

## A BEÉPÍTETT RUTINOK HÍVÁSA

Az operációs rendszer beépített rutinjai az egy byte-os RST 30h utasítással hívhatók, melynek kódja 247, azaz 0F7h. A szükséges paramétereket a rutinok a BC és DE regiszterpárokból várják, illetve itt adják vissza az eredményt. Ezenkívül visszatéréskor az A regiszterben zérus van, ha a rutin hiba nélkül lefutott, egyébként pedig a hibakód, melynek jelentése a TVC Kezelési Útmutató 44. oldalán található. A többi regiszter tartalmát nem változtatja a rutinhívás.

Az operációs rendszer az RST 30h utasítást követő byte-ot tekint a hívott rutin kiválasztási kódjának, úgy, mintha itt nem egy, hanem kettő byte-os utasítás állna, és a második byte lenne az operandus. Gondoskodj arról, hogy visszatéréskor a kiválasztási kódot követő utasításra kerüljön a vezérlés.

### A kiválasztási kód bitjeinek jelentése:

**b7 művelet iránya:** 0=output, 1=input

**b6-b4 funkcióosztály kódja:** 0=video, 1=billentyűzet, ..., 7=Kernel

Kijelöli a hozzárendelési táblázat egy elemét a b7-tel együtt. A táblaelem alapján az operációs rendszer megállapítja, hogy melyik logikai eszköznek kell a rutinhívást kiszolgáltatnia.

**b3-b0 konkrét rutin sorszáma:** 0-15

A Kernel kivételével minden logikai eszköz első három rutinjának rögzített a funkciója:

**0: IT-kiszolgálás**

Az operációs rendszer IT-kezelést végző része hívja, felhasználói programból nem ajánlott hívni.

**1: karakter input-output**

Egy karaktert küld ki vagy olvas be. A karakter helye a processzor C regiszterére.

**2: karaktersorozat input-output**

Adott memóriaterület byte-jaira sorban elvégzi az 1. számú rutint, feltéve, hogy a memóriacím nem haladja meg a HI MEM rendszerváltozó értékét. A karakterek számát a BC regiszterpárban, a memóriaterület kezdőcímét a DE regiszterpárban várja. HI MEM változó kétbyte-os, címe 2841=0B19h. A rutinokat a logikai eszközök szerint csoportosítva ismertetjük. Megadjuk a használt rendszerváltozókat, a rutinok paraméterezését és a rövid működési leírást. A hívási kód értéke a hozzárendelési táblázatok eredeti értéke esetén érvényes. Ha a táblázatokban módosítottunk, akkor ahhoz a hívási kód megfelelő három bitjét is hozzá kell igazítani.

### FONTOS:

A beépített rutinok hívásakor a 0. lapon a rendszerterületet tartalmazó felhasználói RAM-nak kell lenni, a Stack Pointer pedig a 0. vagy 1. lapra kell, hogy mutasson!

## VIDEÓ RUTINOK

A videó rutinok alapvetően a képernyőre rajzolást végzik. Lehetőseget adnak pontok, vonalak rajzolásra, képernyőterület kifésztésére, valamint karakterek kirajzolására is a képernyő tetszőleges (!) helyén. A rajzolás egy képletbeli tollal történik, ezt mozgítja BASIC-ből a PLOT utasítás is. A toll helyét a BASIC-ben megismert koordinátákban kell megadni: X=0...1023, Y=0...959. A képernyő bal alsó sarka a 0;0 pozíció.

### A használt rendszerváltozók a következők:

**\$L MODE - 1 byte, címe 2891=0B4Bh**

Az alsó két bit a pontkírás típusát adja: 0=felülírás, 1=logikai és (AND), 2=logikai VAGY (OR), 3=logikai KIZÁRÓ VAGY (XOR). BASIC-ből a SET MODE állítja.

**\$L STYLE - 1 byte, címe 2892=0B4Ch**

Az alsó négy bit a vonaltípust adja. BASIC-ből a SET STYLE állítja.

**\$INK - 1 byte, címe 2893=0B4Dh**

A színfelbontástól függően az alsó egy, kettő vagy négy bit a tintaszint (palettakódot) adja. BASIC-ből a SET INK állítja.

