



Az Országos Commodore Egyesület lapja

újság

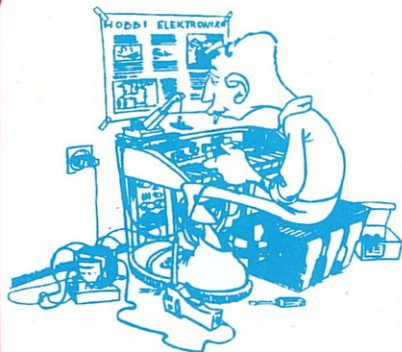
1992/12

Változói lista

SPRITE

Plus 4-en





HOBBI ELEKTRONIKA

Urbán István mérnök áramköreinek szaküzlete

Budapest VII., Dózsa György út 16. (Dózsa-Jobbágy sarok)
Nyitva: H-P 10-17-ig Tel./fax: 122-8892
(Zárás után üzenetrögzítő)

**Super
ajánlat!**

Több, mint tíz éve fókusz a Rádiótechnikában, vezetem a „Zenel hobby elektronika” fejlécű rovatot. A közölt témák nagyrészt szaküzletben - működő minta alapján - vásárolhat panelokat, részegységeket, egységcsomagokat.

KÍNÁLATUNKBÓL:

SZÁMÍTÁSTECHNIKA:

C64 bővítők:		egys.	éleszt.
RT89/3	PLOFI Datassette cartridge	1200 Ft	1500 Ft
	PLOFI Datassette/promon	1300 Ft	1700 Ft
	PLOFI Datassette/help	1300 Ft	1700 Ft
RT89/10	PLOFI Fastload cartridge	1300 Ft	1700 Ft
	PLOFI Fastload speedtape	1400 Ft	1900 Ft
RT90/3	PLOFI Simon's cartridge	1300 Ft	1700 Ft
	FINAL III cartridge		3450 Ft
	ACTION Replay VII		3450 Ft
RT89/11	Fényceruza, szoftverrel	950 Ft	1250 Ft
RT90/8	Hangdigitalizáló + szoftver	1350 Ft	1650 Ft
HE90/8	Hangkapcsoló	300 Ft	
RT89/4	EPROM-égető	3400 Ft	4500 Ft
	Égető szoftver lemezen	1300 Ft	
	Égető szoftvercartridge	1300 Ft	1700 Ft
	User csatlakozó	300 Ft	
RT91/3	EPROM-bank (256 Kb-át)	2900 Ft	4000 Ft
RT91/10	IC teszter	3000 Ft	4500 Ft
RT91/7	Datassette gyorsmásoló	600 Ft	
HE91/7	Datassette fejbeállító	350 Ft	
HE90/12	CPU stop + reset	400 Ft	
	PAGEFOX szövegszerkesztő		3900 Ft
	Mini EPROM-bank	2300 Ft	3000 Ft

IBM bővítők:

	egys.	éleszt.
RT91/4,5 IBM IC teszter + szoftver	5900 Ft	9800 Ft
RT91/11,12 IBM EPROM-égető		
+ szoftver	4800 Ft	9900 Ft
RT91/6 48 csat. IO kártya		3500 Ft

HE92/2	User-Centronics csatlakozó	950 Ft
RT92/3	TTL IC katalógus lemezen	600 Ft
	CMOS IC katalógus lemezen	600 Ft
	Dióda katalógus lemezen	600 Ft
	Tranzisztor katalógus lemezen	600 Ft

ZENE - HANGTECHNIKA:

RT87/10	Fuzz-box torzító	670 Ft
RT90/1	KORG DST-1 torzító	1200 Ft
	KORG DST-3 torzító	1400 Ft
HE90/1	KORG OVERDRIVE torzító	920 Ft
RT87/12	Vau-vau gitáreffekt	640 Ft
RT88/9	Sztereo tremoló	980 Ft
RT88/7	Shifter	1200 Ft
HE90/7	Kiszajú gitárelőerősítő	700 Ft
RT89/5	Sztereo előerősítő	1500 Ft
HE91/12	Hangfrekvenciás erősítő	490 Ft
RT89/8	2 x 14 W sztereo erősítő	990 Ft
RT89/12	2 x 40 W sztereo erősítő	1800 Ft
RT89/1	100 W-os erősítő	1600 Ft
RT90/3	Ritmuszép 32 ritmussal	2500 Ft
RT90/10	Dobszintetizátor	3500 Ft
RT88/3	Süvítő gitáreffekt	780 Ft
RT87/11	Ringmodulátor	995 Ft
RT92/4	Oktávemelő	700 Ft
	Visszhangosító MN3005-tel	4800 Ft

VIDEÓS TÉMÁK:

RT92/6	RGB generátor	1760 Ft
RT92/11	PAL kóder	1400 Ft

FÉNYTECHNIKA:

RT88/10	Diszkófény II (triak nélkül)	1400 Ft
	Diszkófény II (triakkal)	3000 Ft
RT88/12	Fényorgona	2000 Ft
RT90/2	Programozható futófény	1200 Ft
RT91/1	8 x 8-as fénymátrix + 16 K EPROM	2000 Ft
	8 x 8-hoz LED panel 10 mm-es LED-ekkel	2000 Ft
RT89/6	Kivezérlésléző	700 Ft
RT91/8	Sziporkázó szingyűrű	650 Ft
	Kétszínű LED	48 Ft
HE91/10	Karácsonyi fényjáték	700 Ft
HE92/8	Knight Rider futófény	950 Ft

EGYÉB HOBBI TÉMÁK:

RT88/10	Dallamgenerátor Z80-nal	1500 Ft
HE91/1	Dallamgenerátor UM3481-84	800 Ft
HE91/5	Dallamgenerátor UM66Tx	480 Ft
HE91/4	Beszédfelismerő VCP200-zal	750 Ft
	VCP200	1950 Ft
HE91/2	Kutyugatás szintetizátor	800 Ft
RT91/8	Eb-vezérlő	400 Ft
HE91/6	Szúnyogriasztó	700 Ft
HE91/9	Multiszíréna	480 Ft

Az árak a 25% ÁFA-t tartalmazzák!

Választékunk folyamatosan bővül! A Rádiótechnikában, a Hobby Elektronikában megjelent nyák-tervek alapján készült nyomtatott áramköri lapok az üzletben megvásárolhatók.

MŰSZERVÁSÁR

Bontott anyagok,
használt műszerek
bőngésződjé



Állandóan változó készlet!

Vidéki olvasóknak segít a szerző levelező, egységcsomagküldő szolgáltatása: a megrendelt csomagot postán utánvétellel elküldöm. Telefonon/faxon és levélben is rendelhet.

A HOBBI ELEKTRONIKÁ-hoz nem kell hosszú levél. Rendelését néhány sorban, egyértelműen közölje.

Levél cím: 1656 Budapest Pf. 50.

Az üzletben megvásárolhatók az RT évkönyvei, a Rádiótechnika és a Hobby Elektronika korábbi számai.



MIT, HOGYAN, HOL, MIKOR?

EGYESÜLETI ÜGYEK: Egyesületünknek tagja lehet mindenki, aki a tagsági díjat befizeti. A tagdíjat személyesen az egyesület irodájában (1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57), vagy átutalással az MNB 217-98 292, OTP 565-3610-8 számlára lehet befizetni. Megrendelés esetén számlát küldünk.

Pötyөгőszolgáltatunk valamint a szervizkedvezmény és az apróhirdetés lehetősége tagjaink rendelkezésére áll.

A **DEÁKPÁHOLY** tagjai minden hónapban megkapják a C-újságot, a tagsági díj egy évre 777 forint.

A **PLUSZPÁHOLY** tagjai minden hónapban megkapják a C-újságot, és kapnak havonta 3 db vásárlási utalványt. A tagsági díj egy évre 1888 Ft.

A **SZUPERPÁHOLY** tagjai havonta 15 példányt kapnak a C-újságból, és ezzel havonta 15x3 db vásárlási utalványt is. Az éves tagsági díj 20 900 Ft.

ÜGYFÉLFOGADÁS: Minden kedden és csütörtökön 12—16 óra között várjuk tagjainkat és az érdeklődőket.

PÖTYөгŐSZOLGÁLAT: Az újságban megjelenő programokat másolja a megrendelők részére. Megrendelhető személyesen az egyesület irodájában vagy postai utánvétellel. Postacím: 1388 Budapest 62., Postafiók: 86.

APRÓHIRDETÉS: Az egyesületi tagoknak ingyen áll rendelkezésére. Nem tagoknak a hirdetés ára 100 forint. A hirdetés módja: az újságban megjelenő nyomtatvány kitöltésével.

A **C-ÚJSÁG RÉGEBBI SZÁMAI** megvásárolhatók az egyesület irodájában, vagy megrendelhetők utánvétellel.

Kedvezményes ár! Tagoknak olcsóbb!

Az újságban eddig megjelent programok gépenként összegyűjtve megrendelhetők. VC 20, C16, PLUS/4, C128, C64. További felvilágosítást is adunk a 1-76-22-57-es telefonszámon vagy levélben!

Vidéki pluszpáholy-tagjaink háromhavi tikett összegyűjtésekor igénybe vehetik a NOVOTRADE 2C Áruház csomagküldő szolgálatát.

