhoz tartozó kódok váltakozva ismétlődnek.

A LOCK_KEY változó a LOCK állapotokat tárolja:

B0 = 1 : CAPS LOCK
B1 = 1 : SHIFT LOCK
B3 = 1 : ALT LOCK

(A LOCK állapot programból a LOCK_KEY változó átírásával is kiváltható.)

9.3 Billentyűzet funkciók

A billentyűzetmeghajtóhoz két funkció tartozik (15. Függelék):

- karakter input
- állapotolvasás

9.3.1 KBD_CHIN

Karakterbeolvasás

Kód: 91H (145)

Input: -

Output: C = a beolvasott karakter

Hibakód: 0F5H (245): CTRL+ESC-t (STOP) nyomtak.

Ez a rutin megvizsgálja a billentyűzet állapotát jelző belső flaget, "kész" esetén a "C" regiszterben átadja a lenyomott nyomógombhoz tartozó kódot. A CTRL+ESC a 0F5H-s hibajelzés mellett a STOP_FLAG változót is 0FFH-ba írja.

Ez a rutin nem villogtatja a cursort, amíg a karakter leütésére vár.

9.3.2 KBD_STATUS

Billentyűzetállapot olvasás

Kód: 93H (148)

Input: -

Output: C = FF : karakter beolvasható a billentyűzetről
= 00 : még nem nyomtak le nyomógombot

Ezzel a rutinnal oldható meg, hogy a program ne "ragadjon" be a KBD_CHIN rutinba a nyomógombok lenyomására várva. A KBD_CHIN-t elég akkor meghívni, ha a KBD_STATUS "kész" állapotot jelez.

9.4 Programozási példa

BASIC programokból a figurák mozgatásához a billentyűzet vizsgálatára a GET utasítás és az INKEY\$ függvény helyett célszerűbb a PICTURE elemeinek vizsgálata, ez több gomb egyidejű lenyomása esetén kedvezőbb eredményt ad.

Ha GET A\$-ral olvassuk be a lenyomott karaktert, az irányok vizsgálatával ferde irányban csak "lépcsős" mozgás érhető el, pl. 1 lépés jobbra, 1 lépés fölfelé. A PICTURE vizsgálatával ferde irányban is elérhető folyamatos mozgás. A következő BASIC program egy képpontot mozgat a képernyőn, a képpont mindig otthagyja a nyomát, tehát vonalat húz. Figyelembe veszi a Videoton botkormány gyorsítás jelét is, gyorsításkor a lépésköz két pont lesz.

```
10 GRAPHICS 4
```

```
20 X=512 :Y=428 : PLOT X,Y ! A kezdőpont középen van.
```

```
100 A=PEEK(2905) ! A botkormány sora PICTURE-ban.
```

```

110 DX=0 : DY=0 ! Elmozdulások
120 IF A AND 16 THEN LÉPÉS=8:ELSE LÉPÉS=4
121 ! A 120-as sor a gyorsítást vizsgálja
130 IF A AND 2 THEN DY=LÉPÉS ! UP
140 IF A AND 4 THEN DY=-LÉPÉS ! DOWN
150 IF A AND 32 THEN DX=LÉPÉS ! RIGHT
160 IF A AND 64 THEN DX=-LÉPÉS ! LEFT

200 ! X,Y növelés, átgördítés a képhatáron
210 X=X+DX
220 IF X<0 THEN X=1020
230 IF X>1020 THEN X=0
240 Y=Y+DY
250 IF Y<0 THEN Y=956
260 IF Y>956 THEN Y=0

300 PLOT X,Y
310 GOTO 100

```

Ha nincs Videoton botkormányunk, a gyorsítás hatását helyettesíthetjük a tűzzel:

```
120 IF A AND 8 THEN ...
```

vagy az INS nyomógommbal:

```
120 IF A AND 1 THEN ...
```

Ez a program csak a PICTURE használatához szolgál illusztrációként.

Játékprogramoknál próbáljuk meg az irányító nyomógombokat úgy kiválasztani, hogy a korábban említett vezetékek összekapcsolódás ne jöhesse létre.

10. KÉPERNYŐSZERKESZTŐ

A képernyőszerkesztő (a továbbiakban röviden editor) kényelmessé teszi a TV-Computeren a programbegépelést, biztosítja a programok nagyon egyszerű javítását, módosítását.

Az editor logikai eszköz, a video- és a billentyűzetmeghajtóval együttműködve dolgozik: a PAPER és az INK változót az editor és a video-meghajtó közösen használja, a CLS és a VMODE funkció az editort is alaphelyzetbe állítja, közös a karaktergenerátor stb.

A TV-Computerben nincs megkülönböztetve a szövegmód és a rajzoló mód.

Az editor a videomemória-írást nem a videomeghajtón keresztül, hanem közvetlenül végzi, a képernyőn levő grafikus információt felülírja. Nem veszi figyelembe a videomeghajtó V_FLAG nevű változóját sem. A billentyűzet elérése szabványos operációs rendszer hívással történik (KBD_CHIN). Hozzárendeléssel más input forrás is kijelölhető.

A szerkesztési műveletekhez a képmemóriába írt képpontbiteken kívül az editor a megjelenített karakterek kódját (az ún. ASCII kódot) is eltárolja. Erre a célra a 100H (256) címtől kezdve $25 \cdot 64 = 1600$ byte van lefoglalva. A 25. sor nincs kihasználva, itt elhelyezhetők kisebb gépi kódú rutinok.

A képernyőszerkesztő 16, 32 vagy 64 karakteres módban dolgozik a grafikus módnak megfelelően (16 szín, 4 szín, 2 szín). A grafikus mód váltás (VMODE hívás vagy a GRAPHICS parancs) az editort is alaphelyzetbe állítja és beállítja a belső változókat az üzemmódnak megfelelően.

Ha az editor karakter inputra vár, akkor az ED_INT megszakítás rutin villogtatja a cursort. A cursor típusa megfelel a LOCK módnak, ill. a LOCK_KEY változónak:

- normál mód : 7FH (téglalap)
- CAPS LOCK : 9EH (inverz "C")
- SHIFT LOCK : 9FH (inverz "S")
- ALT LOCK : 8FH (inverz "A")

Az inverz "C", "S" és "A" karaktereket újra lehet definiálni, így a háromféle LOCK cursor módosítható. A cursor villogási üteme nem változtatható, a cursor nem tiltható le. A villogtatás során az editor a cursor alatti területhez tartozó képmemóriarészt menti és állítja vissza (10, 20, 40 byte), ezért a cursor a grafikus ábrát sem rontja el.

Az editor a képernyőn több összetartozó fizikai sort logikai sornak, "bekezdésnek"-nek tekint. A bekezdésen belül a cursorral mozogva különböző szerkesztő műveletek hajthatók végre mindaddig, amíg a bevitelt a RETURN karakterrel le nem zártuk. A szerkesztési műveletek (beszúrás, kiejtés) során a módosítási helytől a bekezdés végéig megtörténik a sormaradék igazítása. Tehát nemcsak a fizikai sorvég mozog, hanem a mozgás több fizikai soron keresztül is tarthat. Ha a beírás vagy beszúrás során a fizikai sor betelik, automatikusan egy új, üres fizikai sor adódik a bekezdéshez. Ha a fizikai sor utolsó pozíciójában levő karaktert töröltük, a következő üres fizikai sor - amely a bekezdéshez tartozik - felszabadul.

Az editorral a képernyőn levő akármelyik sor módosítható és újra beolvasható, függetlenül attól, hogy a kezelő vagy a processzor írta-e azt a képernyőre. (Ez nem igaz a videomeghajtó által kiírt sorokra.)

Az editorhoz nem tartoznak a felhasználó számára is hozzáférhető rendszerváltozók.

10.1 Editor funkciók

Az editor funkciókat a 16. Függelék foglalja össze.

10.1.1 ED_INT

Editor megszakítás (interrupt) rutin

Kód: 20H vagy A0H (32 vagy 160)

Input: -

Output: -

Az editor megszakítás rutin villogtatja a cursort, amíg az editor karakterbegépelésre vár. A felhasználó az ED_INT-et ne hívja!

10.1.2 ED_CHIN

Karakterbeolvasó rutin

Kód: 0A1H (161)

Input: -

Output: -

Az ED_CHIN meghívásával az editor a karakterbeolvasó-szerkesztő programhurokba lép: a leütött karaktereket beolvassa a billentyűzetről és kiírja a képernyőre. Közben végrehajthatók a BASIC-ből ismert szerkesztési

műveletek. Az editor mindaddig a szerkesztő hurokban marad, amíg a billentyűzeten nem nyomnak:

- ESC-et,
- CTRL+ESC-et,
- RETURN-t.

ESC hatására kilép a szerkesztő hurokból és visszaadja az ESC kódját (1BH, 27).

CTRL+ESC hatására visszaadja az ESC kódját és "STOP" hibajelzést állít be.

RETURN hatására az ED_CHIN visszaadja a "bekezdés" első karakterét. Ismételt hívás esetén a "bekezdés" további karaktereit adja át a hívónak egészen addig, amíg a bekezdés utolsó karakterét, a RETURN-t is meg nem kapja a hívó. További ED_CHIN hívásra az editor ismét a beolvasó-szerkesztő hurokba lép.

A RETURN beolvasása (a teljes bekezdés átadása) után a cursor a következő bekezdés első karakterére áll, ha nincs több bekezdés, a következő üres sor elejére. A képernyő alján rollozás történik, tehát alulról belép egy üres sor, a legfelső sor elveszik.

10.1.3 ED_CHOUT

Karakterkiküldés

Kód: .21H (33)

Input: C = karakterkód

Output: -

A nyomtatható karaktereket (20H...0DFH, 32...223) kiírja a képernyőre és a cursort eggyel jobbra lépteti (sor végén a következő sor elejére). A

képernyőn levő karaktereket felülírja mindaddig, amíg a cursor egy bekezdés belsejében van. Ha a kiírás eléri a bekezdés utolsó sorának az utolsó pozícióját, egy új üres sort szúr be és az alatta levő sorok lefelé mozognak (kivéve, ha a bekezdés után üres sor van). Ha a cursor az utolsó sorban van, a sor megtelése rollozást vált ki.

A kiírás az aktuális INK és PAPER színnek megfelelően történik. A beszúrás, kiejtés, rollozás során a papír- és tintaszín megmarad.

A 20H-nál kisebb (CTRL) karakterekre az editor ugyanúgy végrehajtja a szerkesztési műveleteket, mintha azokat a billentyűzetről vittük volna be a programbegépelés során.

Kivételek:

RETURN (ODH, 13, CTRL+M) : a cursort a sor elejére mozgatja.

LINE FEED (OAH, 10, CTRL+J) : a cursort lefelé mozgatja a bekezdés utáni sorba, il. a következő üres sorba, változatlan oszloppozícióval. A képernyő alján rollozást vált ki.

Tehát az ED_CHOUT-tal úgy lehet új sort kezdeni, hogy a ODH és a OAH karaktert kell kiküldeni.

10.1.4 ED_BKIN

Blokkbeolvasás

Kód: 0A2H (162)

Input: DE = puffterület kezdőcíme
BC = a beolvasandó karakterek száma

Output: -

Az ED_BKIN a karakteres input ismételt meghívásával dolgozik. Beolvas egy vagy több bekezdést a képernyőről (a bekezdés bevitelét RETURN-nel le kell zárni) mindaddig, amíg a BC-ben megadott értéket a beolvasott karakterek száma el nem éri. Több bekezdés beolvasása esetén a bekezdések között a RETURN karaktert is tárolja. A beolvasott karakterek a DE-ben megadott címtől kezdve találhatók meg a memóriában.

Tehát ha 6 karaktert akarunk beolvasni:

1. 5 karakter + RETURN begépelése esetén a memóriában az 5 karakter + RETURN lesz található.
2. 10 karakter + RETURN begépelése esetén a memóriában az első 6 karakter lesz.
3. 2 karakter + RETURN + 10 karakter begépelése esetén a memóriában az első 2 karakter + RETURN + a 10-ből az első 3 karakter lesz.

Ha több karaktert gépeltünk be, az ED_BKIN ismételt hívása folytatja a beolvasást az előzőleg begépelte karakterekkel.

10.1.5 ED_BKOUT

Blokk-kiírás

Kód: 22H (34)

Input: DE = puffertérület kezdőcím
BC = kiírandó karakterek száma

Output: -

Az ED_BKOUT a karakteres output ismételt hívásával kiírja a DE-vel címzett puffertben tárolt karaktereket. A kiírt karakterek számát BC határozza meg.

10.1.6 CPOS

Cursor pozicionálás

Kód: 23H vagy 0A3H (35 vagy 163)

Input: C = karaktorsor-pozíció (1...24)
B = oszloppozíció (1...16, 32, 64)

Output: -

Hibakód: 0F6H (246) : rossz cursorpozíció

A cursort pozicionálja a B és C-ben megadott oszlop és sorpozícióba. Ha valamelyik érték 0, a megfelelő pozíció változatlan marad.