**\$PAPER - 1 byte, címe 2894=0B4Eh**

Mint a \$INK, csak a papírszínre. BASIC-ből a SET PAPER állítja.

**\$BORDER - 1 byte, címe 2895=0B4Fh**

Az aktuális keretszint adja. A BASIC-beli SET BORDER paraméterének kétezeresét tartalmazza.

**\$V FLAG - 1 byte, címe 2896=0B50h**

Az alsó két bit a videó karakter kirajzolást vezérli: az 1 értékű bit leltítja a 0 értékű bit pedig engedélyezi a karakter megfelelő pontjainak kirajzolását; b0=tintaszínű pontok, b1=papírszínű pontok. A pontok kírása \$L MODE szerint történik.

### A rutinok leírása:

**VID IRQ - hívási kód: 0**

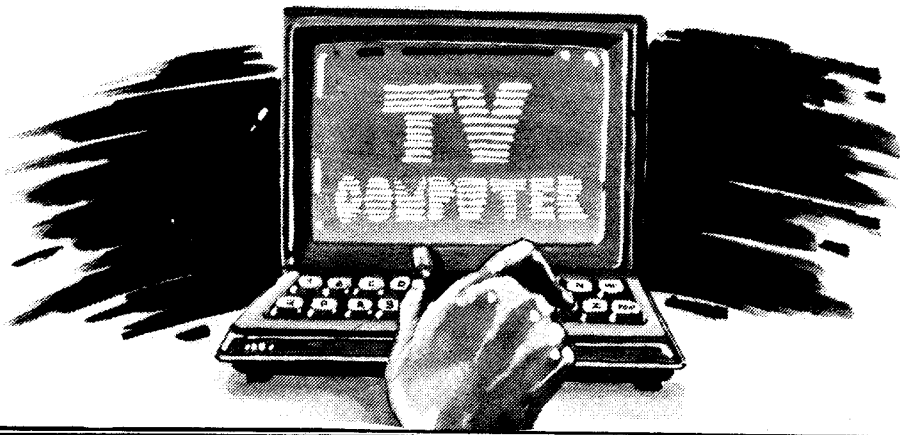
**működés:** Nincs hatása.

**VID CHOUT - hívási kód: 1**

**input:** C=kiírandó karakter

**output:** A=hibakód

**működés:**



## ELSŐ KÉZBŐL

### A TV COMPUTER RŐL

Kirajzol egy karaktert. Az aktuális rajzpozíció lesz a kiírt karakter bal felső sarka. Kiírás után a rajzpozíciót úgy állítja, hogy a következő karakter jobbról kövesse a mostanit. Ha a kép jobb szélén nem fér ki a karakter, akkor tíz pontsorról lejjebb a balszélen folytatja, ha alul nem fér ki, akkor nem ír ki semmit, tehát nem rollozza a képet. A vezérlő karakterek közül csak a CHR\$(10) és CHR\$(13) hatásos, a többire, valamint 223-nál nagyobb kódú karakterekre nem reagál.

**VID BKOUT - hívási kód: 2**

**input:** BC=kiírandó karakterek száma

DE=első karakter címe

**output:** A=hibakód

**működés:** A karakterket egyenként kiírja a VID CHOUT rutinnal.

**BTEXT - hívási kód: 3**

**input:** B=oszlop

C=sor

**output:** A=hibakód

**működés:**

A rajzpozíciót beállítja a B;C karakterpozíció bal felső sarkába, ezzel előkészíti a normál szövegsorba való írást a VID CHOUT rutin számára. A képernyő bal felső karakterének koordinátái: 1;1. Ha B vagy C értéke zérus, akkor az a pozíció változatlan marad. Ez a rutin soha nem húz vonalat!

**V MODE - hívási kód: 4**

**input:** C=grafikus mód: 0=kétszínű, 1=négy színű, 2=tizenhatszínű

**output:** A=hibakód

**működés:**

Beállítja a grafikus felbontást. Hatása teljesen azonos a BASIC-ben kiadott GRAPHICS utasítással.

**CLS - hívási kód: 5**

**működés:** Törli a képernyőt, mint a BASIC-ben kiadott CLS parancs.

**BABS - hívási kód: 6**

**input:** BC=X koordináta: 0...1023

DE=Y koordináta: 0...959

**output:** A=hibakód

**működés:**

Beállítja a rajzpozíciót az X;Y koordinátákra. Ha a jelképes toll le van téve, akkor vonalat is húz a rendszerváltozóknak megfelelően.