VIDÉKEN TOVÁBBI INFORMÁCIÓK KAPHATÓK:

Baja, **AXIS Kft.**,
Győri Bartók Béla Művelődési Ház,
Jászberényi Városi Könyvtár,
Kecskemét, **SZIGMA—BIT**,
Pécsi Apáczai Csere János Gimnázium,
Zalaegerszegi Ságvári Endre Gimnázium.

Az Országos Commodore Egyesület módszertani kiadványa

Egyesületi iroda és szerkesztőség:
1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57
Felelős kiadó: Horváth Judit, az egyesület elnöke
Főszerkesztő: Rados Péter, az OCE főtítkára
Felelős szerkesztő: dr. Horváth András
Művészeti szerkesztő: Bausz Sándor
Levél cím: Commodore Újság, 1388 Budapest, 62. Pf.: 86.
Index: ISSN 0237-756 X
Terjeszti a Magyar Posta
Megvásárolható a hírlapárusoknál
92.0305 MSZH Nyomda és Kiadó Kft., Budapest
Felelős vezető: Nagy László

Tagdíjbefizetés

Kedves Tagtársaink!

Az év vége közeledtével felhívjuk figyelmüket az 1993. évi tagdíjuk befizetésére. Bár nem szeretnénk, de lehet, hogy kénytelenek leszünk emelni a tagdíjakat.

A tavalyihoz hasonlóan azonban az idén is lehetőséget adunk arra, hogy aki még 1992-ben befizeti a jövő évi tagdíját, az mindenképpen a régi áron teheti.

(Deákpáholy 777 Ft, pluszpáholy 1888 Ft, szuperpáholy 20 900 Ft.)

Régi tagjaink a csekket az újsággal együtt kapják meg, de elküldhetik a tagdíjat rózsaszínű postai utalványon is OTP számlaszámunkra. (MNB 217-98 292, OTP 565-3610-8)

Országos Commodore Egyesület

Újra indul

a Commodore Egyesület klubdélelőttje a Havana Közösségi Házban

(1181 Budapest, Kondor Béla sétány 8.).

Megközelíthető: a metró Kőbánya-Kispest-i végállomásától a piros 135-ös busszal.

Várunk benneteket minden hónap 3. vasárnap délelőtt 9 órától.

Legközelebb:

december 20-án.

FELHÍVÁS

Tisztelt Tagtársunk, Olvasóink!

Kérjük, hogy ha rendelkeznek olyan működő számítógéppel, vagy jó állapotban lévő bármilyen számítógéptartozékkal amire nincs szükségük, akkor hozzák el egyesületi irodánkba. (Minden héten kedden és csütörtökön 11-15 óra között.)

Mi összegyűjtjük és eljuttatjuk határainkon túli (erdélyi, kárpátaljai és szlovákiai) magyar iskolákhoz.

Köszönettel:

Országos Commodore Egyesület

Tisztelt Szerkesztőség

Az 1992/5. sz. 19. lapon megjelent „Adatmentes” ismét írásra ösztönzött. Én úgy gondolom, a javasolt módszer nem más, mint floppyszekálás. A kísérleteket nem a hibás blokkok beolvasása céljából építették be, hanem a floppy működési zavarainak kiküszöbölésére!

Ha a leírtak igazak lennének, csak úgy vezethetnének eredményre, ha teljesülne: hibás blokk + hibás beolvasás = helyes végeredmény. Ennek még annyi esélye sincs, mint a blokkhibának!

Tapasztal ismerősöm az EX-DOS-t javasolta hasonló feladatra azzal, hogy addig ismétlem, amíg egyszer sikerül a beol-

vasás; ha sikerült, csak vissza kell írni és kész. Megpróbáltam, ha a türelmemet nem veszítem el, még mindig hallgathatom a program átható gonghangját, amelyet hibajelzéskor ad.

Mindkét megoldás hasonló, a kísérletek számának emelése. Olyan ez, mintha a magasugrónak világcsúcsmagasság kétszerezésén további kísérleteket adnak egy versenyen. Nem akkor, de életében nem ugorja át! A floppy sem olvassa be a hibás blokkot!

Más módszert kell keresni, ami van is, mert azzal én már több lemezemet, programomat állítottam helyre.

A legegyszerűbb módszer, a fájlok tartalékmezre mentése, ezt mindenki ismeri, csak lemez és figyelem kell hozzá!

Nem kívánom részletezni a technikát, csak a javítás elvét, mert a kivitelezéshez speciális ismeretekre van szükség. Az ismeretek közlésére egy levél nem elegendő.

A javítás menete:

1. A hibás blokkot keresse meg (pl. EX-DOS-al).
2. Reset után töltsse be a „Disk-Monitor”-t. (A nagy floppy könyv/273 old.)

3. Olvastassa be a blokkot, ne törődjön azzal, hogy a lemezegység lámpája villog, írassa ki a képernyőre a blokkot. A hibásan beolvasott blokk is kiírható! A hibajelzést adó monitorokkal ez nem kivitelezhető! Ha kiírható, akkor javítható is és visszaírható. Ez adja a javítás lehetőségét!

Tapasztalatom, hogy a blokkláncolással történnek általában a hibák: a sáv, vagy a szektor értéke nem kerül felírásra, a bitek értékei a monitoron eggyel előbbre csúsznak. Ha a javító ismeri a blokkláncolás menetét, belevághat. Jegyezze fel az első bitek értékét (javításhoz majd kellhet), írja fel a (ön szerint) következő blokkcím bitjeit, írja vissza a blokkot a lemezre. Ezután az EX-DOS-al már javíthat, nem kap hibajelzést, akár a fájl-beolvasást is megkísérelheti! A blokkcímek után egy INS-DEL és a monitorral felülírt (feljegyzett) bit beírható.

Legkönnyebb javítani a lemez tartalomjegyzékét, mert ott minden bitnek pontos helye van.

Ha minden javítási kísérlet (lemez anyaghiba miatt) eredménytelen, a fájl előző blokkjának első két bitjével megváltoztatásával ugorja át a hibás blokkot, így csak egy blokk veszik el, a forrásanyag felhasználásával a kiegészítés könnyebb, mint az újraírás.

Megfelelő ismeretek nélkül a lemezt úgy elronthatja, hogy arról a varázsló sem olvas be programot! Ezért a hibás lemezről, ami beolvasható, másolja tartalékmezre. Általában a javítási munkát célszerű így kezdeni!

A javítási munka elméleti ismeretek alapján, gyakorlással, nyomozói szimattal, kevés szerencsével eredményessé tehető. Pontosabban, ezzel a módszerrel legalább megvan az esély az eredményes javításra.

Tisztelettel

Mesterházi Sándor

Tisztelt Pötyögőszolgálat!

Plus/4 szövegszerkesztőjén készítem — részben már kézírás-fogalmazott — regényemet. Jelenleg már két lemeznyi írásom van készen — sajnos, teljesen nélkülözve a magánhangzókra szükséges ékezeteket. Itt, helyben teljesen kudarcot vallott minden igyekezetem, hogy Mr. David Johnson egyébként remek rendszerében magyarul írhassek.

Minthogy egyébként nem állok közel a számítógép-technikához, véletlenül szereztem tudomást a Commodore Újságról, így az Önök szolgáltatásairól is. Ennek nyomán újra feléledt a remény bennem, hogy a rengeteg karakterre utólag talán kirakhatom a szép helyesírás ékezeteket.

Kérem tehát Önöket, szíveskedjenek küldeni nekem eprom vagy modul formájában egy magyar helyesírási karakter-készletet. Az újságjuktól, de egyéb forrásokból is úgy tudom, hogy ez elképzelhető, és igazán nagy örömmel fogadnám, ha a pótláson túl, továbbra is a C+64-gyel dolgozhatnék, annak a Johnson-féle szövegszerkesztő programjával.

A keresgélés közben már eljutottam a NOVOTRADE-ig, ahova kétszer is visszatért levelem után a harmadik már eljutott. Nyilván a költözködés is oka annak, hogy választ máig még nem kaptam. Mivel a postacímük nem azonos, lehet hogy Önökhöz el sem jut az azonos kérésem. Ha mégis, akkor most már kétszeresen is remélem, hogy egy példányban megkaphatom az áhított karakter-készletet.

Nagyon kérem, hogy azt, utánvételezés mellett, szíveskedjenek postán megküldeni. Minthogy nem vagyok gyakorlott számítógépes, kérem, hogy az esetleg szükséges tudnivalókat is közöljék.

Tisztelettel:

Szertics László

8800 Nagykanizsa, Bartók B. u. 6.

Nekünk sincs a Johnson-féle szövegszerkesztőhöz ékezetesítő programunk, de talán olvasóink közül valaki tud segíteni. Kérjük, jelentkezék Szertics Lászlónál.

A szerk.

Tisztelt Szerkesztőség

Az alábbi cikket azért írtam, mert jómagam a GEOS felhasználók szűkös táborához tartozom. Programjaim többsége gyári eredetű, ennek ellenére kisebb fortélyok rejtve maradnak számomra is. Ez a tény ösztönzött a cikk megírására.