10.1.7 CFIX

Cursorpozíció megjegyzés

Kód: 24H vagy 0A4H (36 vagy 164)

Input: -

Output: -

A CFIX hívást az ED_CHIN vagy az ED_BKIN előtt kiadva, az editor megjegyzi a cursor pozícióját. Ezután az editor input rutinok a RETURN hatására a karakterbeolvasást a megjegyzett cursorpozíciótól kezdik, tehát a sor elején levő szöveg nem kerül beolvasásra (l. az INPUT PROMPT BASIC utasítást.)

10.2 Szerkesztési parancsok

A következőkben összefoglaljuk a BASIC-ből már ismert szerkesztési parancsokat:

RIGHT, CTRL+D	(04H,4)	cursor jobbra
LEFT, CTRL+S	(13H,19)	cursor balra
UP, CTRL+E	(05H,5)	cursor föl
DOWN, CTRL+X	(18H,24)	cursor le
CTRL+I	(09H,9)	cursormozgatás a következő TAB pozícióba
CTRL+K	(0BH,11)	sormaradéktörlés
CTRL+N	(0EH,14)	sorbeszúrás
CTRL+Y	(19H,25)	sorkiejtés
INS	(16H,22)	karakterbeszúrás
DEL	(08H,8)	a cursortól balra álló karakter kiejtése
SHIFT+DEL	(07H,7)	a cursor pozíciójában levő karakter kiejtése

Az editor input funkcióhívásoknál:

ESC	(1BH,27)	a begépelés megszakítása
RETURN	(0DH,13)	a begépelés befejezése

Az editor output funkcióhívásoknál:

RETURN	(0DH,13)	cursor mozgatása a sor elejére
CTRL+J	(0AH,10)	cursor mozgatás lefelé, a képernyő alján rollozás

10.3 Programozási példák

Előfordulhat, hogy a TV-Computer által valamilyen módon kiszámítható adathalmazt akarunk DATA sorokba lerakni, pl.:

- debuggerrel vagy assemblerrel bevitt gépi kódú programból kell DATA sorokat csinálni
- a zenei skálához tartozó PITCH értékeket kell tárolni
- a nyomógombokhoz tartozó kódokat kell tárolni stb.

Ilyenkor a sok, könnyen eltéveszthető szám begépelése helyett programmal íratjuk ki a képernyőre a DATA sorokat. Kihasználva az editornak azt a tulajdonságát, hogy a képernyőn megjelenített információ később input adat is lehet, a programfutás után csak a RETURN-t kell megfelelő számban lenyomni, és máris program lesz a kiírt sorokból.

Az első feladatnál legyen a gépi kódú programrész a 6639...6705 címeken elhelyezve. A DATA sorok sorszámozása kezdődjön 8000-rel. A BASIC program:

```
9000 KSOR=8000
9010 GRAPHICS 2
9020 DB=0
9030 FOR I=6639 TO 6705
9040 IF DB=0 THEN PRINT KSOR;"DATA";:KSOR=KSOR+10
9050 A=PEEK(I)
9060 PRINT STR$(A);",":DB=DB+1
9070 IF DB=8 THEN DB=0:PRINT CHR$(8)
9080 NEXT
9090 PRINT CHR$(8)
```

A program lefuttatása után a cursort a képernyő tetejére kell vinni, és a RETURN-t kell lenyomni annyiszor, ahány sor van a képernyőn.

A zenei skála a PITCH értékeihez ugyanez a program kell, csak a FOR ciklus határait (9030-as sor) és az értékmeghatározást (9050) kell átírni. Pl. a normál "A" hanghoz képest lefelé és fölfelé 2-2 oktáv terjedelem esetén:

```
9030 FOR I=---24 TO 24
9050 F=440*2^(I/12):OA=INT(0.5+195312.5/F):A=4096---OA
```

Ha 20 nyomógomb kódját akarjuk megfelelő sorrendben tárolni:

```
9030 FOR I=1 TO 20
9050 GET A$:A=ORD(A$)
```

A program elindítása után a kívánt sorrendben le kell nyomni a 20 nyomógombot, mire a DATA sorok megjelennek a képernyőn. Ezután már csak a RETURN-ök lenyomása van hátra.

11. HANG

11.1 A hangkezelő hardver

A TV-Computerben a hang előállítására egy 12 bites programozható számláló szolgál, a számláló bemenetére 3.125 MHz-es órajel van vezetve. Az osztásarány beállítására szolgáló 12 bites PITCH értéket a PORT04-be (alsó 8 bit) és a PORT05-be (felső 4 bit: B0...B3) kell beírni.

A számláló osztásaránya: $4096 \div \text{PITCH}$

A $\text{PITCH}=4095$ nem állít elő hangot.

A frekvenciaosztó kimenete frekvenciafelezés után adja a soros vonal adási és vételi óráját, ennek megfelelően az adatátviteli órajel-frekvencia (kHz-ben):

$$1562.5 / (4096 \div \text{PITCH})$$

Ne feledjük, hogy a tényleges átviteli sebességet az USART is befolyásolja, a fenti értéket 16-tal vagy 64-gyel el kell osztani, hogy a baud értéket megkapjuk.)

A hangjel frekvenciája további 8-as osztással adódik, így a kimenő hang-frekvencia (Hz-ben):

$$195312.5 / (4096 \div \text{PITCH})$$

A 8-as osztás után egy kapuzó áramkörön keresztül jut a hang az amplitúdószabályozó áramkörre. Az amplitúdószabályzás 4 bites, tehát a beállítható hangerő 0...15.

Ha a hangot letiltjuk (a PORT05-be írt B4=0-val), akkor az amplitúdóértéknek megfelelő egyenszint jelenik meg a TV-Computer hangkimeneten. Ily módon az amplitúdószabályzó áramkör durva, 4 bites digital/analóg konverterként működhet. Az amplitúdó beállítására a PORT06 regiszter B2...B5 bitjei szolgálnak. A PORT05 és a PORT06 más egységekre vonatkozó biteket is tartalmaz, ezért a két port csak a tükörkép memóriarekesz figyelembevételével írható.

A B-as osztás után a hangjel a PORT05 regiszter B5 bitjével kapuzva szolgál a megszakítás generálására.

Az SBH-s (91-es) port olvasásával a frekvenciaosztó alapállapotból indítható, ez a pontos időzítéshez használható.

A hangra vonatkozó tudnivalókat a 17. Függelék foglalja össze.

11.2 Hangkezelő szoftver

A hangkeltő áramkör vezérlésére egyetlen operációs rendszer funkció (TONE_SET) szolgál, amellyel beállítható a hang frekvenciája és amplitúdója, valamint időtartama. Az időtartam beállítás a cursormegszakítás segítségével történik, 20 ms-os lépésekben.

A hanghoz tartozó rendszerváltozók:

SND_ACTIVE (0B14H, 2836):

Ha értéke OFFH, az előzőekben kiadott hang még nem fejeződött be. Nincs hang folyamatban, ha értéke 0.

SND_IRQ (0B15H, 2837):

OFFH esetén az újabb TONE_SET hívás megszakítja a megelőzőt.
0 esetén az újabb hívás megvárja, hogy az előző hang befejeződjön.

SER_OK (0B71H, 2929):

Ez a változó inkább a soros vonalhoz tartozik. Ha értéke OFFH, akkor hangkeltéssel elrontottuk a soros vonal órajel-frekvenciáját. A 0 érték azt jelzi, hogy a frekvenciaosztó a soros vonal által megkívánt értékre van beállítva.

11.3 TONE_SET

Hangkeltés

Kód: 33H (51)

Input: B = időtartam (20 ms-os egységekben)

C = amplitúdó (0...15)

DE = PITCH érték (0...4095)

Output: -

A TONE_SET az SND_IRQ értékétől függően vagy azonnal, vagy az előző hang befejeződését megvárva hangot szólaltat meg.

A DE regiszterpár értéke szabja meg a hangfrekvenciát:

$$f = 195312.5 / (4096 - DE)$$

DE = 4095 esetén nincs hang.

A B regiszterben kell megadni az időtartamot 20 ms-os egységekben, a tényleges időtartamban 20 ms bizonytalanság van. Tehát B=3 esetén az időtartam 40 és 60 ms között lesz.

12. NYOMTATÓKEZELÉS

12.1 Nyomtatócsatló hardver

A TV-Computerbe épített nyomtató interface párhuzamos Centronics típusú, amely a lehető legkevesebb vezetékkel dolgozik:

- 8 adatvezeték
- föld
- adatérvényesítő jel (/STROBE)
- a nyomtató nyugtázó jele (/ACK)

Az adatokat az 1-es I/O címen levő 8 bites adatregiszterbe írja a processzor. Ezután a 6-os cím B7-es bitjét 0-ba, majd néhány μ s múlva 1-be írva, a /STROBE kimeneten megjelenik az aktív 0 szintű adatérvényesítő impulzus. Ez az impulzus törli az esetleg bebillent állapotban levő ACK flip-flop-ot is. A /STROBE impulzust érzékelve a nyomtató az /ACK vezetéken egy 0 szintű impulzussal jelzi az adat elfogadását és feldolgozását. Ez az impulzus bebillenti az ACK flip-flop-ot, amelyet a processzor az 59H-s port B7 bitjén tud beolvasni. Az /ACK jel alap helyzetbe állítja a STROBE flip-flop-ot (ha a 0 szint túl hosszú ideig lenne kiadva).

A PORT06 írásakor (/STROBE) a PORT06 tükörképet figyelembe kell venni.

12.2 Nyomtató szoftver

A nyomtatómeghajtóhoz rendszerváltó nem tartozik. Két operációs rendszer funkció szolgál az adatkiküldésre (l. a 18. Függelék).

12.2.1 PAR_CHOUT

Karakterkiküldés a nyomtatóra

Kód: 41H (65)

Input: C = karakterkód

Output: -

Hibakód: 0F5H (245) : STOP-ot nyomtak

Nyomtatóhiba (OFF LINE állapot, papírhiány, stb.) esetén a PAR_CHOUT "be-
ragad", vár a hiba elhárítására. Ha a nyomtatást nem kívánjuk folytatni,
STOP-pal a nyomtatás megszakítható.

12.2.2 PAR_BKOUT

Karaktersorozat kiküldése a nyomtatóra

Kód: 42H (66)

Input: DE: a kiküldendő adatsor kezdőcíme

BC: a nyomtatandó karakterek száma

Output: BC: STOP esetén a ki nem küldött karakterek száma

A PAR_BKOUT a karakteres output ismételt hívását jelenti.

13. KAZETTAKÉZELÉS

13.1 Kazettakezelő hardver

A TV-Computer az adatokat, programokat a legegyszerűbb esetben magnókazettán tudja tárolni. Ehhez igen egyszerű hardver tartozik:

- A PORT50H írása (az írt adat közömbös) ellenkező állapotba billent egy flip-flop-ot, a flip-flop kimenőjele amplitúdóosztás után szolgáltatja a magnó számára a felveendő jelet.
- A magnóról jövő jel jelformálás után az 59H-s port B5-ös bitjén olvasható be, a jelátmenetek érzékelését a szoftver végzi.
- Az 5-ös port B6, B7 bitje adja a két magnócsatlakozó motorvezérlő jelét.

Az információt a szalagra írt jelsorozat periódusideje hordozza. A felírás tömbökben történik.

Egy file egy "fej" (header) és 0, 1 vagy több adattömbből áll. A tömbök szektorokat tartalmaznak, egy tömb 1...256 szektort. (Nem lehet 0 szektort tartalmazó tömböt írni.)

A visszaolvasás szinkronizálásához, a szalagsebesség megállapításához a tömbök elején egy speciális jelsorozat található:

1. Bevezető hang, a header tömb elején 10240 periódus, az adattömb elején 5120 periódus.

Periódusidő: 470 μ s

Frekvencia: 2128 Hz
Időtartam: 4.8/2.4 s

2. Szinkronperiódus (1 db), olvasásakor a jel fázisának a megállapítására szolgál.

Periódusidő: 736 μ s
Frekvencia: 1359 Hz

A továbbiakban már byte struktúrájú a felírt információ, a bitek "0" vagy "1" értéke a periódusidőt határozza meg:

"0" bit: 552 μ s, 1812 Hz
"1" bit: 388 μ s, 2577 Hz

Ennek megfelelően a felírási sebesség:

00H-s byte-ok esetén: 226.5 byte/s
0FFH-s byte-ok esetén: 322.2 byte/s
55H-s byte-ok esetén: 266.0 byte/s

A tömb felírása 5 periódus 470 μ s-os bevezető hanggal fejeződik be.

A motorvezérlő jel bekapcsolása és ennek megfelelően a motorindítás kb. 1 másodperccel a bevezető hang előtt van. Az 1 másodperc a szalagsebesség stabilizálódásához szükséges.

A kazettakezelő hardvert a 19. Függelék foglalja össze.

13.2 Adatstruktúra a szalagon

A szinkronperiódust 6 byte követi, ez a 6 byte határozza meg a tömböt:

1. Üres byte: a hosszú szinkronperiódus által a bemenő erősítőben okozott zavar hatásának a kiküszöbölésére szolgál.
2. Szinkronbyte: 6AH (01101010), a szinkronizálás korrektségének további ellenőrzését teszi lehetővé.
3. Tömbtípus-azonosító: 00 - adattömb
0FFH - fejtömb
4. File-típus: 1: pufferezt file
17: nem pufferezt file
5. Írásvédelem-byte, értéke 0 a nem védett file-oknál.
6. Szektorszám: hány szektor van a tömbben.