**BREL - hívási kód: 7**

**input:** BC=X relatív koordináta

DE=Y relatív koordináta

**output:** A=hibakód

**működés:**

Hasonló a BABS rutinhoz, csak itt a megadott koordinátákat hozzáadja az aktuális értékekhez, így kapja az új pozíciót.

**BON - hívási kód: 8**

**működés:**

A jelképes tollat leteszi, és kiír egy pontot \$INK és \$L MODE szerint. Ettől kezdve a BABS és BREL rutinok vonalat húznak.

**BOFF - hívási kód: 9**

**működés:**

A jelképes tollat felemeli. Ettől kezdve a BABS és BREL rutinok csak pozícionálást végeznek.

**FILL - hívási kód: 10 (0Ah)**

**működés:**

Beírt az aktuális rajzpozícióval szomszédos és azonos színű képpontokat a \$INK szerinti színre az "özönvíz" algoritmus szerint. A rutin rekurzív, verem-orientált. Hatása azonos a BASIC-ben kiadott PLOT PAINT utasítással.

**DEFC - hívási kód: 11 (0Bh)**

**input:** C=a definiálni kívánt karakter kódja: 128...223

DE=a karaktert definiáló 10 byte kezdőcíme

**output:** A=hibakód

**működés:**

Átírja a karakter képét a RAM-ban. Hatása azonos a BASIC-ben kiadott SET CHARACTER utasítással.

**PAL - hívási kód: 12 (0Ch)**

**input:** DE=paletta-definiáló 4 byte kezdőcíme

**működés:**

Definiálja a palettaszíneket a 2 és 4 színű üzemmódhoz. Hatása azonos a BASIC-ben kiadott SET PALETTE utasítással.

Cseh Tibor

A szerkesztő azért van, hogy a lap olyan legyen, amilyenek az olvasói!

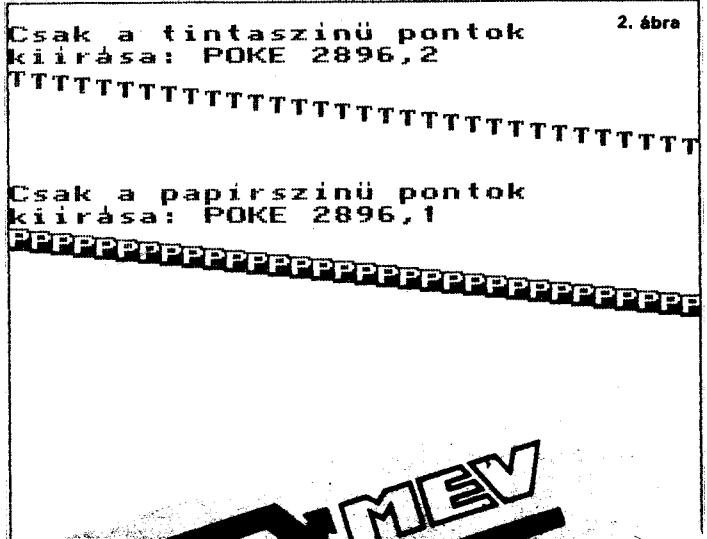
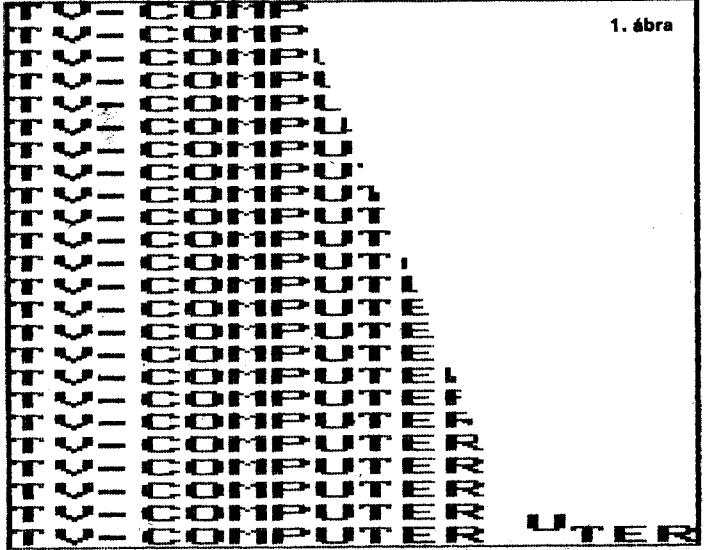
# KARAKTERKIÍRÁS BASICBŐL VIDEO RUTINNAL