GEOS Magyarországon

A GEOS azoknak, akik ismerik, egy megbízható, sokrétű programcsomagot jelent, másoknak pedig semmit. Miért van az, hogy nálunk ez a programrendszer nem terjedt el? Valószínűleg azért, mert senki nem fogja össze és terjeszti a programokat ellentétben Németországgal, ahol van terjesztője a GEOS-nak, a GEOS User Club, ahová új programért és segítségért fordulhatnak.

Nálunk sajnos ilyen klub nincs, holott volna rá szükség. A GEOS-szal foglalkozó írásokkal és cikkekkel sem vagyunk bőségesben. Egyedüli komolyabb újság ebből a szempontból a Commodore Újság. Holott a GEOS rendszer nagy testvéreinek közeli kezelése és hatékonysága bármilyen számítástechnikai lapban egyoldalas állandó rovatot érdemelne ki.

A szoftver-ellátottság régebben nem létezett. Szerencsére a Markt & Technik forgalmazója által már mindegyik program megrendelhető Budapesten. Igaz, 2–3000 Ft, mely nem lebe-

csülendő kiadás. De egy előnye van, az, hogy eredeti! Működése kifogástalan és a kezelői könyv segítségével a megvett program kezelése egyszerű.

Mivel nem mindenkinek van lehetősége a vásárlásra, ezért másolt programot használ, kezelése pedig csak az elmondott és sokáig tartó próbálgatás árán válik fokozatosan egyszerűvé. Így a rendszer teljes képességei kihasználatlanok maradnak. Ezen csak az segítene, ha a GEOS Felhasználói Klub megalakulna, így összefogva a GEOS „szerelmeseit”. Erre lehetőség adódna minden bizonnyal.

A gondolataimat a GEOS 2.0 és a GeoWrite Workshop segítségével vettem papírra.

Nagy Péter
1056 Budapest, Molnár u. 20

A szerk. megj.: Nagy Péter levelét azért közöljük, hogy a GEOS klub iránt érdeklődők felvehessék vele a kapcsolatot. Reméljük, a klubnak nem az illegálisan másolt programok dokumentációival történő ellátása lesz a fő feladata.

Tisztelt Szerkesztőség

T+T 64

Az 1992. 7–8. újságban megjelent a 45. oldalon egy Szövegkiadás jobb kötésben című cikk. Az itt lévő parancssort rövidebben is megfogalmazhatjuk:

```
FOR I=1 TO 40—LEN(A$):? " ";:NEXT:?A$
helyett:
? SPC (40—LEN(A$))A$
A hatás megegyezik.
```

Szűcs Gábor
(Hungary Soft)

Tisztelt Szerkesztőség

A hibák mellett nem mehetünk el szótlánul; főképpen akkor nem, ha mi követtük el azokat. Most több hibát találtam!

1. Hiba a reklámban:

A lap rendszeres reklámja (Novotrade Szerviz Kft.) évek óta hibásan tartalmaz egy szombathelyi telefonszámot. A nevet még az új telefonkönyv is rosszul tartalmazza, így ott is nehezen található meg. Én is csak a tudakozótól tudtam meg, hogy a reklámban levő sor pontosan így szól:

9700 Szombathely, Szalonok u. 31. Telefon: 94-14-519

2. Hiba a UNISEQ szövegében:

'92/9. szám 4. old. Van egy másik módszer is... bekezdésből egy általam írt sor szedésakor kimaradt. Az helyesen az alábbi: ... (C-64 belső felépítése című könyvből). A rutin SYSP0,X,Y,... paranccsal hívható. A rutin a memória más helyén is elhelyezhető...

3. IF THEN...

A szakirodalom általános kritikájában ('92/7—8 Karakter-szerkesztő) bölcsen megállapítottam, hogy az
IF C THEN900

utasítássor esetében a gép megáll, mert nem tudja értékelni a feladatot. A leírás óta már én is tudom, hogy ez nem igaz! Elnézést kérek! Csak ellenőrizni kellett volna! Ezt biztosan más meg is tette. Ez is egy sajtóságos írásmód, mint a 0 helyett . (pont) írás, amit én is szívesen alkalmazok.

A helyes értelmezés:

```
IF C THEN 200 azonos IF C∅ THEN 200
```

4. Hexadecimális táblázat készítése a memóriában:

'91/11. szám 13. oldalán Egy trükk címmel azt fejtegettem, hogy ezt a speciális feladatot a READ-DATA kombináció helyett kevesebb írással meg lehet oldani. A javasolt számátalakítás azonban lassú. A következtetésem az volt, hogy a kettő közül kell választani. Ez sem igaz!

Van egy harmadik lehetőség is: a szövegátalakítással! Ez is rövid rutin, és kb. olyan gyors, mint a READ-DATA mód. Teljesség kedvéért mindhárom megoldást leírom.

1. mód:

```
10 DIM H$(255):REM READ—DATA MEGOLDAS
100 FORI=0TO255:READH$(I):NEXTI
30000 DATA "00", "01", "02",...
```

.....

```
30090 DATA ....., "FD", "CE", "FF"
```

2. mód:

```
1159 DIM H$(255):REM '91/11. SZAMATALAKITASSAL
1160 FORI=0TO255:H=I
1161 FORJ=ITO0STEP—1
1162 A=INT(H/(16∅J)):H=H—(A*(16∅J))
1163 H$(I)=H$(I)+CHR$(A+48—(A>9)*7)
1164 NEXTJ:PRINT I" = "H$(I)
1165 NEXTI
```

3. mód:

```
1159 DIM H$(255)=REM SZOVEGATALAKITÁSSAL
1160 A$="0123456789ABCDEF":O=0
1161 FORI=0TO240STEP16:O=O+1
1162 FORJ=0TO15
1163 H$(I+J)=MID$(A$,0,1)+MID$(J+1,1)
1164 NEXTJ:PRINTI+J" = "H$(I+J)
1165 NEXTI
```

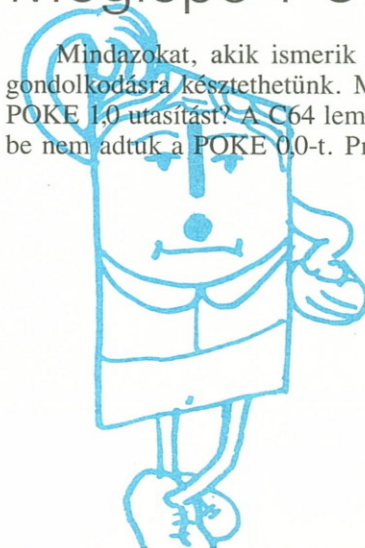
Stopper nélkül is érezhető a különbség! Van más mód is? Mivel a saját hibák bölcsé teszik az embert, azt kell mondanom, nincs kizárva!

Tisztelettel

Mesterházi Sándor

Meglepő POKE 1,0

Mindazokat, akik ismerik az 1-es tárolócella funkcióját, gondolkodásra készíthetünk. Mert mi történik, ha beadjuk a POKE 1,0 utasítást? A C64 lemerevedik? Le, kivéve, ha előbb be nem adtuk a POKE 0,0-t. Próbáljuk csak ki.





SPRITE Plus-4-en

Ez a program lehetővé teszi, hogy 8×24 képpontból álló Sprite-ot pontonként mozgathassunk. Ezenkívül tartalmazza az alakzat megtervezéséhez és kezeléséhez szükséges BASIC kiégészítést. Betöltés után RUN-nal indítható.

Az utasítások és értelmezésük:

SPRITE X, Y: X lehetősége értéke: 0—296
Y lehetősége értéke: 0—192
ezen értékeknél a sprite a képernyőn marad a sprite megjelenítése

SPRON: a sprite kikapcsolása

SPROF: a sprite színének kiválasztása

SPRCOL X: X lehetősége értéke: 0—255

EDIT: a sprite szerkesztése. A képernyő jobb felső sarkában megjelenik az alakzat, középen pedig

a nyolcszoros nagyítás egy 8×24 -es téglalapban. Ebben szerkeszthetjük újra a sprite-ot egy kereszttel, amit a kurzormozgató billentyűvel irányíthatunk. A kereszt alatti pont szóköz billentyűvel be-, INST/DEL billentyűvel kikapcsolható. A szerkesztőből RETURN-nel lehet kilépni.

KILL: a program megsemmisítése

SCREEN X: ha X=1, akkor az alakzat színe dominál, ha X=0, akkor az alakzat háttérszínű.

WORDS: Más értékre hibát jelez. Az utasítások kiírása.