A szektorok felépítése a következő:

1. Szektorsorszám (1 byte), az első szektor sorszáma 0. Ha az olvasás során a beolvasott szektorszám eltér a várt értéktől, akkor az olvasás hibaüzenettel megszakad.
2. A szektorban levő adatbyte-ok száma (1 byte), a nulla 256 byte-ot jelent. (Nem lehet 0 hosszúságú a szektor.)
3. Adatbyte-ok (1...256 db)

4. File-vég jelzés: 0 az összes szektorban, kivéve az utolsót.

5. CRC (Cyclic Redundancy Check): 2 byte-os ellenőrzőszám.

Először a magasabb, utána az alacsonyabb helyi értékű byte-ot írja fel a TV-Computer.

A kazettakezelő szoftver 1-1 írásra és olvasásra a megnyitott file egyidejű kezelését teszi lehetővé.

Az olvasás-írás során a tömbök között a szalag leállítható, a tömbök alatt nem.

Két fogalmat kell még megmagyarázni:

Pufferelt/nem pufferelt file: A pufferelt file írása során a kazettakezelő szoftver egy átmeneti pufferbe gyűjti az információt, a tényleges felírás akkor történik, amikor a puffer megtelik vagy CLOSE-zal az írást befejezzük. Nem pufferelt file-ról akkor beszélünk, amikor egy nagyobb memóriarészt közvetlenül a memóriából írunk ki (pl. programmentés).

Írásvédett file: írásvédett file-t nem lehet megnyitni olvasásra, ha korábban írásra már nyitottunk file-t. Írásvédett file olvasási megnyitása után nem lehet file-t írásra megnyitni. Másszóval nem lehet egyidejűleg nyitva írásvédett file olvasásra és egy másik file írásra, bármilyen sorrendben akarjuk is a file-okat megnyitni.

13.3 Pufferelt file-ok

A pufferelt file-ok egy fejtömböt és nulla vagy több adattömböt tartalmaznak. Az adattömbök mindig egy szektorból állnak. A BASIC OPEN OUTPUT utasítása pufferelt file-t nyit meg.

A fejtömb adatként csak a file-nevet tartalmazza, az adatok a második tömbtől kezdődően helyezkednek el. Ha a megnyitást adatírás nélkül a file lezárása követi, a file csak fejtömbből fog állni, a fejtömb fogja tartalmazni a file-vég jelzést.

A pufferelt file-ok írása/olvasása a LAPQ-on található állandó helyű pufferen keresztül zajlik.

Két tömb között a motorvezérelt magnó leáll, az adatforgalmazás a magnóval tetszőleges sebességű lehet. Fizikai írás a puffer megtelése után következik csak be, pontosabban a megtelés után küldött első karakter hatására. A fizikai írást a CLOSE is kiváltja.

13.4 Nem pufferelt file-ok

A nem pufferelt file-ok egy egyszektoros fejtömbből és 0 vagy 1 adattömbből állnak. Az adattömb 1...256 szektort tartalmazhat.

A fejtömb a file-nevet tartalmazza és tartalmazhat még maximálisan 16 adatbyte-ot. Ezek az adatbyte-ok karakter írás/olvasással kezelhetők. A tényleges felírást a fejinformációkat követő blokkírási parancs fogja kiváltani.

A fejtömb 16 byte-ja az adattömb jellemzésére szolgál.

Programfile-ok esetén:

- 1. byte: 0
- 2. byte: típus: 1 = programfile, 0 = ASCII file stb.
- 3.-4. byte: hossz
- 5. byte: auto-run üzemmód nem nulla esetén betöltés után a program azonnal indul
- 5.-15. byte: nem használt, értéke 0
- 16. byte: változatszám

13.5 Kazetta rendszerváltozók

BUFFER (0B6BH, 2923)

A file-típus kijelölésére szolgál: 0 - nem pufferelt file
<>0 - pufferelt file

A BUFFER változó értékét az írásra megnyitás előtt kell beállítani, a megnyitott file a BUFFER változó szerint lesz pufferelt vagy nem pufferelt. Az olvasásra nyitás után a BUFFER-ből kiolvasható, hogy a megnyitott file milyen típusú. A BUFFER értéke csak közvetlenül az írási megnyitás előtt ill. az olvasási megnyitás után számít, ezért az írásra és olvasásra egyidőben különböző típusú file-ok is megnyithatók.

REMRED (0B6CH, 2924)

A motorvezérlés kijelölésére szolgál, megszabja, hogy a két magnó-csatlakozó közül melyik motorvezérlő jel lesz az aktív az írási és melyik az olvasási műveleteknél. Jelöljük B-vel a bal oldali, a tápfeszültség csatlakozó melletti magnócsatlakozót, J-vel a jobb oldalit.

A REMRED B7, B6 bitjétől függően:

00 (00H,0)	olvasás: B,	írás: J
01 (40H,64)	olvasás: J,	írás: J
10 (80H,128)	olvasás: B,	írás: B
11 (0C0H,192)	olvasás: J,	írás: B

Az alapkijelölés B7=1, B6=0, tehát íráskor is és olvasáskor is a bal oldali magnócsatlakozó motorvezérlő jele lesz aktív.

Megjegyezzük, hogy a ki- és bemenetek a két magnócsatlakozón párhuzamosan vannak kötve. Tehát ha a magnó motorja nem vezérelhető kívülről, közömbös, hogy melyik csatlakozót használjuk felvételre és melyiket lejátszásra.

PROTECT (0B6DH, 2925)

Ha értéke nem nulla, az írási megnyitás eredménye védett file lesz. A kazettakezelő szoftver ezután törli a PROTECT változót.

EOF (0B6EH, 2926)

Ez a változó nullától eltérő értékű lesz, ha a file-ból az utolsó karaktert is kiolvastuk. Vizsgálatával elkerülhető, hogy az utolsó karakter után file-vég hibát váltson ki az olvasási parancs. Ezt a lehetőséget nem használja a BASIC INPUT utasítás.

MIDDLE (0B6FH, 2927)

Ezt a kétbyte-os változót közvetlenül a file-megnyitás előtt lehet beállítani. Tartalma a CRC számítás gyöke lesz az összes írási és olvasási művelethez a file lezárásáig. Ha értékét átírjuk az írási megnyitás előtt, akkor olvasáskor is át kell írni a MIDDLE változót ugyanarra az értékre, különben a rendszer CRC hibát jelez.

A kazetta rendszerváltozókat a 19. Függelék foglalja össze.

13.6 Kazettafunkciók

A kazettakezelőnél egy-egy hibakód több funkciónál is felléphet, ezért a hibakódokat csak a végén adjuk meg.

A kazettafunkciókat a 20. Függelék foglalja össze.

13.6.1 CAS_CHIN

Karakterbeolvasás

Kód: 0D1H (209)

Input: -

Output: C = karakter

A CAS_CHIN egy karaktert olvas a kazettás file-ból, elsősorban pufferelt file-oknál célszerű használni. Nem pufferelt file esetén maximum 16 karaktert tud olvasni, ill. annyit, amennyit az írási megnyitás után CAS_CHOUT-tal felírtunk. Ha több karaktert akarunk olvasni, belső hiba (0E7H, 231) lép fel. A felírtnál kevesebb karakter olvasása nem vált ki hibaüzenetet. Nem pufferelt file esetén az adattömb beolvasását a blokk input funkció (CAS_BKIN) meghívása váltja ki.

13.6.2 CAS_CHOUT

Karakter kiírása kazettára

Kód: 51H (81)

Input: C = karakterkód

Output: -

A CAS_CHOUT karaktert ír a kazettás file-ba, elsősorban pufferelt file-oknál célszerű használni. Nem pufferelt file esetén maximum 16 karakter írható a CAS_CHOUT-tal. A 17. karakter írása hibát okoz. A teljes file fizikai felírását nem pufferelt file-nál a blokk output funkcióhívás (CAS_BKOUT) váltja ki.

13.6.3 CAS_BKIN

Blokk beolvasása a kazettáról

Kód: 0D2H (210)

Input: DE = memóriapuffer kezdőcíme
BC = a beolvasandó byte-ok száma

Output: BC = a be nem olvasott byte-ok száma (hiba esetén)

A CAS_BKIN blokkok beolvasására szolgál. Nem pufferelt file esetén pontosan kell tudni a kazettán levő byte-számot, ellenkező esetben a kazettakezelő hibát fog jelezni. A kazettakezelő sohasem olvas be a kívánnál több karaktert.

Pufferelt file-nál a blokk nem a kazettára írt tömböt jelenti, hanem az eddig megismert blokkinputokhoz hasonlóan tetszőleges mennyiségű adatot.

13.6.4 CAS_BKOUT

Blokk-kiírás a kazettára

Kód: 52H (82)

Input: DE = memóriapuffer kezdőcíme
BC = a kiírandó byte-ok száma

Output: BC = a ki nem küldött byte-ok száma (hiba esetén)

Memóriatartalom kiírására szolgál. Ne felejtsük el, hogy nem pufferelt file esetén csak egy ilyen blokkírás parancs adható ki.

Pufferelt file-oknál a hossz nem a magnóra írt tömb hosszát fogja jelenteni, a felírás 256 byte-os (egyszektoros) tömbökben történik.

13.6.5 CAS_OPEN

File-megnyitás olvasásra

Kód: 003H (211)

Input: DE = file-név kezdőcím

Output: DE = a megtalált file nevének kezdőcíme

A CAS_OPEN megnyitja a file-t olvasásra a megadott névspecifikáció szerint. A név megadása: hossz+NÉV. Ha a hossz=0, akkor a CAS_OPEN a kazettán talált első file-t nyitja meg, egyébként az adott nevű file-t. A hosszabb file-név 16 byte-ra csönkul, a kisbetűk nagybetűkre cserélődnek. Sikeres megnyitás esetén DE az átalakított, ill. a megtalált file nevére fog mutatni. A CAS_OPEN a megnyitott file típusától függően beállítja a BUFFER változót.

A CAS_OPEN alatt a kazettakezelő a BASIC-ből ismert üzenetekkel jelentkezik ki:

Searching

Found: <file-név>

Reading: <file-név>

13.6.6 CAS_CRTE

File-megnyitás írásra

Kód: 53H (83)

Input: DE = file-név kezdőcím

Output: DE = az átkonvertált file-név kezdőcíme

A CAS_CRTE írásra nyit file-t a megadott névvel, a nevet hossz+NÉV alakban kell a memóriába letenni. A nevet 16 karakterre csonkítja és a kisbetűket nagybetűkké alakítja (konvertálja), az átalakított file-névre mutat visszatéréskor a DE.

A CAS_CRTE a BUFFER és a PROTECT változók figyelembevételével nyitja meg a file-t.

13.6.7 CAS_CLOSE

File-lezárás

Kód: 004H (212): olvasásra megnyitott file esetén
54H (84): írásra megnyitott file esetén

Input: -

Output: -

A CAS_CLOSE a file-ok lezárására szolgál. Az olvasásra megnyitott file-oknál ez csak az input csatorna lezárását jelenti, ezáltal lehetővé teszi újabb CAS_OPEN kiadását. Sokkal fontosabb a hatása írásnál: felírja az utolsó, tört tömböt és lezárja a csatornát.

13.6.8 CAS_VERIFY

Kazettás file és memóriatartalom összehasonlítás

Kód: 0D5H, 213

Input: DE = memória-kezdőcím
BC = hossz

Output: -

A DE regiszterben megadott kezdőcímtől összehasonlítja a memóriatartalmat és a magnószalag tartalmát.

Ellenőrzési (0E8H, 232) hiba keletkezik, ha

- nem pufferelt file-nál a BC-ben megadott hossz nem egyezik a beolvasott blokk hosszával,
- a beolvasott adat és a memóriatartalom eltér egymástól.

Pufferelt file esetén a nem egyező byte memóriacíme a DE-ben, a még ellenőrizendő hossz (beleértve a hibás karaktert) a BC-ben van. Az eltérő byte-ot a kazettáról nem lehet újra beolvasni.

13.7 Kazettahibák

A kazettakezelő a következő hibakódokat adhatja vissza a KERNEL-nél megismert általános hibákon kívül:

OECH, 236: File-vég hiba (EDF): a file vége után kiadott olvasáskor lép fel.

OEBH, 235: File-megnyitási kísérlet már megnyitott file esetén, tehát CLOSE nélkül újra kiadott OPEN vagy CRTE.

OEAH, 234: CRC hiba: olvasáskor a szektor után felírt ellenőrző szám nem egyezik a számítottal, a szalag tartalma sérült vagy rosszul olvasható.

OE9H, 233: Nincs nyitott file, ennek ellenére írási, olvasási vagy CLOSE parancsot adtunk ki.

OEBH, 232: Ellenőrzési hiba:

- Nem pufferelt file-nál a kazettán levő blokkhossz és az ellenőrzési hossz nem egyezik.
- Eltérés van a memóriatartalom és a szalagon levő adatok között (mindkét file-típusnál).