A program bemutatja a videó rutinnal történő karakter- és stringkiírást. Ez gyakorlatilag lehetőséget biztosít a képernyőn kis ábrák mozgására, pl. saját tervezésű karakterek (figurák) segítségével.  
 A program a kezdőszöveg kiírása után egy billentyű leütésére vár (110-199. sor). Az első részben (210-250. sor) a „TV-COMPUTER” stringet kiírja 16 színű módban minden sor elejére, majd az első sorból a karakterek egyenként elvándorolnak az alsó sor jobb oldalára, miközben beletörölnek a képernyőn levő szövegbe (260-340. sor, 1. ábra). Amikor az egész string összeáll, akkor függőlegesen felfelé mozog az első sorig (410-440. sor), majd a sor végétől a sor elejére, és egy billentyű leütésére vár (510-590. sor).  
 A második rész bemutatja, hogy kiírhatók külön a karakter tintaszínű és papírszínű pontjai (610-750. sor, 2. ábra)

Cseh Tibor

```

10 !
20 !-----!
30 !
40 ! TV-COMPUTER
50 !
60 ! Karakterkiírás a beépített
70 ! VIDEO rutinnal
80 !
90 !-----!
100 !
110 GRAPHICS 4:RANDOMIZE
120 PRINT AT 5,4:"Karakterkiírás a beépített"
130 PRINT AT 8,8:"VIDEO rutinnal"
199 GET !<--- vár egy billentyűre
200 !-----!
202 ! egy karakter mozog
204 !-----!
210 GRAPHICS 16
220 T$="TV-COMPUTER":T=LEN(T$)
230 FOR I=1 TO 24
240 : SET INK 9+RND(7):PRINT AT I,1:T$;
250 NEXT I
260 DX=(16-T)*64/230
270 FOR I=T TO 1 STEP -1
280 : SET INK 9+RND(7)
290 : X=(I-1)*64-DX
300 : FOR J=956 TO 36 STEP -4
310 : X=X+DX
320 : PLOT X,J:PRINT #0:T$(I);
330 : NEXT J
340 NEXT I
400 !-----!
402 ! string mozog
404 !-----!
410 FOR J=36 TO 956 STEP 4
420 : SET INK 9+RND(7)
430 : PLOT X,J:PRINT #0:T$;
440 NEXT J
500 !-----!
510 FOR J=X TO 0 STEP -4
520 : SET INK 9+RND(7)
530 : PLOT J,956:PRINT #0:T$;
540 NEXT J
599 GET !<--- vár egy billentyűre
600 !-----!
602 ! V_FLAG bemutatása
604 !-----!
610 V_FLAG=2896
620 GRAPHICS 4
630 TINTA=1:PAPIR=2
640 PRINT:PRINT "Csak a tintaszínű pontok"
650 PRINT "kiírása: POKE 2896,2";:POKE V_FLAG,2
660 FOR I=0 TO 31
670 : SET INK 0:PLOT 32*I,836-4*I
680 : SET INK TINTA:PRINT #0:"T";
690 NEXT I
700 PRINT AT 9,1:"Csak a papírszínű pontok"
710 PRINT,"kiírása: POKE 2896,1";:POKE V_FLAG,1
720 FOR I=0 TO 31
730 : SET PAPER PAPER
740 : PLOT 32*I,556-4*I:PRINT #0:"P";
750 NEXT I
800 !-----!
999 GET:PRINT AT 20,1;
    
```



**MEV**  
**TELETECHNKA**  
**TELEKOMUNIKÁCIÓK**  
**INTEGRÁLT ÁRAMKÖRÖK**  
**ES CSATLAKOZÓK.**  
 SZAKTANÁCSADÓ ÉS SZERELŐ IRODA