Németh Gábor

```

0 GRAPHIC1:GRAPHIC0
1 COLOR0,1:COLOR4,1:COLOR1,2:SCNCLR
2 PRINT"MMM SPRITE BASIC BETOLTO"
3 PRINT"MMM KIS TURELMET KEREK !"
4 PRINT"MMM A DATA-SOROK HAMAROSAN BETOLTODNEK"
5 PRINT"MMM DATA-SOR: (MAX.:249)"
6 FORX=0TO3199STEP16:CHAR1,11,11,STR$(X/16+50)
7 FORY=XTOX+15
8 READB#:POKE13184+Y,DEC(B#):NEXT:NEXT
9 SYS13294
50 DATA9,79,85,01,A9,0C,85,02,A2,00,8D,03,30,40,A0,07
51 DATA68,4A,48,90,03,A9,A0,2C,A9,20,91,01,88,10,F1,68
52 DATAA5,01,18,69,28,85,01,90,02,E6,02,E8,E0,08,90,DA
53 DATAC6,02,A5,01,38,E9,38,85,01,00,02,C6,02,AD,8B,33
54 DATA18,69,08,8D,88,33,C9,08,90,0E,68,85,02,68,85,01
55 DATA60,02,60,53,5A,4F,56,41,4C,20,40,49,20,4E,45,20
56 DATAA5,01,48,A5,02,48,A9,C0,8D,8B,33,4C,80,33,EA,A9
57 DATA01,85,2B,A9,40,85,2C,A9,00,8D,00,40,4C,00,36,00
58 DATA20,4F,FF,93,20,20,53,50,52,49,54,45,20,45,44,49
59 DATA54,20,28,32,34,2A,38,29,20,3A,11,00,20,73,37,A9
60 DATA01,8D,ED,3C,A9,20,8D,EE,3C,8D,EF,3C,A9,78,85,01
61 DATAA9,0C,85,02,85,04,A2,08,A0,00,A5,01,18,69,19,85
62 DATAD3,A5,04,69,00,85,04,A9,6E,91,01,91,03,A5,01,18
63 DATA69,28,85,01,A5,02,69,00,85,02,85,04,CA,08,0B,A9
64 DATA0D,20,02,FF,4C,72,34,A2,1A,A9,A6,20,02,FF,CA,00
65 DATAF8,60,20,67,34,A2,09,A9,0D,20,02,FF,CA,08,F8,20
66 DATA67,34,A9,79,85,01,A9,0C,85,02,20,E0,33,4C,99,34
67 DATAA0,00,81,01,49,00,91,01,60,20,90,34,20,E4,FF,F0
68 DATAF8,A2,06,0D,AE,34,F0,0D,CA,10,F8,4C,9C,34,9D,1D
69 DATA91,11,20,14,0D,20,90,34,8A,C9,00,00,02,A2,FE,C9
70 DATA01,00,02,A2,01,C9,02,00,02,A2,07,C9,03,00,02,A2
71 DATA28,C9,04,00,02,A2,01,C9,05,00,02,A2,01,C9,06,00
72 DATA0C,20,8D,35,20,86,37,A9,93,20,02,FF,60,A8,8A,30
73 DATA28,85,03,C0,85,F0,29,C0,04,00,06,A9,A0,00,91
74 DATAD1,A5,01,18,65,03,85,01,A5,02,69,00,85,02,A0,00
75 DATAB1,01,C9,66,F0,13,4C,3E,35,49,FF,85,03,4C,29,35
76 DATAA9,20,A0,00,91,01,4C,01,35,A5,01,38,E5,03,85,01
77 DATAA5,02,E9,00,85,02,A0,00,81,01,C9,66,F0,C3,A9,00
78 DATA85,07,A9,79,85,04,A9,0C,85,05,A2,00,A0,00,A9,00
79 DATA85,06,85,08,B1,04,18,69,80,26,06,C9,E6,08,A5,09
80 DATAC9,08,90,EE,A5,06,9D,C0,3B,8A,13,69,08,AA,81,04
81 DATAC9,66,00,DA,A5,04,18,69,28,85,04,A5,05,69,00,85
82 DATAD5,E6,07,A6,07,8A,C9,08,90,C2,4C,99,34,A2,17,8D
83 DATAC0,3B,9D,00,3C,CA,10,F7,60,8D,00,3C,30,0B,A9,00
84 DATA8D,FF,33,8D,00,3C,29,7F,60,A9,FF,8D,FF,33,4C,A3
85 DATA35,B1,E8,4D,FF,33,11,E6,60,20,84,9D,8A,4A,F0,03
86 DATA4C,1C,99,E0,01,00,06,A9,38,8D,60,3F,60,A9,60,4C
87 DATAC9,35,53,50,52,49,54,C5,53,50,52,4F,CE,53,50,52