OE7H, 231: Belső hiba:

- Nem pufferelt file-nál több byte-ot akartunk kiolvasni a fejből, mint amennyi fel lett írva.
- Rossz byte-szám beolvasás.

0E6H, 230: A védelem megsértése:

- Védett file olvasásra nyitása után file-nyitási kísérlet írásra.
- Írási megnyitás után kísérlet védett file megnyitására olvasásra.

0E5H, 229: Hibás blokkszám:

- A beolvasott szektor sorszáma nem egyezik a folyamatosan növekvő sorszám alapján várt értékkel.
- Fejtömb jött a várt adattömb helyett.

13.8 Programozási példa

Egy gépi kódú programot írtunk, amely 2000H-tól 2FFFFH-ig tart. Hogyan lehet ezt a programot BASIC programként kimenteni?

Először is kell egy indító BASIC program:

```
10 X=USR(8192)
```

vagy

```
10 PRINT USR(8192)
```

(A 8192 a 2000H-nak felel meg.)

A BASIC programok mentése (SAVE) a memóriatartalom kiírását jelenti a 19EFH címtől a program végéig. Ezt kell leutánoznunk azzal az eltéréssel,

hogy a kimentés 2FFFFH-ig tartson. A részletes mentőprogram helyett csak a programozási lépéseket soroljuk föl:

1. Le kell tenni valahová a memóriába a program nevét hossz+NÉV alakban.
2. A PUFFER változóba írjunk 0-t (nem pufferelt file).
3. CAS_CRTE hívás, előtte DE-t állítsuk be úgy, hogy a névre mutasson.
4. A kiküldendő memóriatartomány hossza

$$3000H---19EFH = 1611H$$

5. 16 byte adat írása a fejtömbbe CAS_CHOUT-tal. A C regiszterbe írandó adatok:

- 00H
- 01H (programfile)
- 11H (hossz L része)
- 16H (hossz H része)
- 0 vagy 0FFH (utóbbi esetben autoindítás)
- 00H (tizenegyszer)

A 16 byte természetesen lehet a memóriába is és programciklussal is kiküldhető.

6. CAS_BKOUT hívás az alábbi paraméterekkel

$$DE = 19EFH$$

$$BC = 1611H$$

7. CAS_CLOSE hívás (kód: 54H)

Az elmondottak szerint egy kis gépi kódú programot írva és lefuttatva, az így kimentett program normál BASIC programként betölthető lesz. RUN-nal indítható, ill. autoindítás esetén a betöltés után rögtön elindul. Az egy-

soros BASIC program és a gépi kódú rész között nem kell feltétlenül 600H byte-ot hagyni, de valamekkora távolságra szükség van, nehogy a BASIC változók elrontsák a gépi kódú program elejét.

Ne feledjük el, hogy a 19EFH kezdőcím csak akkor igaz, ha a LOMEM-et, VLOMEM-et nem állítottuk át. Az ily módon kimentett programok csak akkor töltődnek a helyükre, ha a VLOMEM nincs átírva. Ezért gépi kódú játékprogramok betöltése előtt célszerű kettős RESET-et nyomni.

14. A SOROS VONAL

Bár a TV-Computerhez vásárolható soros vonal külső bővítőkártya, mégis itt kell vele foglalkoznunk: a kártyát működtető eszközmeghajtó szoftver nem a kártyára épített ROM-ban található, hanem az alapkészülék EXT ROM-jában.

14.1 Hardverfelépítés

A TV-Computer soros vonala egy 8251A típusú USART-ot tartalmaz, amelynek ki- és bemenetei V.24 adó-vevő áramkörökön keresztül kapcsolódnak a vonali csatlakozóhoz.

A V.24 interface jelekből a soros kártya a következőket kezeli (zárójelben a 25 pólusú csatlakozó kivezetés számai):

- 102 (7): Föld.
- 103 (2): Adási adatvezeték, az USART TxD kimenete.
- 104 (3): Vételi adatvezeték, az USART RxD bemenete. A 104-es jelet a 109-es jel kapuzza.
- 105 (4): Az USART /RTS kimenete, belső átkötéssel tartósan "BE" állapotba köthető. A TV-Computer az ellenállomás adását engedélyezi a 105-ös jellel.
- 106 (5): Az USART /CTS bemenete, adáshoz "BE" állapotba kell kötni.
- 107 (6): Az USART /DSR bemenete. A TV-Computer ezen a vezetéken érzékeli, hogy az ellenállomás a küldendő adatot fogadni tudja.
- 108 (20): "BE" állapotú jel, az USART /DTR kimenete is ráköthető. Nincs használva.
- 109 (8): A TV-Computernél a 104-es jelet kapuzza, ezért vétel csak akkor lehetséges, ha "BE" állapotban va.

111 (23): "BE" állapotra van kötve, átköthető "KI" állapotra is.

Modem nélküli összeköttetést feltételezve, két TV-Computer közötti összekötő kábel bekötése az alábbi:

```
102 (7) <---> 102 (7)
103 (2) ---> 104 (3)
104 (3) <--- 103 (2)
105 (4) ---> 107 (6)
107 (6) <--- 105 (4)
```

Mindkét csatlakozón belül össze kell kötni a 111, 109 és 106-os jeleket (23, 8, 5), ez biztosítja a 109-es és a 106-os jel "BE" állapotát.

Az előbbiekből kitűnik, hogy a modemkezelő jelek nem az eredeti célra vannak felhasználva, hanem handshake vezetékeknek. A handshake kézfogást jelent: az adó TV-Computer csak akkor küldheti az adatot, ha a vevő TV-Computer az /RST (105) vezetéket "BE" állapotba hozta, ezt az adó TV-Computer a /DSR (107) bemeneten érzékeli. Ezzel elérhető, hogy nem lép fel karaktervesztés nagyobb átviteli sebesség esetén akkor sem, ha a vett karakter feldolgozása hosszabb időt vesz igénybe. Természetesen a handshake csak akkor működik, ha erre az ellenállomás is fel van készítve. Ha az ellenállomás nem kezeli a handshake jeleket, az átviteli sebességet kell csökkenteni.

A soros vonali kártya USART-ja az alábbi I/O címeken érhető el.

X0H: adat írás/adat olvasás

X1H: parancs írás/állapot olvasás

(Az X értéke 1, 2, 3 vagy 4 lehet a 0, 1, 2, 3 csatlakozóhelynek megfelelően.)

Az I/O címek decimálisan:

$(n+1)*16$: adat írás/adat olvasás

$(n+1)*16+1$: parancs írás/állapot olvasás

(Az "n" a csatlakozóhely száma: 0, 1, 2, 3)

14.2 A 8251A típusú USART programozása

Az USART közvetlen programozására általában nincs szükség, az adatforgalmazás a soros vonallal a funkcióhívások segítségével lebonyolítható. Ennek ellenére célszerű összefoglalni a 8251A programozási tudnivalóit.

Az USART két I/O címet foglal le, az X0H-s címen keresztül az adatforgalmazás, az X1H-s címen keresztül a parancskiküldés, ill. az állapotolvasás bonyolítható.

Bekapcsoláskor vagy az ún. "belső RESET" parancs hatására az USART alapállapotba kerül, ilyenkor a parancscímre küldött byte-ot üzemmód-kiválasztó utasításnak értelmezi. Az üzemmód-kiválasztással érhető el, hogy az USART szinkron vagy aszinkron üzemmódban dolgozzon-e. Csak az aszinkron üzemmódot tekintjük át.

Az üzemmód-kiválasztó utasítás bitjei:

B7, B6 : a STOP bitek száma

00: nem megengedett

01: 1 bit

10: másfél bit

11: 2 bit

- B5: paritás
 0: páratlan
 1: páros
- B4: paritás engedélyezése
 0: adásnál nem küld paritásbitet, vételnél nincs ellenőrzés
 1: adásnál küld paritásbitet, vételnél ellenőrzés van
- B3, B2: karakterhossz
 00: 5 bit
 01: 6 bit
 10: 7 bit
 11: 8 bit
- B1, B0: üzemmód-kiválasztás, órajel
 00: szinkron üzemmód kijelölés lenne
 01: egyszeres órafrekvencia, nem megengedett
 10: 16-szoros órafrekvencia
 11: 64-szörös órafrekvencia

A TV-Computerben használt alapbeállítás DEEH (238):

2 STOP bit
nincs paritás kiküldés/paritás
8 bit
16-szoros óra

Ez azt jelenti, hogy a soros vonalon kiküldött adat 11 ütemből áll:

1 START bit (logikai "0")
8 adatbit (B0...B7)
2 STOP bit (logikai "1")

A "HANG" fejezetben elmondottak szerint előállított órajel frekvenciáját 16-tal kell osztani, tehát a "baud" érték:

$$97656.25/(4096\text{---PITCH})$$

Az üzemmód-kiválasztó utasítás kiadása után az X1H-s (parancs-) címre kiküldött byte-okat az USART parancsnak értelmezi, a parancsbyte bitjei külön-külön hatásosak:

B7: Csak szinkron üzemmódban értelmezett, ezért B7=0 legyen a kiküldött érték.

B6=1: Belső reset, az ezután küldött parancs üzemmódkiválasztó utasítás lesz.

B5=1: Aktív állapotba teszi az /RTS kimenetet, tehát a 105-ös jel "BE" állapotba kerül.

B4=1: Hibatörlés, a vétel során fellépő paritás-, túlfutás- és kerethibákat törli.

B3=1: Az ún. "BREAK" kiküldése a vonalra (a "BREAK" 1 karakternél hosszabb "START" állapotot jelent).

B2=1: A vétel engedélyezése.

B1=1: A /DTR kimenetet aktív állapotba teszi.

B0=1: Az adás engedélyezése.

A könyv megírása idején forgalmazott TV-Computerek (BASIC 1.2 verzió) soros vonalának inicializálásából hiányzik egy adás/vétel engedélyezés parancs, ezért ezt a TV-Computer bekapcsolása után egyetlenegyszer ki kell adni (a csatlakozószámtól függően):

OUT 17,5 (0-ás csatlakozó)
OUT 33,5 (1-es csatlakozó)
OUT 49,5 (2-es csatlakozó)
OUT 65,5 (3-as csatlakozó)

Az USART állapotáról az XIH-s cím olvasásával lehet tájékozódni:

B7=1: A /DSR (107-es) vezeték "BE" állapotban van.

B6=1: A vevő BREAK-et (tartós START) érzékelt.

B5=1: Kerethiba (FE, Framing Error), azt jelenti, hogy az adatbitek nem követte legalább 1 STOP bit.

B4=1: Vevő túlfutás (OE, Overrun Error), a vett karaktert nem olvasta be idejében a CPU, a vonalról jövő újabb karakter ezt felülírta.

B3=1: Paritáshiba (PE, Parity Error): paritásellenőrzés volt kijelölve, a vett karakter paritásbitje nem megfelelő.

B2=1: Az adópuffer üres, karakter küldhető.

B1=1: Az USART karaktert vett, a karakter kiolvasható.

B0=1: Az USART adásra kész.

A hardver tudnivalókat a 21. Függelék foglalja össze.

14.3 A soros vonalat működtető szoftver

A soros vonalat működtető szoftver 4 soros vonalat tud kezelni.

Az USART-ot inicializáló rutin két rendszerváltozó figyelembevételével dolgozik:

BAUD (0B69H, 2921):

A BAUD-ban tárolt érték szabja meg a soros vonal sebességét:

0 =	110 baud
1 =	150 "
2 =	300 "
3 =	600 "
4 =	1200 "
5 =	2400 baud
6 =	4800 "
7 =	9600 "
8 =	19200 "

Az alapbeállítás: 4 = 1200 baud

FORMAT (0B6AH, 2922):

Ebbe a változóba kell az üzemmód-kiválasztó utasításnál elmondottak szerint a karakterhosszt, a paritást és a STOP biteket meghatározó értékeket letenni.

Megjegyezzük, hogy a soros vonalat inicializáló rutin a FORMAT-ból csak a B2, B4, B5, B7 biteket veszi figyelembe. A FORMAT értékétől függetlenül

B0=0, B1=1, B3=1, B6=1 lesz az USART-ba írva. A FORMAT-ból figyelembe vett bitek jelentése:

- B2=1/0 : 8/7 adatbit
- B4=1/0 : van/nincs paritásképzés és -ellenőrzés
- B5=1/0 : páros/páratlan paritás
- B7=1/0 : 2/1 STOP bit

A FORMAT alapértéke: 0EEH, 238: 8 bit, 2 STOP, nincs paritás.

A BAUD és a FORMAT változók a "meleg" RESET-re is felveszik az alapértékeiket, ezek figyelembevételével történik meg az USART inicializálása. Ha valamit módosítani akarunk:

- Írjuk át a BAUD, ill. a FORMAT változót a kívánt értékre.
- Gépi kódú programból hívjuk meg az RS232_SET funkciót, vagy írjuk át a SER_OK változót (0B71H, 2929) 0FFH-ra POKE-kal. A POKE helyett egy SOUND utasítás kiadása is elég, a SOUND is 0FFH-ra írja a SER_OK változót.