```



88 DATA4F,46,C6,53,50,52,43,4F,CC,45,44,49,04,57,4F,52
89 DATA44,D3,4B,49,4C,CC,53,43,52,45,45,CE,00,00,00,00
90 DATA79,A9,57,8D,12,03,A9,36,8D,13,03,A9,00,8D,56,36
91 DATA58,20,4F,FF,93,11,20,20,20,53,50,52,49,54,45,20
92 DATA42,41,53,49,43,20,56,34,2E,30,20,00,20,71,36,20
93 DATA4F,FF,11,20,20,20,20,20,20,20,20,20,46,4F,52,20
94 DATA48,45,4C,50,20,54,59,50,45,20,27,57,4F,52,44,53
95 DATA27,11,00,4C,9D,36,00,AD,12,FF,29,03,09,C0,8D,12
96 DATAFF,A9,39,8D,13,FF,AD,56,36,F0,03,4C,88,3D,4C,42
97 DATACE,A5,38,C9,33,90,00,A9,FD,95,38,A9,00,95,37,A5
98 DATA37,38,E5,2B,AA,A5,38,E5,2C,20,5F,A4,20,4F,FF,20
99 DATA42,59,54,45,53,20,46,52,45,45,00,00,00,A2,05,8D
100 DATAA9,36,9D,0C,03,CA,10,F7,60,AF,36,CA,36,08,36,C9
101 DATA40,90,11,C9,5B,00,00,A9,35,A0,D2,20,07,8A,10,04
102 DATAC8,4C,D4,89,20,79,04,4C,6A,89,AA,04,49,A0,35,04
103 DATA23,A0,D2,84,22,4C,9E,00,49,00,0A,08,09,00,37,48
104 DATA89,0A,37,48,4C,73,04,00,00,00,00,00,00,00,00
105 DATA00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
106 DATA00,00,00,00,00,00,00,00,00,00,00,00,C3,37,72,37,85,37
107 DATA93,37,FF,33,9A,37,19,37,B8,35,A2,05,00,6D,37,9D
108 DATA9C,03,CA,10,F7,20,85,3F,A2,00,0D,CC,00,20,D2,FF
109 DATAF0,03,E8,D0,F5,A9,04,8D,12,FF,A9,D1,8D,13,FF,58
110 DATA20,71,36,20,4F,FF,11,20,20,20,20,53,50,49,54,45
111 DATA20,42,41,53,49,43,20,52,45,53,54,41,52,54,20,3A
112 DATA20,53,59,53,20,31,33,32,39,35,11,00,60,6A,89,88
113 DATA8B,00,8C,AD,56,36,00,10,78,20,A2,3E,A9,01,8D,56
114 DATA36,20,37,3F,58,60,AD,56,36,F0,00,20,3F,3E,A9,00
115 DATA8D,56,36,60,20,84,9D,8E,EB,3C,60,A2,00,A0,00,0D
116 DATAD2,35,F0,EF,10,16,29,7F,20,D2,FF,E8,A9,20,20,D2
117 DATAFF,C9,98,29,09,C9,09,00,F3,4C,9D,37,20,D2,FF,E8
118 DATAC8,4C,9F,37,20,D2,9D,A5,15,F0,09,4A,00,19,A5,14
119 DATAC9,20,00,13,0A,C9,00,00,0E,0E,EF,3C,A5,14,8D,EE
120 DATA3C,A5,15,8D,ED,3C,60,4C,1C,99,FF,FF,00,00,FF,FF
121 DATA00,00,FF,FF,00,00,FF,FF,00,00,FF,FF,00,00,FF,FF
122 DATA3C,66,6E,6E,60,62,3C,00,10,3C,66,7E,66,66,66,00
123 DATA7C,66,66,7C,66,66,7C,00,3C,66,60,60,60,66,3C,00
124 DATA79,6C,66,66,66,6C,78,00,7E,60,60,78,60,60,7E,00
125 DATA7E,60,60,78,60,60,60,00,3C,66,60,6E,66,66,3C,00
126 DATA6E,66,66,7E,66,66,66,00,3C,10,10,10,10,10,3C,00
127 DATA1E,0C,0C,0C,0C,6C,38,00,66,6C,78,70,78,6C,66,00
128 DATA60,60,60,60,60,60,7E,00,63,77,7F,60,63,63,63,00
129 DATA66,76,7E,7E,6E,66,66,00,3C,66,66,66,66,66,3C,00
130 DATA7C,66,66,7C,60,60,60,00,3C,66,66,66,66,3C,0E,00
131 DATA7C,66,66,7C,78,6C,66,00,3C,66,60,3C,06,66,3C,00
132 DATA7E,10,10,10,10,10,10,00,66,66,66,66,66,66,3C,00
133 DATA66,66,66,66,66,66,3C,10,00,63,63,63,60,7F,77,63,00
134 DATA66,66,3C,10,3C,66,66,00,66,66,66,3C,10,10,10,00
135 DATA7E,06,0C,10,30,60,7E,00,3C,30,30,30,30,30,3C,00
136 DATA0E,12,30,7C,30,62,FC,00,3C,0C,0C,0C,0C,0C,3C,00
137 DATA00,10,3C,7E,10,10,10,10,00,10,30,7F,7F,30,10,00
138 DATA00,00,00,00,00,00,00,00,10,10,10,10,00,00,10,00
139 DATA66,66,66,00,00,00,00,00,66,66,FF,66,FF,66,66,00
140 DATA19,3E,60,3C,06,7C,10,00,62,66,0C,10,30,66,46,00
141 DATA3C,66,3C,38,67,66,3F,00,06,0C,10,00,00,00,00,00
142 DATA0C,10,30,30,30,10,0C,00,30,10,0C,0C,0C,10,30,00
143 DATA00,66,3C,FF,3C,66,00,00,00,10,10,7E,10,10,00,00
144 DATA00,00,00,00,00,10,10,00,00,00,7E,00,00,00,00,00
145 DATA00,00,00,00,00,10,10,00,00,00,03,06,0C,10,30,60,00
146 DATA3C,66,6E,76,66,66,3C,00,10,10,30,10,10,10,7E,00
147 DATA3C,66,06,0C,30,60,7E,00,3C,66,06,1C,06,66,3C,00
148 DATA06,0E,1E,66,7F,06,06,00,7E,60,7C,06,06,66,3C,00
149 DATA3C,66,60,7C,66,66,3C,00,7E,66,0C,10,10,10,10,00
150 DATA3C,66,66,3C,66,66,3C,00,3C,66,66,3E,06,66,3C,00
151 DATA00,00,10,00,00,10,00,00,00,00,10,00,00,10,10,30
152 DATA0E,10,30,60,30,10,0E,00,00,00,7E,00,7E,00,00,00
153 DATA70,10,0C,06,0C,10,70,00,3C,66,06,0C,10,00,10,00
154 DATA00,00,00,FF,FF,00,00,00,00,1C,3E,7F,7F,1C,3E,00
155 DATA10,10,10,10,10,10,10,10,00,00,00,FF,FF,00,00,00
156 DATA00,00,FF,FF,00,00,00,00,00,FF,FF,00,00,00,00,00
157 DATA00,00,00,00,FF,FF,00,00,30,30,30,30,30,30,30,30
158 DATA0C,0C,0C,0C,0C,0C,0C,0C,00,00,00,00,E0,F0,30,10,10
159 DATA10,10,1C,0F,07,00,00,00,10,10,38,F0,E0,00,00,00
160 DATAC0,C0,C0,C0,C0,C0,FF,FF,C0,E0,70,30,1C,0E,07,03
161 DATA03,07,0E,1C,38,70,E0,C0,FF,FF,C0,C0,C0,C0,C0,C0
162 DATAFF,FF,03,03,03,03,03,03,00,3C,7E,7E,7E,7E,3C,00





```

163 DATA00,00,00,00,00,FF,FF,00,36,7F,7F,7F,3E,1C,00,00
164 DATA00,60,60,60,60,60,60,60,00,00,00,07,0F,1C,18,18
165 DATA03,E7,7E,3C,3C,7E,E7,C3,00,3C,7E,66,66,7E,3C,00
166 DATA18,18,66,66,18,18,3C,00,06,06,06,06,06,06,06,06
167 DATA00,1C,3E,7F,3E,1C,00,00,18,18,18,FF,FF,18,18,18
168 DATA00,C0,30,30,C0,C0,30,30,18,18,18,18,18,18,18,18
169 DATA00,00,03,3E,76,36,36,00,FF,7F,3F,1F,0F,07,03,01
170 DATA00,00,00,00,00,00,00,00,00,F0,F0,F0,F0,F0,F0,F0,F0
171 DATA00,00,00,00,FF,FF,FF,FF,FF,FF,00,00,00,00,00,00
172 DATA00,00,00,00,00,00,00,00,FF,C0,C0,C0,C0,C0,C0,C0
173 DATA00,CC,33,33,CC,CC,33,33,03,03,03,03,03,03,03,03
174 DATA00,00,00,00,CC,CC,33,33,FF,FE,FC,F8,F8,E0,C0,00
175 DATA03,03,03,03,03,03,03,03,18,18,18,1F,1F,18,18,18
176 DATA00,00,00,00,0F,0F,0F,0F,18,18,18,1F,1F,00,00,00
177 DATA00,00,00,00,F8,F8,18,18,18,00,00,00,00,FF,FF
178 DATA00,00,00,1F,1F,18,18,18,18,18,FF,FF,00,00,00
179 DATA00,00,00,FF,FF,18,18,18,18,18,18,F8,F8,18,18,18
180 DATA00,C0,C0,C0,C0,C0,C0,C0,E0,E0,E0,E0,E0,E0,E0,E0
181 DATA07,07,07,07,07,07,07,07,FF,FF,00,00,00,00,00,00
182 DATA3E,E1,DE,D8,D8,DE,E1,3E,FD,02,FD,CD,CD,FD,02,FD
183 DATAFC,07,FB,9B,FB,07,3E,F8,00,00,00,00,00,00,00,00
184 DATA00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
185 DATA00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
186 DATA3E,E1,DE,D8,D8,DE,E1,3E,FD,02,FD,CD,CD,FD,02,FD
187 DATAFC,07,FB,9B,FB,07,3E,F8,00,00,00,00,00,00,00,00
188 DATA00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
189 DATA00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
190 DATA00,00,00,00,01,00,00,00,00,00,00,00,00,00,00,00
191 DATA00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
192 DATA00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
193 DATA00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
194 DATAFF,E1,DE,D8,D8,DE,E1,FF,FF,02,FD,CD,CD,FD,02,FF
195 DATAFF,07,FB,9B,FB,07,07,FF,00,00,00,00,00,00,00,00,00
196 DATA00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
197 DATA00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
198 DATA00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
199 DATA20,20,20,20,20,20,20,20,10,10,10,10,10,10,10,10
200 DATA00,00,00,00,00,00,00,00,00,00,00,00,01,01,20,20
201 DATA42,CE,24,04,24,04,20,20,00,10,18,20,18,03,00,00
202 DATA82,40,A9,00,9D,7F,3C,CA,D0,FA,8D,FD,3C,AC,FF,3C
203 DATA8D,00,3C,99,00,3C,C8,E8,9C,FC,3C,AC,FD,3C,8A,09
204 DATAF8,3C,09,AC,FC,3C,28,F8,19,8E,FC,3C,AE,FD,3C,90
205 DATA0D,F8,3C,08,AE,FC,3C,28,D0,D6,98,18,69,18,A8,18
206 DATA90,CE,8E,FC,3C,98,CD,FC,3C,F0,05,98,38,E9,18,A8
207 DATAEE,FD,3C,AD,FD,3C,C9,03,D0,B6,AC,FE,3C,F0,20,A2
208 DATA07,5E,00,3C,7E,88,3C,7E,90,3C,7E,98,3C,5E,00,3C
209 DATA7E,A8,3C,7E,B0,3C,7E,B8,3C,CA,10,E5,88,D0,E0,60
210 DATAAD,ED,3C,29,01,8D,ED,3C,CD,EC,3C,D0,13,AD,EE,3C
211 DATA0D,F6,3C,D0,0B,AD,EF,3C,CD,F7,3C,D0,03,6C,F0,3C
212 DATAAD,EE,3C,29,07,8D,FE,3C,AD,EF,3C,29,07,8D,FF,3C
213 DATAAD,ED,3C,4A,AD,EE,3C,6A,4A,4A,8D,F2,3C,AD,EF,3C
214 DATA4A,4A,4A,8D,F3,3C,CD,F5,3C,D0,08,AD,F2,3C,CD,F4
215 DATA3C,F0,09,20,3F,3E,20,A2,3E,20,37,3F,AD,ED,3C,AE
216 DATAEE,3C,AC,EF,3C,8D,EC,3C,8E,F6,3C,8C,F7,3C,AD,F2
217 DATA3C,8D,F4,3C,AD,F3,3C,8D,F5,3C,20,00,3D,A2,07,A9
218 DATA00,85,E9,20,99,35,0A,26,E9,0A,26,E9,0A,26,E9,85
219 DATAE8,A5,E9,69,38,85,E9,8A,0A,0A,0A,69,80,85,E6,A9
220 DATA3C,85,E7,A0,07,EA,20,B1,35,91,E6,88,10,F7,CA,10
221 DATA0E,A2,3F,BD,80,3C,9D,C0,30,CA,10,F7,6C,F0,3C,AC
222 DATAF5,3C,B9,05,3F,18,6D,F4,3C,85,E6,B9,1E,3F,69,0C
223 DATA85,E7,A0,03,B9,D0,3C,91,E6,88,10,F8,18,A5,E6,69
224 DATA28,85,E6,A5,E7,69,00,85,E7,A0,03,B9,D4,3C,91,E6
225 DATA88,10,F8,38,A5,E6,E9,28,85,E6,A5,E7,E9,04,85,E7
226 DATA00,03,B9,D8,3C,91,E6,88,10,F8,18,A5,E6,69,28,85
227 DATAEE,A5,E7,69,00,85,E7,A0,03,B9,DC,3C,91,E6,88,10
228 DATAF8,60,AC,F3,3C,B9,05,3F,18,6D,F2,3C,85,E6,89,1E
229 DATA3F,69,0C,85,E7,A0,03,B1,E6,99,D0,3C,88,10,F8,18
230 DATAA5,E6,69,28,85,E6,A5,E7,69,00,85,E7,A0,03,B1,E6
231 DATA99,D4,3C,88,10,F8,38,A5,E6,E9,28,85,E6,A5,E7,E9
232 DATA04,85,E7,A0,03,B1,E6,99,D0,3C,88,10,F8,18,A5,E6
233 DATA69,28,85,E6,A5,E7,69,00,85,E7,A0,03,B1,E6,99,DC
234 DATA3C,88,10,F8,60,00,28,50,78,A0,C8,F8,18,40,60,90
235 DATA8E,00,00,30,58,00,A8,D0,F8,20,40,70,90,C0,00
236 DATA00,00,00,00,00,01,01,01,01,01,01,02,02,02,02,02
237 DATA02,02,03,03,03,03,03,AC,F3,3C,B9,05,3F,18,6D,F2

```



```

238 DATA3C,85,E6,89,1E,3F,69,0C,85,E7,A0,83,89,9A,3F,91
239 DATAE6,88,10,F8,18,A5,E6,69,78,85,E6,A5,E7,69,00,85
240 DATAE7,A0,83,89,9E,3F,91,E6,88,10,F8,60,A5,E6,E9,28
241 DATA85,E6,A5,E7,E9,04,05,E7,A0,83,AD,EB,3C,91,E6,88
242 DATA10,F8,18,A5,E6,69,28,85,E6,A5,E7,69,00,85,E7,A0
243 DATA03,AD,EB,3C,91,E6,88,10,F8,60,78,79,7A,7B,7C,7D
244 DATA7E,7F,78,A9,80,8D,14,03,A9,3D,8D,15,03,20,A2,3E
245 DATA20,37,3F,58,60,78,A9,0E,8D,14,03,A9,CE,8D,15,03
246 DATA20,3F,3E,58,60,00,00,00,00,00,00,00,00,00,00
247 DATA00,53,50,CF,31,35,35,39,20,34,02,FF,0D,01,FF,53
248 DATA59,53,31,36,33,30,39,08,FF,0D,01,FF,53,59,53,31
249 DATA39,32,30,35,04,FF,8D,3E,FF,4C,A4,F2,48,03,00,FD
250 DATA00
    
```

READY.

COLOR

Ezzel a programmal azt a sokak által ismert problémát tudjuk kiküszöbölni, miszerint közvetlenül CLI-ből nem állíthatjuk a képernyő színeit.

Az utasítás formátuma:

COLOR szín piros zöld kék

Ahol a

szín — (0–31) az átírandó szín-regiszter száma

piros — (0–15) a szín RGB kódjából a piros

zöld — (0–15) a szín RGB kódjából a zöld

kék — (0–15) a szín RGB kódjából a kék.

```
/*
```

```
    COLOR col r g b
```

```
    AztecC: cc +l <file>
```

```
            ln <file>.o -lc32 -lm -o
```

```
    Code by TPE
```

```
*/
```

```
#include <functions.h>
```

```
#include <exec/types.h>
```

```
#include <intuition/intuition.h>
```

```
#include <intuition/intuitionbase.h>
```

```
struct IntuitionBase *IntuitionBase;
```

```
struct GfxBase *GfxBase;
```

```
struct ViewPort *VP;
```

```
main(argc,argv)
```

```
int     argc;
```

```
char    *argv[];
```

```

o {
o     int     col,r,g,b;
o
o     if(argc != 5)
o     {
o         printf("USAGE: COLOR col red green blue\n");
o         exit();
o     }
o
o     if(!(IntuitionBase=(struct IntuitionBase *)
o         OpenLibrary("intuition.library",0))) exit(FALSE);
o
o     if(!(GfxBase=(struct GfxBase *)
o         OpenLibrary("graphics.library",0)))
o     {
o         CloseLibrary(IntuitionBase);
o         exit(FALSE);
o     }
o
o     VP = &(IntuitionBase->ActiveScreen->ViewPort);
o
o     col= atoi(argv[1]);
o     r  = atoi(argv[2]);
o     g  = atoi(argv[3]);
o     b  = atoi(argv[4]);
o
o     SetRGB4(VP,col,r,g,b);
o
o     CloseLibrary(IntuitionBase);
o     CloseLibrary(GfxBase);
o     exit(TRUE);
o }
    
```

MOUSE

Ez a program képes arra, hogy a sokszor igen zavaró egér képét eltüntesse a képernyőről

Formátuma

MOUSE <ON/OFF>

Az OFF kiterjesztéssel ki, míg az ON-nal értelemszerűen bekapcsolhatjuk az egeret.

```

o
o     /*
o
o         MOUSE <on/off>
o
o         AztecC:  cc +l <file>
o                 ln <file>.o -lc32 -lm -o
o
o                 Code by TPE
o
o                 */
o
o     #include <functions.h>
o     #include <exec/types.h>
o     #include <intuition/intuition.h>
o     #include <intuition/intuitionbase.h>
    
```



```

O struct IntuitionBase *IntuitionBase;
O struct Window *Window;
O
O UWORD *mouse;
O
O main(ArgC,ArgV)
O
O int ArgC;
O char *ArgV[];
O {
O     if(ArgC !=2){
O         printf("USAGE: MOUSE <on/off>\n");
O         exit(FALSE);
O     }
O
O     if(!(IntuitionBase=(struct IntuitionBase *)
O         OpenLibrary("intuition.library",0))) exit(FALSE);
O
O     Window=IntuitionBase->ActiveWindow;
O
O     if(!(strcmp(ArgV[1],"on"))) M_on();
O     if(!(strcmp(ArgV[1],"off"))) M_off();
O
O     CloseLibrary(IntuitionBase);
O }
O
O M_on()
O {
O     ClearPointer(Window);
O }
O
O M_off()
O {
O     mouse=AllocMem(8L,0x00010003);
O     SetPointer(Window,mouse,0L,0L,0L,0L);
O }

```

ECHO+

Ez a program tulajdonképpen a hagyományos echo DOS utasításnak egy továbbfejlesztett változata.

Használatával képesek leszünk a kívánt feliratot vastagon (bold), dőlten (italic) és aláhúzva (underlined) megjeleníteni. Természetesen ezek az opciók összefűzhetők egy „+” jellel (bold+italic, italic+underlined...).

AMIGA

```

O /*
O
O     ECHO+ <style> <string>
O
O     AztecC: cc +l <file>
O             ln <file>.o -lc32 -lm -o
O
O     Code by TPE

```

```

                                                                    */
○ #include <exec/types.h> ○
○ #define bold          printf("\033[1m\033[1m\033[11m"); ○
  #define italic       printf("\033[3m\033[3m\033[21m"); ○
○ #define underlined   printf("\033[4m\033[4m\033[55m"); ○

○ main(ArgC,ArgV) ○
○ int      ArgC; ○
  char    *ArgV[]; ○
○ { ○
  if(ArgC !=3){ ○
○ printf("USAGE: ECHO+ <style> <string>\n"); ○
  printf(" Styles: bold, italic, underlined, bold+italic\n"); ○
  printf("          bold+italic+underlined, italic+underlined\n\n") ○
  exit(FALSE); ○
○ } ○

○ if(!(strcmp(ArgV[1],"bold"))) ○
  bold; ○

○ if(!(strcmp(ArgV[1],"italic"))) ○
  italic; ○

○ if(!(strcmp(ArgV[1],"underlined"))) ○
  underlined; ○

○ if(!(strcmp(ArgV[1],"bold+italic"))){ ○
  bold;italic;} ○

○ if(!(strcmp(ArgV[1],"bold+underlined"))){ ○
  bold;underlined;} ○

○ if(!(strcmp(ArgV[1],"bold+italic+underlined"))){ ○
  bold;italic;underlined;} ○

○ if(!(strcmp(ArgV[1],"italic+underlined"))){ ○
  italic;underlined;} ○

○ printf("%s\033[0m\n",ArgV[2]); ○
○ }
    
```

C-64 II. + fényceruza + hangdigitalizáló + 15 db játékkazetta (530) program + 10 szakkönyv (9999 Ft) eladó! Tel.: 1683-835.

C-64-re lemezre és kazettára játékok eladók. Válaszborítékért listát küldünk. 7400 Kaposvár, TBS Herman O. 60.

1541 II. drive + 100 db lemez + tartó eladó 16 000 Ft-ért. Kiss György, 6000 Kecskemét, Bajza u. 13. IV/12.

Keresem a Master base program magyar vagy angol leírását. Kalló János, 6400 Kiskunhalas, Dózsa György u. 3.

Keresem az MPS 1250 típusú nyomtató gépkönyvét magyar vagy angol nyelven. Kalló János, 6400 Kiskunhalas, Dózsa György u. 3.

C-64-es programokat cserélnék lemezen, kazettán. Jernei Marianna, 1209 Budapest, Majosház u. 2.

C-64-re programok eladása minden mennyiségben lemezen, kazettán. Utántöltős programok olcsó áron. Kérj listát, információt. Deutsch Szabolcs, 8749 Zalakaros, Fő u. 19.

C-64-re új játék- és felhasználói programok eladók. Bélyeggel ellátott válaszborítékért listát küldünk. (Vízhatlan csomagolás, kedvezmények!) FRIEND TWO CREW, 1399 Bp. Pf.: 701/55.

5000 szavas szótárdisk, bővíthető szókészlettel - Angol I-II, Német I-II, Themen I-II, C-64-re. Az egyes változatok ára: 950 Ft. Megrendelhető: válaszborítékért részletes információ kérhető: SULI-SOFT 1327 Újpest 3, Pf.: 91.