A soros vonal input/output funkciói mindig megvizsgálják a SER_OK változót. Ha értéke nem 0, a tényleges I/O művelet előtt végrehajtódik az USART és a hanggenerátor újrainicializálása.

A soros vonal a karakteres és blokkos I/O funkciókon kívül üres CLOSE funkciót is tartalmaz abból a célból, hogy a SAVE#6:, ill. a LOAD#6: ne okozza nemlétező funkciók meghívását. A kazetta OPEN funkcióknak (3) az RS232_SET felel meg. A soros vonal funkcióhívásai alatt is felléphetnek a szokásos KERNEL hibák (pl. CTRL+ESC). A lehetséges funkcióhívások a következők:

14.3.1 SER_CHOUT

Karakterkiküldés

Kód: 61H (97)

Input: C = karakterkód

Output: -

A SER_CHOUT megvizsgálja a DSR bitet: ha az 1, akkor az ellenállomás kész fogadni a következő karaktert. Ezután teszteli, hogy az USART kész-e az adásra (BO bit). Ha igen, kiküldi a karaktert a vonalra.

14.3.2 SER_CHIN

Karakterbeolvasás

Kód: 0E1H (225)

Input: -

Output: C = a vett karakter kódja

A SER_CHIN megvizsgálja, hogy nincs-e vett karakter az USART vételi pufferekjében (B1=1 az állapotszóban). Ha van, akkor beolvassa az USART-ból és visszatér a hívóhoz.

Ha nincs vett karakter, akkor a /RTS kimenetet "BE" állapotba teszi, ezzel jelezve az ellenállomásnak, hogy küldheti a következő karaktert. Ezután a B1 bit állandó vizsgálatával figyeli, hogy az USART vett-e karaktert. Ha a karakter megérkezett, az /RTS-t kikapcsolja és a karakter beolvasása után visszatér a hívóhoz. A SER_CHIN vizsgálja az állapotszó beolvasásával, hogy nincs-e vételi hiba (FE, DE, PE).

Hiba esetén a megfelelő hibakódot adja vissza:

0F4H (244) : paritáshiba,

0F3H (243) : kerethiba (nincs STOP bit),

0F2H (242) : túlfutás (nem történt meg a kiolvasás a következő karakter vétele előtt).

A SER_CHIN várakozó állapota alatt a megszakítás le van tiltva, nehogy egy hosszabb megszakításprogram alatt karaktervesztés következzen be.

14.3.3 SER_BKOUT

Blokkkiküldés a soros vonalra

Kód: 62H (98)

Input: DE = a memóriapuffer kezdőcíme
BC = a kiküldendő byte-ok száma

Output: -

14.3.4 SER_BKIN

Blokkbeolvasás a soros vonalról

Kód: 0E2H (226)

Input: DE = a memóriapuffer kezdőcíme
BC = az olvasandó byte-ok száma

Output: -

14.3.5 RS232_SET

A soros vonal inicializálása

Kód: 063H vagy 0E3H (99 vagy 227)

Input: -

Output: -

Hibakód: -

Ez a funkció inicializálja az USART-ot a FORMAT rendszerváltozó alapján, és beállítja a hanggenerátort a BAUD változó szerint. A soros vonal input/output funkciói automatikusan meghívják, ha a SER_OK nem egyenlő 0-val (hangkeltés vagy kazettahívás elrontotta a hanggenerátor frekvenciáját).

14.3.6 RS232_CLOSE

Üres funkció, nem csinál semmit

Kód: 064H vagy 0E4H (100 vagy 228)

Input: -

Output: -

Hibakód: -

A soros vonali rendszerváltozók, rendszerhívások összefoglalása a 22. Függelékben található.

FÜGGELÉK

1. Memórialapozás

LAP0	0000	0	SYS (00)	U0 (10)	CART (08)	U3 (18)	**
	3FFF	16383					
LAP1	4000	16384	U1 (00)	VID (04)			*
	7FFF	32767					
LAP2	8000	32768	VID (00)	U2 (20)			
	BFFF	49151					
LAP3	C000	49152				IOMEM	
	DFFF	57343					
	E000	57344	CART (00)	SYS (40)	U3 (80)	----- (C0)	
	FFFF	65535				EXT	

PORT02H (2) írás

B7	B6	B5	B4	B3	B2	B1	B0
LAP3	LAP2	LAP0	LAP1	-	-		
00: CART (00)	0: VID (00)	00: SYS (00)	0: U1 (00)				
01: SYS (40)	1: U2 (20)	01: CART (08)	1: VID *(04)				
10: U3 (80)		10: U0 (10)					
11: EXT (C0)		11: U3 *(18)					
0003H 3 P_SAVE DEFS 1 ; Aktuális lapozási összetétel							

PORT03H (3) írás

B7	B6	B5	B4	B3	B2	B1	B0
IOMEM	-	-	+	+	+	+	+
00: CSTL0 (00)							
01: CSTL1 (40)							
10: CSTL2 (80)							
11: CSTL3 (C0)							
0B11H 2833 PORT03 DEFS 1 ; A 3-as I/O port tükörképe							

PORT0FH (15) írás *

B7	B6	B5	B4	B3	B2	B1	B0
-	-	CRT	LAP2	LAP1			
		00: VID0 (00)	00: VID0 (00)	00: VID0 (00)			
		01: VID1 (10)	01: VID1 (04)	01: VID1 (01)			
		10: VID2 (20)	10: VID2 (08)	10: VID2 (02)			
		11: VID3 (30)	11: VID3 (0C)	11: VID3 (03)			

0B19H 2841 HI MEM DEFS 2 ; A legnagyobb használható RAM cím
0B1BH 2843 P3RAM DEFS 1 ; 0: U3 jó; FF: U3 hibás

Megjegyzések:

* Csak TV-Computer 64 k+

** Csak TV-Computer 64 k+ és TV-Computer 32 k (32 k bővítéssel)

2. Megszakításkezelő hardver

PORT05H (5) írás

B7	B6	B5	B4	B3	B2	B1	B0
+	+	Hang IT	+	+	+	+	+
		0: tilt					
		1: eng					
0B12H 2834 PORT05 DEFS 1 ; Az 5-ös I/O port tükörképe							

Képernyővezérlő R10 írás

B7	B6	B5	B4	B3	B2	B1	B0
0	00: cursor eng.	0	0	0	1	1	
	01: cursor tilt	Cursor kezdősor a karaktersorban					

PORT07H (7) írás

B7	B6	B5	B4	B3	B2	B1	B0
-	-	-	-	-	-	-	-
Cursor/hang IT törlés							

PORT58H (88) írás: CSTL0

PORT59H (89) írás: CSTL1

PORT5AH (90) írás: CSTL2

PORT5BH (91) írás: CSTL3

B7	B6	B5	B4	B3	B2	B1	B0
CSTL IT	-	-	-	-	-	-	-
0: törl							
tilt							
1: eng							

PORT59H (89) olvasás

B7	B6	B5	B4	B3	B2	B1	B0
+	+	+	Függően lévő IT kérések (0: aktiv)				
			CU/HANG	CSTL3	CSTL2	CSTL1	CSTL0

PORT5BH (91) olvasás

B7	B6	B5	B4	B3	B2	B1	B0
-	-	-	-	-	-	-	-
Hanggenerátor frekvenciaosztó törlése (pontos időzítéshez)							

Megjegyzés: az IT a megszakítás szó angol megfelelőjének (interrupt) rövidítése.

3. Megszakításkezelő szoftver

```

0038 0056      IT_ENTRY      ; RST 38H
;
0038 0056  F5      PUSH  AF
0039 0057  3E 70    LD     A,70H      ; U0, U1, U2, SYS lapozas
003B 0059  D3 02    OUT    (2),A
003D 0061  C3 12 C4  JP     0C412H    ; SYSTEM ROM
;                                     Nem garantált cím
;
;
;
0B10 2832  FC  INT_DES  DEFS  1      ; A cursor/hang IT által
;                                     kiszolgált eszközök
;                                     B0=0: VIDEO
;                                     B1=0: BILLENTYÜZET
;                                     B2=0: EDITOR
;                                     B3=0: HANG
;                                     B4=0: CSTL0
;                                     B5=0: CSTL1
;                                     B6=0: CSTL2
;                                     B7=0: CSTL3
;
0B1D 2845      INTINC  DEFS  2      ; 16 bit számláló (OS IT)
;                                     (20.096 ms)
;
0B1F 2847  01  IRQ_STAT  DEFS  1      ; IT engedélyezés az OS
;                                     IT kiszolgálása után
;                                     B0=1: CURSOR
;                                     B1=1: HANG
;                                     B2=1: CSTL0
;                                     B3=1: CSTL1
;                                     B4=1: CSTL2
;                                     B5=1: CSTL3
;
0B20 2848      INT_FLAG  DEFS  1      ; FF: IT kiszolgálás
;                                     alatt
;
;
;
0B41 2881      INT_EXIT      ; Kilépés az IT-ből
;
0B41 2881  32 03 00    LD     A,(3)    ; Felhasználói mem.
0B44 2884  D3 02    OUT    (3),A      ; lapozás visszaállítás
0B46 2886  F1      POP    AF
0B47 2887  FB      EI
0B48 2888  C9      RET
;
;

```

4. Csatolóókártyák

A csatolókartvák I/O címtartománya:

```
0. csatlakozó: 10H ... 1FH (16 ... 31)
1. csatlakozó: 20H ... 2FH (32 ... 47)
2. csatlakozó: 30H ... 3FH (48 ... 63)
3. csatlakozó: 40H ... 4FH (64 ... 79)
```

PORT03H (3) irás

B7	B6	B5	B4	B3	B2	B1	B0
IOMEM lapozás		-	-	+	+	+	+
00: CSTL0 (00)							
01: CSTL1 (40)							
10: CSTL2 (80)							
11: CSTL3 (C0)							
0B11H 2833 PORT03 DEFS 1 ; A 3-as I/O port tükörképe							

```
PORT58H (88) iràs: CSTL0
PORT59H (89) iràs: CSTL1
PORT5AH (90) iràs: CSTL2
PORT5BH (91) iràs: CSTL3
```

	B7	B6	B5	B4	B3	B2	B1	B0
CSTL IT	-	-	-	-	-	-	-	-
0: töröl								
tilt								
1: eng								

PORT59H (89) olvasás

B7	B6	B5	B4	B3	B2	B1	B0
+	+	+	Függőben lévő IT kérések (0: aktiv)				
			CU/HANG!	CSTL3	CSTL2	CSTL1	CSTL0

PORT5AH (90) olvasás

B7	B6	B5	B4	B3	B2	B1	B0
3. csatlakozó		2. csatlakozó		1. csatlakozó		0. csatlakozó	
..... kártyaazonosító							
00: soros vonal (RS232)				01: nem használt			
10: floppy csatló				11: üres csatl. v. nem specifikált			

5. Input/output rendszer

```

0030 0048 OS_ENTRY ; RST 30H
;
0030 0048 C3 23 0B JP INFUNC
;
;
0040 0064 DEFS 8 ; 0. csatoló azonosító
; 0. byte: névhossz
; 1-6. byte: név
; 7. byte: egységszám
; azonos típusú
; csatolóknál
0048 0072 DEFS 40 ; 0. csatoló RAM terület
;
0070 0112 DEFS 8 ; 1. csatoló azonosító
0078 0120 DEFS 40 ; 1. csatoló RAM terület
;
00A0 0160 DEFS 8 ; 2. csatoló azonosító
00A8 0168 DEFS 40 ; 2. csatoló RAM terület
;
00D0 0208 DEFS 8 ; 3. csatoló azonosító
00D8 0216 DEFS 40 ; 3. csatoló RAM terület
;
0B00 2816 IN_TABLE DEFS 8 ; Input hozzárendelési
; táblázat (l. 6. Függelék)
0B08 2824 OUT_TABLE DEFS 8 ; Output hozzárendelési
; táblázat (l. 6. Függelék)
;
0B1C 2844 EX_DEF DEFS 1 ; Bővítő kártya alaphozzá-
; rendelés (RS232, ha van)
;
;
0B23 2851 INFUNC ; OS hívás RAM része
;
0B23 2851 E3 EX (SP),HL ; A verem tartalma a funkció-
0B24 2852 7E LD A,(HL) ; kódra mutat, f.kód olvasás
0B25 2853 23 INC HL
0B26 2854 E3 EX (SP),HL ; Visszatérési cím -> verem
0B27 2855 08 EX AF,AF'
0B28 2856 F5 PUSH AF ; AF' mentése
0B29 2857 3A 03 00 LD A,(3) ; P_SAVE
0B2C 2860 F5 PUSH AF ; P_SAVE mentése
0B2D 2861 3E 70 LD A,70H ; U0,U1,U2,SYS lapozás
0B2F 2863 32 03 00 LD (3),A
0B32 2866 D3 02 OUT (2),A
0B34 2868 C3 63 C3 JP 0C363H ; SYS ROM, nem garantált cím
;
0B37 2871 OUTFUNC ; OS hívás befejező rész
;
0B37 2871 08 EX AF,AF' ; AF: hibakód
0B38 2872 F1 POP AF ; P_SAVE és felhasználói
; lapozás visszaállítás
0B39 2873 32 03 00 LD (3),A
0B3C 2876 D3 02 OUT (2),A
0B3E 2878 F1 POP AF ; AF' visszaállítás
0B3F 2879 08 EX AF,AF' ; AF: hibakód
0B40 2880 C9 RET
;

```

6. Input/output hozzárendelés

Funkció osztály	Memóriacím	Eszköz
I n p u t t á b l á z a t		
0. Video input	0B00 2816	FF: Nem értelmezett
1. Billentyűzet input	0B01 2817	01: Billentyűzet
2. Editor input	0B02 2818	02: Editor
3. Hang input	0B03 2819	FF: Nem értelmezett
4. Nyomtató input	0B04 2820	FF: Nem értelmezett
5. Kazetta input	0B05 2821	05: Kazetta
6. Csatolókartya input	0B06 2822	06: Csatolókartya
Csatolókartya input kijelölés	0B07 2823	FF: Nincs kijelölés 0X: X. csatoló (0..3)
O u t p u t t á b l á z a t		
0. Video output	0B08 2824	00: Video
1. Billentyűzet output	0B09 2825	FF: Nem értelmezett
2. Editor output	0B0A 2826	02: Editor
3. Hang output	0B0B 2827	FF: Nem értelmezett
4. Nyomtató output	0B0C 2828	04: Nyomtató
5. Kazetta output	0B0D 2829	05: Kazetta
6. Csatolókartya output	0B0E 2830	06: Csatolókartya
Csatolókartya output kijelölés	0B0F 2831	FF: Nincs kijelölés 0X: X. csatoló (0..3)

Példák:

1. Az editor hozzárendelése a nyomtató osztályhoz (az LPRINT, LLIST BASIC utasítás a képernyőre fog dolgozni):

0B0C (2828) nyomtató output: 02H (editor)

2. Az 1. interface csatlakozóban soros vonal, RS232 van, a nyomtató funkciók a soros vonalhoz kétféleképpen rendelhetők:

2.1. 0B0C (2828) nyomtató output: 06H (cs. kártya)
0B0F (2831) cs. kártya output kijelölés: 01H (RS 232)

2.2. 0B0C (2828) nyomtató output: 81H (01+80H, 129)

7. Kernel funkciók

Funkció	Kód hex dec	Input	Output	Hiba hex.
HI_MEM_SET memória foglalás	71 113 F1 241	DE= lefoglalandó mem. méret	DE= első lefoglalt cim	FB
EXP_DIRECT csat. kártya kijelölés	72 114 F2 242	DE= azonosítócím B= FF: input kij. C= egység szám		FD
I/O_DIRECT I/O hozzá- rendelés	73 115 F3 243	D= FF: input h.r. B= osztály C= eszköz		FE
SLOT_NUM csat. kártya pozíció olv.	74 116 F4 244	DE= azonosítócím C= egység szám		FD

Hibakódok:

OFFH (255) : nem létező hívási kód
 OFEH (254) : I/O hozzárendelési hiba
 OFDH (253) : nem létező kártya
 OFBH (251) : nincs elég memória

Blokk I/O hibák:

OFCH (252) : hibás paraméterezés (LAP3 kijelölése)
 OFAH (250) : hibás paraméterezés (HI_MEM átlépése)

Billentőzet hiba:

OF5H (245) : CTRL+ESC-et (STOP-ot) nyomtak

Megjegyzés:

A HI_MEM átlépést csak az 1.2 verziójú rendszer ellenőrzi.

8. Videomemória

tv sor	"Karakter"										"Kar." sor
	0	1	2	3	-	-	-	61	62	63	
0	8000	8001	8002	8003	-	-	-	803D	803E	803F	0
1	8040	8041	8042	8043	-	-	-	807D	807E	807F	
2	8080	8081	8082	8083	-	-	-	80BD	80BE	80BF	
3	80C0	80C1	80C2	80C3	-	-	-	80FD	80FE	80FF	
4	8100	8101	8102	8103	-	-	-	813D	813E	813F	1
5	8140	8141	8142	8143	-	-	-	817D	817E	817F	
6	8180	8181	8182	8183	-	-	-	81BD	81BE	81BF	
7	81C0	81C1	81C2	81C3	-	-	-	81FD	81FE	81FF	
8	8200	8201	8202	8203	-	-	-	823D	823E	823F	2
9	8240	8241	8242	8243	-	-	-	827D	827E	827F	
10	8280	8281	8282	8283	-	-	-	82BD	82BE	82BF	
11	82C0	82C1	82C2	82C3	-	-	-	82FD	82FE	82FF	
232	BA00	BA01	BA02	BA03	-	-	-	BA3D	BA3E	BA3F	58
233	BA40	BA41	BA42	BA43	-	-	-	BA7D	BA7E	BA7F	
234	BA80	BA81	BA82	BA83	-	-	-	BABD	BABE	BABF	
235	BAC0	BAC1	BAC2	BAC3	-	-	-	BAFD	BAFE	BAFF	
236	BB00	BB01	BB02	BB03	-	-	-	BB3D	BB3E	BB3F	59
237	BB40	BB41	BB42	BB43	-	-	-	BB7D	BB7E	BB7F	
238	BB80	BB81	BB82	BB83	-	-	-	BBBD	BBBE	BBBF	
239	BBC0	BBC1	BBC2	BBC3	-	-	-	BBFD	BBFE	BBFF	

GRAPHICS 2 (8 pont/byte, pontsorrend: 1 2 3 4 5 6 7 8)

B7	B6	B5	B4	B3	B2	B1	B0
1.p. C0	2.p. C0	3.p. C0	4.p. C0	5.p. C0	6.p. C0	7.p. C0	8.p. C0
..... C1=0							

GRAPHICS 4 (4 pont/byte, pontsorrend: 1 2 3 4)

B7	B6	B5	B4	B3	B2	B1	B0
1.p. C0	2.p. C0	3.p. C0	4.p. C0	1.p. C1	2.p. C1	3.p. C1	4.p. C1

GRAPHICS 16 (2 pont/byte, pontsorrend: 1 2)

B7	B6	B5	B4	B3	B2	B1	B0
1.p. I	2.p. I	1.p. G	2.p. G	1.p. R	2.p. R	1.p. B	2.p. B

9. Video hardver

PORT00H (0) írás: borderszín

B7	B6	B5	B4	B3	B2	B1	B0
I	-	G	-	R	-	B	-
0B4FH	2895	BORDER	DEFS	1	; Borderszín		

PORT06H (6) írás: grafikus mód

B7	B6	B5	B4	B3	B2	B1	B0
+	-	+	+	+	+	00 : 2 szín 01 : 4 szín 1X : 16 szín	
0B13H	2835	PORT06	DEFS	1	; A 6-os I/O port tükörképe		

PORT59H (89) olvasás: szinkikapcsoló

B7	B6	B5	B4	B3	B2	B1	B0
+	0:F/F 1:szín.	+	+	+	+	+	+

PORT60H (96) írás: 0. paletta
 PORT61H (97) írás: 1. paletta
 PORT62H (98) írás: 2. paletta
 PORT63H (99) írás: 3. paletta

B7	B6	B5	B4	B3	B2	B1	B0
-	I	-	G	-	R	-	B

Színek

I G R B	Szín	GRAPHICS 16 tinta papír	GR. 2, 4 paletta		Border	
			hex.	dec.	hex.	dec.
0 0 0 0	Fekete	0	00	0	00	0
0 0 0 1	Sötétkék	1	01	1	02	2
0 0 1 0	Sötétvörös	2	04	4	08	8
0 0 1 1	Sötétlila	3	05	5	0A	10
0 1 0 0	Sötétzöld	4	10	16	20	32
0 1 0 1	Sötét kékeszöld	5	11	17	22	34
0 1 1 0	Sötétsárga	6	14	20	28	40
0 1 1 1	Szürke	7	15	21	2A	42
1 0 0 0	Fekete	8	40	64	80	128
1 0 0 1	Kék	9	41	65	82	130
1 0 1 0	Vörös	10	44	68	88	136
1 0 1 1	Lila	11	45	69	8A	138
1 1 0 0	Zöld	12	50	80	A0	160
1 1 0 1	Kékeszöld	13	51	81	A2	162
1 1 1 0	Sárga	14	54	84	AB	168
1 1 1 1	Fehér	15	55	85	AA	170

10. A képernyővezérlő regiszterei

Reg.	R/W	Egység	B7	B6	B5	B4	B3	B2	B1	B0	Hex.	Dec.	
R0	W	kar.	(Összes karakter)/sor - 1								63	99	
R1	W	kar.	(Megjelenített karakter)/sor								40	64	
R2	W	kar.	Vízszintes szinkron pozíció - 1								4B	75	
R3	W	tv sor kar.	*Függ. szinkron szélesség (tv sor)				Vissz. szinkron szélesség (kar.)				32	50	
R4	W	k. sor	-	(Összes karaktersor)/kép - 1								4D	77
R5	W	tv sor	-	-	-	Kiegészítő tv sorszám					02	2	
R6	W	k. sor	-	(Megjelenített karaktersor)/kép								3C	60
R7	W	k. sor	-	Függőleges szinkron pozíció - 1								42	66
R8	W		* Cursor eltolás		* DISPTMG eltolás		-	-	Váltott sor X0: nincs 01: van		00	0	
R9	W	tv sor	-	-	-	(tv sor)/(kar. sor) - 1					03	3	
R10	W	tv sor	-	Cursor 00: eng. 01: tilt.		Cursor kezdő tv sor a karaktersorban					03	3	
R11	W	tv sor	-	-	-	Cursor befejező tv sor					03	3	
R12	R/W		0	0	Képkezdő memóriacím (H)					00	0		
R13	R/W				Képkezdő memóriacím (L)					00	0		
R14	R/W		0	0	Cursorpozíció (H)					0E	14		
R15	R/W				Cursorpozíció (L)					FF	255		
R16	R		0	0	Fényceruzapozíció (H)								
R17	R				Fényceruzapozíció (L)								

Szállítótól és típusváltozattól függően

- a * jelű funkciók hiányozhatnak
- R12 és R13 csak írható (W)

PORT70H (112) iraq

B7	B6	B5	B4	B3	B2	B1	B0
-	-	-	Képernyővezérlő regiszter cím				

PORT71H (113) irās/olvasās

B7	B6	B5	B4	B3	B2	B1	B0
Képernyővezérlő				regiszter		adata	

11. Video rendszerváltozók

```

0740 1856 RAM_FONT DEFS 960 ; A 128-223 kódú karakterek pontmátrixa
0B13 2835 PORT06 DEFS 1 ; A 6-os port tükörképe
; B1,B0: grafikus mód
; 0: 2 szín
; 1: 4 szín
; 2: 16 szín
; 3: 16 szín
0B4B 2891 L_MODE DEFS 1 ; Vonalkeresztelési mód
; 0: felülírás
; 1: OR
; 2: AND
; 3: XOR
0B4C 2892 L_STYLE DEFS 1 ; Vonaltípus
; 0: (-----)
; 1: (-----)
; 2: (- - - -)
; 3: (--- --)
; 4: (--- --)
; 5: (- - -)
; 6: (--- --)
; 7: (--- --)
; 8: (--- --)
; 9: (--- --)
; 10: (-----)
; 11: (-----)
; 12: (-----)
; 13: (-----)
; 14: (--- --)
; 15: (-----)
; 16: (-----)
0B4D 2893 INK DEFS 1 ; Aktuális tintaszín
; GR 2, GR 4: paletta
; sorszám
; GR 16: színsorszám
0B4E 2894 PAPER DEFS 1 ; Aktuális papírszín
; GR 2, GR 4: paletta
; sorszám
; GR 16: színsorszám
0B4F 2895 BORDER DEFS 1 ; Képkeret szín: PORT00
; írása IT alatt
; B7: I
; B5: G
; B3: R
; B1: B
0B50 2896 V_FLAG DEFS 1 ; Karakter felülírás flag
; 0: teljes felülírás
; 1: "színtelen" tinta
; 2: "átlátszó" papír
; 3: nincs felülírás

```

12. Video funkciók

Funkció	Kód hex dec		Input	Output	Hiba hex.
VID CHOUT karakter- kiírás	01	1	C= karakterkód		-
VID BKOUT szövegblokk- kiírás	02	2	DE= kezdőcím BC= hossz		
BTEXT sugar poz. kar. poz.-ba	03	3	B= oszlop (1-16, 1-32,1-64) C= sor (1-24)		F9
VMODE grafikus mód beállítás	04	4	C= 0 : 2 szín 1 : 4 szín 2 : 16 szín		F7
CLS képernyő- törles	05	5			-
BABS sugar mozg. absz. koord.	06	6	DE= Y koordináta BC= X koordináta		F9
BREL sugar mozg. rel. koord.	07	7	DE= Y elmozdulás BC= X elmozdulás		F9
BON sugar bekapcsolás	08				-
BOFF sugar kikapcsolás	09	9			-
FILL képkitöltés	0A	10			-
DEFC karakter- definíálás	0B	11	DE= kar. mátrix 10 byte címe C= karakterkód		F8
PAL színpaletta- -beállítás	0C	12	DE= színpaletta 4 byte címe		-

Hibakódok:

0F9H (249) : nem létező képpontpozíció
 0F8H (248) : nem definiálható karakter
 0F7H (247) : nem létező grafikus mód (színmód)

13. Billentyűzet

PORT03H (3) írás

B7	B6	B5	B4	B3	B2	B1	B0
+	+	-	-	Billentyűzetsor kiválasztása			
0B11H	2833	PORT03	DEFS	1	: A 3-as I/O port tükörképe		

PORT58H (88) olvasás

B7	B6	B5	B4	B3	B2	B1	B0
Billentyűzetadat (oszlop) beolvasása: aktiv 0							

Billentyűzet mátrix

Sor	PICTURE hex. dec.	B7	B6	B5	B4	B3	B2	B1	B0
0	0B51 2897	!	'		/	&	"	+	%
		4	1	1	6	0	2	3	5
1	0B52 2898	=	ö	o	#	ü	)	(	~
		7			*		9	8	^
2	0B53 2899	R	Q	@	Z	\$	W	E	T
3	0B54 2900	U	P	U	{	ö	o	I	}
					[				]
4	0B55 2901	F	A	>	H	\	S	D	G
				<					
5	0B56 2902	J	E	ü	RET	A	L	K	DEL
6	0B57 2903	V	Y	LOCK	N	SHIFT	X	C	B
7	0B58 2904	M	=	SPACE	CTRL	ESC	:	?	ALT
							.	,	
8	0B59 2905		LEFT	RIGHT	ACC	FIRE	DOWN	UP	INS
9	0B5A 2906		LEFT	RIGHT	ACC	FIRE	DOWN	UP	

Megjegyzések:

1. A 8-as sor a beépített botkormány és az előlső botkormány-csatlakozó. (A "Kezelési útmutató"-ban a 13-as.)
2. A 9-es sor a hátsó botkormánycsatlakozó. (A "Kezelési útmutató"-ban a 12-es.)
3. Az ALT, SHIFT, CTRL és LOCK nem állít be bitet a PICTURE-ben.

14. Billentyűzet kódok

Ny. gomb	Alap	SHIFT	CTRL	ALT	Ny. gomb	Alap	SHFT	CTRL	ALT
A	61	41	01	B1	0 &	30	26	89	A0
B	62	42	02	B2	1 '	31	27	99	A1
C	63	43	03	B3	2 "	32	22	8A	A2
D	64	44	04	B4	3 +	33	2B	9A	A3
E	65	45	05	B5	4 !	34	21	8B	A4
F	66	46	06	B6	5 %	35	25	9B	A5
G	67	47	07	B7	6 /	36	2F	8C	A6
H	68	48	08	B8	7 =	37	3D	9C	A7
I	69	49	09	B9	8 (	38	28	8D	A8
J	6A	4A	0A	BA	9)	39	29	9D	A9
K	6B	4B	0B	BB	@ ' ,	40	60	00	B0
L	6C	4C	0C	BC	* #	2A	23	8E	AA
M	6D	4D	0D	BD	^ ~	5E	7E	1E	CE
N	6E	4E	0E	BE	; \$	3B	24	3B	AB
O	6F	4F	0F	BF	< >	3C	3E	3C	AC
P (2)	70	50	10	C0	\	5C	7C	1C	CC
Q	71	51	11	C1	[{	5B	7B	1B	CB
R	72	52	12	C2	] }	5D	7D	1D	CD
S	73	53	13	C3	, ?	2C	3F	2C	AF
T	74	54	14	C4	. :	2E	3A	2E	AE
U	75	55	15	C5	- _	2D	5F	1F	AD
V	76	56	16	C6					
W	77	57	17	C7					
X	78	58	18	C8	ESC (3)	1B	1B	1B	1B
Y	79	59	19	C9	RETURN	0D	0D	0D	0D
Z	7A	5A	1A	CA	DEL	08	07	08	08
					INS	16	16	16	16
					SPACE	20	20	20	20
A	90	80	CF	D0					
E	91	81	D9	D1					
I	92	82	DA	D2	Belső				
O	93	83	DB	D3	botk.				
Ö	94	84	DC	D4					
Ö	95	85	DD	D5	UP		05		
U	96	86	DE	D6	DOWN		18		
U	97	87	DF	D7	RIGHT		04		
Ö	98	88	98	D8	LEFT		13		
Elülső					Hátsó				
csatl.					csatl.				
UP		05			UP		E5		
DOWN		18			DOWN		F8		
RIGHT		04			RIGHT		E4		
LEFT		13			LEFT		F3		
FIRE		06			FIRE		E6		
ACC		01			ACC		E1		

Megjegyzések:

1. A kódok hexadecimálisan vannak megadva.
2. A CTRL+P leállítja a TV-Computert, ha a HOLD_DIS változó =0.
3. A CTRL+ESC beállítja a STOP_FLAG változót 0FFH-ra.
4. A SHIFT, ALT, CTRL és LOCK önmagában nem generál kódot.
5. A botkormány tartós működtetése által kiváltott auto-repeat váltakozva adja az egyidejűleg működtetett irányokhoz tartozó kódokat.

15. Billentyűzetkezelő szoftver

Rendszerváltozók

```

:
OB16  2838  STOP_FLAG  DEFS  1      ; FF: STOP-ot nyomtak
:                                00: nem volt STOP
:
OB51  2897  PICTURE   DEFS  10     ; A billentyűzetmátrix lekér-
:                                dezése (l. 13. Függelék)
:
OB5B  2907  OLDPICT   DEFS  10     ; A billentyűzetmátrix előző
:                                lekérdezése
:
OB65  2917  DELAY_KEY  DEFS  1      ; Auto-repeat késleltetés
:                                20 ms-os egységekben
:
OB66  2918  LOCK_KEY   DEFS  1      ; CAPS/SHIFT/ALT LOCK állapot
:                                B0=1: CAPS LOCK
:                                B1=1: SHIFT LOCK
:                                B3=1: ALT LOCK
:
OB67  2919  RATE_KEY   DEFS  1      ; Auto-repeat ütem
:                                20 ms-os egységekben
:
OB68  2920  HOLD_DIS   DEFS  1      ; HOLD mód tiltás
:                                FF: CTRL+P hatástalan
:                                00: CTRL+P leállítja a
:                                TV-Computert
:

```

Funkciók

Funkció	Kód hex dec	Input	Output	Hiba hex.
KBD_CHIN karakter beolvasás	91 145		C= karakterkód	F5
KBD_STATUS bill. állapot beolvasás	93 147		C= FF: karakter vár beol- vasásra	-

Hibakód:

0F5H (245) : CTRL+ESC-et (STOP-ot) nyomtak

16. Editor funkciók

Funkció	Kód hex dec	Input	Output	Hiba hex
ED_INT IT rutin (cursor vil- logtatás)	20 32 A0 160			
ED_CHIN karakter beolvasás	A1 161		C= karakterkód	
ED_CHOUT karakter kiírás	21 33	C= karakterkód		
ED_BKIN szövegblokk beolvasás	A2 162	DE= kezdőcím BC= hossz		
ED_BKOUT szövegblokk kiírás	22 162	DE= kezdőcím BC= hossz		
CPOS cursor pozicionálás	23 35 A3 163	B= oszlop (1-16, 1-32, 1-64) C= sor (1-24)		F6
CFIX cursor poz. megjegyzés	24 36 A4 164			

Hibakód:

0F6H (246) : nem létező cursorpozíció

Szerkesztési parancsok:

```

RIGHT, CTRL+D (04H, 4): cursor jobbra
LEFT,  CTRL+S (13H, 19): cursor balra
UP,    CTRL+E (05H, 5): cursor föl
DOWN,  CTRL+X (18H, 24): cursor le
        CTRL+I (09H, 9): tabulálás
        CTRL+K (0BH, 11): sormaradéktörölés
        CTRL+N (0EH, 14): sorbeszűrés
        CTRL+Y (19H, 25): sorkiejtés
INS     (16H, 22): karakterbeszűrés
DEL     (08H, 8): karakterkijetés a cursortól balra
SHIFT+DEL (07H, 7): karakterkijetés a cursor alatt

```

Csak input funkcióhívásoknál:

```

ESC      (1BH, 27): a begépelés megszakítása
RETURN   (0DH, 13): a begépelés lezárása

```

Csak output funkcióhívásoknál:

```

RETURN   (0DH, 13): cursormozgatás a sor elejére
CTRL+J   (0AH, 10): cursor le, a kép alján rollozás

```

17. Hangkeltés

PORT04H (4) írás: PITCH alsó 8 bit

PORT05H (5) írás

B7	B6	B5	B4	B3	B2	B1	B0
+	+	Hang IT	Hangjel	PITCH felső 4 bit			
		0: tilt	0: tilt				
		1: eng	1: eng				
0B12H	2834	PORT05	DEFS	1	; Az 5-ös I/O port tükörképe		

PORT06H (6) írás

B7	B6	B5	B4	B3	B2	B1	B0
+	-	Hang amplitúdó				+	+
0B13H	2835	PORT06	DEFS	1	; A 6-os I/O port tükörképe		

PORT5BH (91) olvasás

B7	B6	B5	B4	B3	B2	B1	B0
-	-	-	-	-	-	-	-
Hanggenerátor frekvenciaosztó törlés (pontos időzítéshez)							

Rendszerváltozók

0B14	2836	SND_ACTIVE	DEFS	1	; FF: az előző hang nem fejeződött be
0B15	2837	SND_IRQ	DEFS	1	; FF: TONE_SET megszakítja a folyamatban lévő hangot
0B71	2929	SER_OK	DEFS	1	; FF: a hangkeltés elrontotta a soros vonal órajelet

Funkcióhívás

Funkció	Kód [hex dec]	Input	Output	Hiba [hex.]
TONE_SET hang megszo- laltatás	33 51	DE= PITCH érték C= ampl. (0-15) B= időtartam (20 ms egys.)		

$f = 195312.5 / (4096 - \text{PITCH})$

PITCH=0 ... 4095 (4095: nincs hang)

18. Nyomtatókezelés

PORT01H (1) írás: nyomtató adat

PORT06H (6) írás

B7	B6	B5	B4	B3	B2	B1	B0
Nyomtató /STROBE	-	+	+	+	+	+	+
0B13H 2835 PORT06 DEFS 1 ; A 6-os I/O port tükörképe							

PORT59H (B9) olvasás

B7	B6	B5	B4	B3	B2	B1	B0
Nyomtató ACK	+	+	+	+	+	+	+

Nyomtató funkciók

Funkció	Kód hex dec		Input	Output	Hiba hex.
PAR_CHOUT karakter kiírás	41	65	C= karakterkód		
PAR_BKOUT szövegblokk kiírás	42	66	DE= kezdőcím BC= hossz		

19. Kazettakezelés

PORT05H (5) írás: motorvezérlés

B7	B6	B5	B4	B3	B2	B1	B0
Jobb- oldali csatl.	Bal- oldali csatl.	+	+	+	+	+	+
0B12H 2834 PORT05 DEFS 1 ; Az 5-ös I/O port tükörképe							

PORT50H (80) írás: magnó kimenet

B7	B6	B5	B4	B3	B2	B1	B0
-	-	-	-	-	-	-	-

PORT59H (89) olvasás

B7	B6	B5	B4	B3	B2	B1	B0
+	+	Magnó adat	+	+	+	+	+

Rendszerváltozók

```

;
0B6B 2923 BUFFER      DEFS 1 ; =0: nem pufferelt file
;                               <>0: pufferelt file
;
0B6C 2924 REMRED      DEFS 1 ; Motorvezérlés:
;                               00H, 0: olvasás: B, írás: J
;                               40H, 64: olvasás: J, írás: J
;                               80H, 128: olvasás: B, írás: B
;                               0C0H, 192: olvasás: J, írás: B
;
0B6D 2925 PROTECT     DEFS 1 ; =0: nem írásvédett file
;                               <>0: írásvédett file
;
0B6E 2926 EOF         DEFS 1 ; <>0: file-vég, nincs több byte
;
0B6F 2927 MUDDLE      DEFS 2 ; A CRC számítás kiinduló értéke
;

```

20. Kazettafunkciók

Funkció	Kód hex dec	Input	Output	Hiba hex.
CAS_CHIN byte- beolvasás	D1 209		C= a beolvasott byte	
CAS_CHOUT byte- kiírás	51 81	C= a kiküldendő byte		
CAS_BKIN blokk- beolvasás	D2 210	DE= kezdőcím BC= hossz		
CAS_BKOUT blokk- kiírás	52 82	DE= kezdőcím BC= hossz		
CAS_OPEN megnyitás olvasásra	D3 211	DE= a file-név kezdőcíme (hossz+NÉV)	DE= a módosított (megtalált) file-név címe	
CAS_CRTE megnyitás írásra	53 83	DE= a file-név kezdőcíme (hossz+NÉV)	DE= a módosított file-név címe	
CAS_CLOSE olv. file: irt file:	D4 212 54 84			
CAS_VERIFY blokk komparálás	D5 213	DE= kezdőcím BC= hossz		

Hibakódok:

OECH (236) : file-vég jelzés
 OEBH (235) : nyitott file esetén újabb file nyitás kísérlete
 OEAH (234) : CRC hiba
 OE9H (233) : nincs nyitott file
 OE8H (232) : ellenőrzési (komparálási) hiba
 OE7H (231) : belső hiba (túl sok karaktert akartunk kiolvasni)
 OE6H (230) : a védelem megsértése
 OE5H (229) : hibás blokk sorszám (szektorszám)

21. Soros vonal

Adatkiküldés (USART):

PORT10H (16) írása (0. csatlakozó)
 PORT20H (32) írása (1. csatlakozó)
 PORT30H (48) írása (2. csatlakozó)
 PORT40H (64) írása (3. csatlakozó)

USART üzemmód-kiválasztó utasítás a "belső reset" parancs után:

PORT11H (17) írása (0. csatlakozó)
 PORT21H (33) írása (1. csatlakozó)
 PORT31H (49) írása (2. csatlakozó)
 PORT41H (65) írása (3. csatlakozó)

B7	B6	B5	B4	B3	B2	B1	B0
STOP bitek sz.		Paritás		Karakterhossz		Órafrekvencia	
00: -		X0: nincs par.		00: 5 bit		00: -	
01: 1	STOP			01: 6 bit		01: 1x (tilos)	
10: 1.5	STOP	01: páratlan		10: 7 bit		10: 16x	
11: 2	STOP	11: páros		11: 8 bit		11: 64x	

USART parancsok:

PORT11H (17) írása (0. csatlakozó)
 PORT21H (33) írása (1. csatlakozó)
 PORT31H (49) írása (2. csatlakozó)
 PORT41H (65) írása (3. csatlakozó)

B7	B6	B5	B4	B3	B2	B1	B0
0	Belső RESET	RTS "BE"	Hiba- törlés	BREAK kiküld.	Vétel eng.	DTR "BE"	Adás eng.

Adatbeolvasás (USART):

PORT10H (16) olvasása (0. csatlakozó)
 PORT20H (32) olvasása (1. csatlakozó)
 PORT30H (48) olvasása (2. csatlakozó)
 PORT40H (64) olvasása (3. csatlakozó)

Allapotbeolvasás (USART):

PORT11H (17) olvasása (0. csatlakozó)
 PORT21H (33) olvasása (1. csatlakozó)
 PORT31H (49) olvasása (2. csatlakozó)
 PORT41H (65) olvasása (3. csatlakozó)

B7	B6	B5	B4	B3	B2	B1	B0
DSR	BREAK vétele	Keret- hiba	Túlfu- tás	Paritás- hiba	Adópuf. üres	Karakt. jött	Adásra kész

Csatlakozó jelek:

102 (7): föld
 103 (2): adott adat (kimenet)
 104 (3): vett adat (bemenet)
 105 (4): az ellenállomás karaktert küldhet (RTS, kimenet)
 106 (5): CTS bemenet, "BE" állapotba kell kötni
 107 (6): az ellenállomás karakter fogadására képes (DSR, bemenet)
 108 (20): "BE" állapotú kimenet, nem használt
 109 (8): a 104-et kapuzó bemenet, "BE" állapotba kell kötni
 111 (23): "BE" állapotú kimenet

22. A soros vonal kezelése

```

;
OB69 2921 BAUD      DEFS 1 ; Vonali sebesség
;      0: 110 baud
;      1: 150 baud
;      2: 300 baud
;      3: 600 baud
;      4: 1200 baud
;      5: 2400 baud
;      6: 4800 baud
;      7: 9600 baud
;      8: 19200 baud
;
OB6A 2922 FORMAT    DEFS 1 ; USART üzemmód
;      0FEH (238): 8 bit, 2 STOP, nincs p.
;      0FEH (254): 8 bit, 2 STOP, páros p.
;      0DEH (222): 8 bit, 2 STOP, ptlan p.
;      06EH (110): 8 bit, 1 STOP, nincs p.
;      07EH (126): 8 bit, 1 STOP, páros p.
;      05EH (94): 8 bit, 1 STOP, ptlan p.
;      0EAH (234): 7 bit, 2 STOP, nincs p.
;      0FAH (250): 7 bit, 2 STOP, páros p.
;      0DAH (218): 7 bit, 2 STOP, ptlan p.
;      06AH (106): 7 bit, 1 STOP, nincs p.
;      07AH (122): 7 bit, 1 STOP, páros p.
;      05AH (90): 7 bit, 1 STOP, ptlan p.
;
OB71H 2929 SER_OK   DEFS 1 ; 0: a hanggenerátor a soros vonalhoz
;                          van inicializálva
;                          0FFH: a hangfrekvencia nem jó
;

```

A soros vonal funkciói

Funkció	Kód hex dec	Input	Output	Hiba hex.
SER_CHIN karakter beolvasás	E1 225		C= karakterkód	
SER_CHOUT karakter kiírás	61 97	C= karakterkód		
SER_BKIN blokk beolvasás	E2 226	DE= kezdőcím BC= hossz		
SER_BKOUT blokk kiírás	62 98	DE= kezdőcím BC= hossz		
RS232_SET seb. és USART inicializálás	E3 227 63 99			-
RS232_CLOSE nincs hatása	E4 228 64 100			-

Hibakódok:

```

0F4H (244) : paritáshiba
0F3H (243) : kerethiba (nincs STOP bit)
0F2H (242) : túlfutás (nem történt meg a kiolvasás a következő
             karakter vétele előtt)

```

23. A rendszerváltozók összefoglalása

Operációs rendszer:

0003	0003	(1)	P_SAVE	Memórialapozás
------	------	-----	--------	----------------

BASIC:

0008	0008	(25)		BASIC hibakezelő rutinok
0021	0033	(14)	USRTAB	EXT utasítások címtáblázata

Operációs rendszer:

0030	0048	(8)	OS_ENTRY	A funkcióhívások belépési pontja
0038	0056	(8)	IT_ENTRY	IT belépési pont
0040	0064	(192)		A csatolókarták azonosítói
0100	0256	(1600)		ASCII képernyő
0740	1856	(960)		A definiálható karakterek mátrixa

Rendszerváltozók:

0B00	2816	(8)	IN_TABLE	Input hozzárendelési tábla
0B08	2824	(8)	OUT_TABLE	Output hozzárendelési tábla
0B10	2832	(1)	INT_DES	Cursor IT által kiszolgált eszközök
0B11	2833	(1)	PORT03	A 3-as I/O port tükörképe
0B12	2834	(1)	PORT05	Az 5-ös I/O port tükörképe
0B13	2835	(1)	PORT06	A 6-os I/O port tükörképe
0B14	2836	(1)	SND_ACTIVE	FF: hang van folyamatban
0B15	2837	(1)	SND_IRQ	FF: az új hang megszakítja a régit
0B16	2838	(1)	STOP_FLAG	FF: CTRL+ESC-et nyomtak
0B17	2839	(2)	ST_LIMIT	A verem alsó határa képkitöltésnél
0B19	2841	(2)	HI_MEM	A legmagasabb használható RAM cím
0B1B	2843	(1)	P3RAM	0/FF: az U3 RAM jó/hibás
0B1C	2844	(1)	EX_DEF	Bővítő kártya alaphozzárendelés
0B1D	2845	(2)	INTINC	IT számláló (20 ms)
0B1F	2847	(1)	IRQ_STAT	Csatolókartya IT engedélyezés
0B20	2848	(1)	INT_FLAG	IT kiszolgálás alatt
0B21	2849	(1)	WARM_FLAG	FF: "meleg" RESET van folyamatban
0B22	2850	(1)	COLD_FLAG	FF: a "meleg" RESET tiltott
0B23	2851	(20)	INFUNC	A funkcióhívások kezdő programrésze
0B37	2871	(10)	OUTFUNC	A funkcióhívások befejező része
0B41	2881	(8)	INT_EXIT	Az IT alprogram befejező része
0B49	2889	(2)	STACKT	Verem-mutató ideiglenes tárolás

Video:

0B4B	2891	(1)	L_MODE	Vonalkeresztezési mód
0B4C	2892	(1)	L_STYLE	Vonaltípus
0B4D	2893	(1)	INK	Tintaszin
0B4E	2894	(1)	PAPER	Papírszin
0B4F	2895	(1)	BORDER	Képkeretszin
0B50	2896	(1)	V_FLAG	Karakter felülírási flag

Billentyűzet:

0B51	2897	(10)	PICTURE	A billentyűzet mátrix letapogatása
0B5B	2907	(10)	OLDPICT	A billentyűzet előző letapogatása
0B65	2917	(1)	DELAY_KEY	Auto-repeat késleltetés (20 ms)
0B66	2918	(1)	LOCK_KEY	CAPS/SHIFT/ALT lock állapot
0B67	2919	(1)	RATE_KEY	Auto-repeat ismétlési ütem (20 ms)
0B68	2920	(1)	HOLD_DIS	FF: CTRL+P hatástalan

Soros vonal:

0B69	2921	(1)	BAUD	A soros vonal sebessége
0B6A	2922	(1)	FORMAT	USART üzemmód

Kazetta:

0B6B	2923	(1)	BUFFER	0/FF: nem pufferealt/pufferealt file
0B6C	2924	(1)	REMRED	A motorvezérlés kijelölése
0B6D	2925	(1)	PROTECT	0/FF: nem védett/védett file
0B6E	2926	(1)	EOF	FF: file-vég
0B6F	2927	(2)	MUDDLE	CRC alapérték

Soros vonal/hang:

0B71	2929	(1)	SER_OK	0/FF: az órafrekvencia jó/nem jó
------	------	-----	--------	----------------------------------

Munkaterületek:

0B72	2930	(15)		Video munkaterület
0B81	2945	(100)		I/O munkaterület
0BE5	3045	(10)		Billentyűzet munkaterület
0BEF	3055	(1)		Hang munkaterület
0BF0	3056	(600)		Kazetta munkaterület
0E48	3656	(80)		Editor munkaterület
0E98	3736	(20)	PROGID	Programazonosító, "TV COMPUTER BASIC" (csak az 1.3 verziótól)
0EAC	3756	(2048)		Rendszer verem
16AC	5804	(835)		BASIC munkaterület

BASIC:

1703	5891	(1)	NEW	Ha értéke FF, parancs módban törölődik a BASIC program.
1707	5895	(1)	AUTO	Autoindítás
1708	5896	(1)	TYPE	A szimbólumtábla aktuális elemének típusa
170C	5900	(2)	START	Az aktuális BASIC sor kezdőcíme
1720	5920	(2)	VLOMEM	A BASIC terület kezdete
1722	5922	(2)	TEXT	A BASIC program kezdőcíme
1724	5924	(2)	CHAIN	A szimbólumtábla utolsó elemének címe
1726	5926	(2)	TOP	A szimbólumtábla következő szabad byte-jának címe
1732	5938	(255)	COMMAND	Az aktuális BASIC utasítássort tartalmazó puffer
1831	6193	(255)	BUFFER	Puffer az INPUT utasítás számára
19C0	6592	(7)	X	Lebegőpontos aritmetikai regiszter
19C7	6599	(7)	Y	Lebegőpontos aritmetikai regiszter
19CE	6606	(17)	FILENAME	A név nélküli LOAD, SAVE, OPEN stb. innen veszi a file nevet (normál esetben üres string).
19EF	6639			A BASIC terület kezdete

24. A hibakódok összefoglalása

KERNEL-hibák:

OFFH (255) : Nem létező funkciószám hívása
OFEH (254) : I/O hozzárendelési hiba
OFDH (253) : Nem létező kártya
OFCH (252) : Hibás blokk I/O paraméterezése (LAP3)
OFBH (251) : Nincs elég memória
OFAH (250) : Hibás blokk I/O paraméterezése (HI_MEM átlépés)

Videohibák:

OF9H (249) : Nem létező sugárpozíció
OF8H (248) : Nem (újra)definiálható karakter
OF7H (247) : Nem létező grafikus mód (színmód)

Editorhiba:

OF6H (246) : Nem létező cursorpozíció

Billentyűzethiba:

OF5H (245) : CTRL+ESC-et (STOP-ot) nyomtak

Soros vonali hibák:

OF4H (244) : Paritáshiba
OF3H (243) : Kerethiba (nincs STOP bit)
OF2H (242) : Túlfutás: a vett karaktert felülírta a következő

A kazettakezelés hibái:

OECH (236) : File-vég hiba, olvasás az utolsó karakter után
OEBH (235) : Megnyitási kísérlet nyitott file esetén
OEAH (234) : CRC hiba, a szalag sérült vagy rosszul olvasható
OE9H (233) : Nincs nyitott file
OE8H (232) : Ellenőrzési hiba komparálásnál
OE7H (231) : Belső hiba, nem megfelelő számú karakter olvasási kísérlete
OE6H (230) : A védelem megsértése
OE5H (229) : Rossz blokksorszám (szektorszám)

Megjegyzés:

A HI_MEM átlépést csak az 1.2 verziójú rendszer ellenőrzi.

190,- Ft

VIDEOTON

ELEKTRONIKAI VÁLLALAT
SZÁMÍTÁSTECHNIKAI GYÁRA