PRINTFOX EXTENSI, CHARAKTER FOX, OLD-FOX, LUPE FOX programokat keres lemezen Kövesdi Tibor, 3700 Kazincbarcika, Szegfű u. 14.

Változó lista

Egy terjedelmes BASIC program megírása a csak röpké szabad idejét erre fordító amatőrnek hosszú időt vesz igénybe, de ettől függetlenül is sokszor előfordul, hogy régebbi programunk kiegészítésre szorul. Gondot jelent ilyenkor a hosszú programlistából kikeresni, milyen változókat használtunk. Ezen a problémán segít az általam írt gépi kódú program.

Indítási címe \$C400 (50176), de az „előtte” lévő \$C000—\$C3FF (49152—50175) tárterületet adattárolásra felhasználja. (Az esetleg ott lévő gépi kódú programot törli.)

A program több részből áll, az első rész az inicializálást végzi, tehát törli a \$C000—\$C3FF területet, a \$C000 értékét cím formátumban elhelyezi a \$FB—\$FC (251—252) tárcímen, beolvassa a BASIC program kezdőcímét és azt beírja a programszámlálóba.

A program második része bytonként végigolvassa a BASIC programot. A mindenkori láncolási címet, a \$C3FA—\$C3FB címen, míg a BASIC sorszámot a \$FD—\$FE címen tárolja. Mivel a CHGET (\$73) rutin a szóközt (\$20) átlépi, ezt a tulajdonságot a láncolási cím és a sorszám olvasásakor meg kell szüntetni, hogy a programlista megcsonkítását elkerüljük. Megoldás (lásd 1. lista) a KI subroutine, ami a feltételes elágazás relatív címét 0-ra állítja (0-t ír a \$83 címre), így a feltétel teljesülésétől függetlenül fut le a CHGET, ill. CHGOT subroutine. Természetesen ezen négy byte beolvasása után vissza kell állítani az eredeti állapotot, amit a BE subroutine végez el.

Ezután ellenőrzi a beolvasott bytot, hogy változó-e (betűt keres, mert minden változónév betűvel kezdődik) ehhez a \$B113 operációs rendszer-rutint használja, közben figyel, hogy a programlista véget ért-e (csak a láncolási cím magas byte-ját figyel), a BASIC-sor véget ért-e. Keresi a REM utasítást (ha talál, új sort olvas, ekkor használja a tárolt láncolási címet), az idézőjelet (az idézőjelek között nem keres változót), a DATA utasítást (a következő kettőspontig vagy sorvégig terjedő részt változókeresés nélkül átolvassa). Itt kell megjegyezni, programom nincs felkészítve le nem zárt idézőjel kezelésére! A fenti „szűrőn” átjutó adatokat (a változót jelentő betű kivételével) a 2-es tárcímen tárolja. Ennek célja az FN utasítás

és az E formátummal megadott számok felismerésének biztosítása.

Ha betűt talál, a 3. főrésze ugrik a program. Az átmeneti tároló első helyén (\$C3FC) tárolja a változó betűjének ASCII értékét. Ezt követően keresi az FN utasítást (ha talál, azt külön jelzi a változó típusában), majd vizsgálja, hogy a talált betű E-betű-e (ha nem, törli az ennek is jelzésére használt 2-es tárcímet). Ezt követően keresi a változónév 2. karakterét (betű vagy szám), ennek ASCII értékét tárolja a \$C3FD címen (ha nincs 2. karakter, itt 0 áll).

Fontos! Bizonyos — emberi — hibalehetőségeket megelőzendő, SYNTAX ERROR kiírással leáll a program és nem ír ki változót, ha a második karakter szám és azt a szám vagy betű követi, vagy a változónévben máshol szám előfordul.

A többi betűből álló változónevek többi („fölsleges”) karakterét átolvassa. A változónév után keresi az egész típusú, ill. a szövegváltozó jelét (% , ill. \$), ha ilyen van, azt tárolja a megfelelő ASCII érték formájában a \$C3FE címen, valószínűleg esetén itt 0 áll.

Ezután a tömböt jelentő nyitó zárójel keresi, ha ilyen van, tárolja a \$C3FF címen.

A fenti eljárás eredményeként 4 egymás utáni tárcacímen rendelkezésünkre áll az aktuális változó minden jellemzője.

Most ellenőrzi a 2-es címet, ha nincs törölve (azaz egyedül álló E betűt talált), a CHGET rutin megfelelő részével ellenőrzi, hogy az E betű előtt (ez ugyanis ekkor a 2-es tárcím tartalma) szám áll-e, ha igen, az E betű a kitevős formában megadott számérték része, nem változót jelent, változóként nem veszi figyelembe, új változót keres.

A program 4. része az így tárolt változóadatokat a nyilvántartó tármezőben (\$3FC—\$C3C1).

Itt szólni kell ennek technikájáról: az átmeneti tárolóban (\$C3FC—\$C3FF) ASCII értékek szerepeltek, a nyilvántartó tárban a változó első és második karakterének értéke címe konvertálva jelenik meg (azaz az adott tárterület minden egyes címe megfelel egy-egy betű-betű, betű-szám, ill. betű-üres hely kombinációjának), míg a tárcímre írt érték a változó típusának és esetleges tömb formájának, mégpedig bitenként, így egy címen megkülönböztethetően tárolható a 6 le-

hetséges változótípus és a függvény is. Az aktuális változó adatait a korábbiakban megtaláltakkal az ORA utasítás egyesíti.

Programunk a BASIC program olvasását befejezve az 5. részére ugrik, a nyilvántartó tárterületet sorban végigvizsgálja, ahol 0-tól különböző értéket talál, a cím-konverziót visszafelé elvégzi. A változó első betűjének ASCII értéke az X-regiszterben, a második karakteré az Y-regiszterben jelenik meg. A különböző változótípusok felismeréséhez az aktuális tárcím értékét a 2-es címre átírja, mert az LSR utasítás a nulláslapon közvetlenül tudja kezelni. A C kapcsolótól függően írja ki az egyes változótípusokat. A tömböket inverzen írja ki, az FN utasítás betűjét (ill. betű második karaktert) szintén, mégpedig a jobb figyelemfelhívás érdekében külön sorban.

A 21. képernyősor után megáll a kiírás, ill. akkor is, ha véget ért a kiírás. Tovább lépés bármely billentyű megnyomása és *elengedése* után. (Így a nehéz kezűek keze alól sem szalad el a folytatás!)

A program hátralévő része a különböző subroutineokat tartalmazza, döntő többségük a kiíráshoz szükséges.

Megjegyzések. Ha a program a VÁLTOZÓ LISTA felirat után változót nem ír ki, csak sorszám nélküli SYNTAX ERROR-t, a \$FD—\$FE tárcacíméből kiolvasható azon sor száma, ahol az általam hibássá minősített nevű változó szerepel (1. a korábbi megjegyzést is!).

Ha hosszabb program után rövidebbet töltünk be, programom lefutása után (és a kiírás megtörténte után, ilyenkor a VÁLTOZÓ LISTA felirat törlődik a képernyőről) valamilyen sorszámmal jelzett SYNTAX ERROR IN... néha egyéb hibahüvely jelenik meg, olykor az előző, hosszabb program valahonnan folytatódik (a programszámlálóban lévő értéktől függően). Ezen jelenség nem valódi hibát jelent, a program szolgáltatását nem érinti.

Az ASSEMBLER lista a „Gépi kódú programozás C64-en” című könyvben közölt (és működőképesé tett...!) ASSEMBLER programra készült, annak szintaxisára. Vigaszul: a BASIC betöltőt is rendelkezésre bocsátottam. Indítás: SYS 50176. Jó munkát!

Baráti üdvözléssel:

Dr. Kunszenti Tamás

```

O      0073          10 CHGET = #73
O      0079          20 CHGOT = #79
O      007C          25 CHG = #7C
O      B113          30 BETUE = #B113
O      AF08          40 ERROR = #AF08
O      FF02          50 KILLR = #FF02
O      C3F9          55 JELZO = #C3F9
O      C3FA          60 KOVA = #C3FA
O      C3FB          70 KOVF = #C3FB
O      C3FC          80 BETU1 = #C3FC
O      C3FD          90 BETU2 = #C3FD
O      C3FE          100 TIPUS = #C3FE
O      C3FF          110 TOMB = #C3FF
O      00CB          120 LENY = #CB
O      00D6          130 SOR = #D6
O      00D3          135 OSZ = #D3
O      E544          140 CLR = #E544
O      E56C          145 POZ = #E56C
O      C400          150 *= #C400
O      C400          20 44 E5 160 JSR CLR
O      C403          20 00 C6 162 JSR IRCIM
O      C406          20 95 C6 165 JSR CIMBE
O      C409          A2 04 170 LDX #4
O      C40B          A0 FF 180 TOR2 LDY #FFF
O      C40D          91 FB 190 TOR1 STA (*FB),Y
O      C40F          88 200 DEY
O      C410          D0 FB 210 BNE TOR1
O      C412          91 FB 220 STA (*FB),Y
O      C414          E6 FC 230 INC #FC
O      C416          CA 240 DEK
O      C417          D0 F2 250 BNE TOR2 ;MESTORTENT A MUNKATERULET TORL
ESE
O      C419          A5 2B 260 LDA #*2B ;43
O      C41B          A6 2C 270 LDX #*2C ;44
O      C41D          85 7A 280 STA #*7A ;A BASIC PR. KEZDETE
O      C41F          86 7B 290 STX #*7B ;BETOLTESE
O      C421          20 95 C6 300 JSR CIMBE
O      C424          20 9E C6 310 UJSOR JSR KI
O      C427          20 79 00 315 JSR CHGOT
O      C42A          8D FA C3 320 USOR STA KOVA ;KOVETO CIM ALSO BYTE
O      C42D          20 73 00 330 JSR CHGET
O      C430          8D FB C3 340 STA KOVF ;FELSO BYTE
O      C433          C9 00 350 CMP #0
O      C435          D0 06 360 BNE TOWA
O      C437          20 A3 C6 365 JSR BE
O      C43A          4C 57 C5 370 JMP VEGE ;HA A KOVETO CIM FELSO BYTJA=0,
H PROGRAM LISTA VEGET ERT
O      C43D          20 73 00 380 TOWA JSR CHGET ;ATOLVASSA A
O      C440          85 FD 390 STA #*FD ;BASIC SORSZAMOT
O      C442          20 73 00 400 JSR CHGET ;ES
O      C445          85 FE 410 STA #*FE ;ELTAROLJA
O      C447          20 A3 C6 415 JSR BE
O      C44A          20 90 C6 417 JSR TOR
O      C44D          20 73 00 420 UJ JSR CHGET ;BEOLVASSA A KOVETKEZO BYTOT
O      C450          C9 00 430 UJVAL CMP #0
O      C452          D0 09 440 BNE NSORV ;NEM SOR VEG
O      C454          20 9E C6 445 JSR KI
O      C457          20 73 00 450 JSR CHGET
O      C45A          4C 2A C4 460 JMP USOR
O      C45D          C9 8F 470 NSORV CMP #*8F ;REM?
O      C45F          D0 00 480 BNE NREM ;NEM REM
O      C461          A0 FA C3 490 LDA KOVA
O      C464          A6 FB C3 500 LDX KOVF
O      C467          85 7A 510 STA #*7A
O      C469          86 7B 520 STX #*7B
O      C46B          4C 24 C4 530 JMP UJSOR
O      C46E          C9 22 540 NREM CMP #*22 ;IDEZOJEL?
O      C470          D0 00 550 BNE NIDEZ ;NEM
O      C472          20 73 00 560 IDEZO JSR CHGET
O      C475          C9 22 570 CMP #*22
O      C477          D0 F9 580 BNE IDEZO ;HA NEM JOTT MEG A ZARO IDEZOJE
L
O      C479          20 90 C6 585 JSR TOR
O      C47C          4C 4D C4 590 JMP UJ
O      C47F          C9 83 591 NIDEZ CMP #*83 ;DATA?
O      C481          D0 0D 592 BNE NDATA ;NEM DATA UTASITAS
O      C483          20 73 00 593 ADAT JSR CHGET
O      C486          D0 FB 594 BNE AURI
O      C489          48 595 PHA
O      C48B          20 90 C6 596 JSR TOR
O      C48C          68 597 PLA
O      C48D          4C 50 C4 598 JMP UJVAL
O      C490          20 13 B1 600 NDATA JSR BETUE ;BETU? HA IGEN, C=1
O      C493          80 05 605 BCS VAL ;VALTOZO, C=1
O      C495          85 02 610 STA #2
O      C497          4C 4D C4 615 JMP UJ
O      C49A          8D FC C3 620 VAL STA BETU1
O      C49D          A6 02 621 LDX #2
O      C49F          E0 A5 622 CPX #*A5 ;FN
O      C4A1          D0 03 623 BNE NEMFN
O      C4A3          8E FE C3 624 STX TIPUS
O      C4A6          C9 45 625 NEMFN CMP #*45 ;E
O      C4A8          F0 03 626 BEQ E
O      C4AA          20 90 C6 628 JSR TOR

```



0	C480	20	73	00	630	E	JSR	CHGET		
	C480	20	05		640		BCC	BEIR		;C=0,AZAZ SZAM
0	C482	20	13	B1	650		JSR	BETUE		
	C485	20	1F		660		BCC	TYPUS		;C=0,NEM BETU [ES NEM SZAM]
0	C487	20	FD	C3	670	BEIR	STA	BETU2		
	C48A	20	90	C6	675		JSR	TOR		
0	C48D	20	00		680		BCC	BETU		;A MASODIK ELEM BETU
	C48F	20	73	00	690		JSR	CHGET		;A MASODIK ELEM SZAM
0	C4C2	20	05		700		BCC	HIBA		;NEM LEHET KETJEGYU SZAM
	C4C4	20	13	B1	710		JSR	BETUE		;HA NEM BETU, HIBA
0	C4C7	20	00		720		BCC	TYPUS		;NEM SZAM, NEM BETU
	C4C9	40	08	AF	730	HIBA	JMP	ERROR		
0	C4CC	20	73	00	740	BETU	JSR	CHGET		
	C4CF	20	F8		750		BCC	HIBA		
0	C4D1	20	13	B1	760		JSR	BETUE		;BETU?
	C4D4	20	F6		770		BCC	BETU		;BETU
0	C4D6	09	25		780	TYPUS	CMF	#425		;K
	C4D8	F0	04		790		BEQ	TYPBE		;UGRIK A TIPUSBEIRASRA
0	C4DA	09	24		800		CMF	#424		;#
	C4DC	08	09		810		BNE	LEB		;VALOS
0	C4DE	20	FE	C3	820	TYPBE	STA	TIPUS		
	C4E1	20	90	C6	825		JSR	TOR		
0	C4E4	20	73	00	830		JSR	CHGET		
	C4E7	09	28		840	LEB	CMF	#428		;C
0	C4E9	08	06		850		BNE	FELD		;NEM TOMB
	C4EB	20	FF	C3	860		STA	TOMB		
0	C4EE	20	90	C6	865		JSR	TOR		
	C4F1	F5	02		870	FELD	LDA	#2		
0	C4F3	F0	08		871		BEQ	FELD2		;TOROLT 2-ES CIM
	C4F5	20	7C	00	872		JSR	CHS		;SZAM?, HA IGEN, C=0
0	C4F8	20	06		873		BCC	FELD2		;C=1,NEM SZAM ALL AZ E ELOTT
	C4FA	20	90	C6	875		JSR	TOR		
0	C4FD	40	40	C4	876		JMP	UJ		
	C500	20	FC	C3	878	FELD2	LDA	BETU1		
0	C503	20			880		SEC			;C=1
	C504	E9	41		890		SBC	#441		;A
0	C506	AA			900		TAX			
	C507	F0	0E		910		BEQ	NULLA		;AZ ELSO BETU 'A', NEM KELL SZO
0	ROZNI									
	C509	A5	F8		920		LDA	#4FB		
0	C50B	18			930	UJRA	CLC			
	C50C	69	25		940		ADC	#425		;37
0	C50E	90	02		950		BCC	NONAT		;NINCS ATVITEL
	C510	E6	FC		960		INC	#4FC		
0	C512	0A			970	NONAT	DEX			
	C513	08	F6		980		BNE	UJRA		
0	C515	85	F8		990		STA	#4FB		;BEALLITOTTA A SOR KEZDOCIMET
	C517	AC	FD	C3	1000	NULLA	LDY	BETU2		
0	C51A	F0	0D		1010		BEQ	CIM		;NINCS MASODIK ELEM
	C51C	98			1020		TYA			
0	C51D	20	13	B1	1030		JSR	BETUE		
	C520	08	03		1040		BCC	MBETU		;A MASODIK ELEM BETU
0	C522	18			1050		CLC			
	C523	69	07		1060		ADC	#7		
0	C525	38			1080	MBETU	SEC			
	C526	E9	36		1090		SBC	#436		;54
0	C528	A8			1100		TAY			
	C529	A9	01		1110	CIM	LDA	#1		
0	C52B	AE	FE	C3	1120		LDX	TIPUS		
	C52E	F0	0F		1130		BEQ	VS		;VALOS VALTOZO
0	C530	E0	A5		1132		CPX	#4A5		;FN
	C532	08	05		1134		BNE	NFN		;NEM FN
0	C534	A9	40		1136		LDA	#440		;#64
	C536	4C	47	C5	1138		JMP	NTB		
0	C539	0A			1140	NFN	ASL	A		;AKKU=2
	C53A	E0	24		1150		CPX	#424		;#
0	C53C	F0	01		1160		BEQ	VS		;SZOVEGVALTOZO
	C53E	0A			1170		ASL	A		;AKKU=4
0	C53F	AE	FF	C3	1180	VS	LDX	TOMB		
	C542	F0	03		1190		BEQ	NTB		;NEM TOMB
0	C544	0A			1200		ASL	A		;3 BITTEL
	C545	0A			1210		ASL	A		;ELORE, MERT
0	C546	0A			1220		ASL	A		;TOMB
	C547	11	F8		1230	NTB	ORA	(#FB),Y		;A JELENLEGI VALTOZO ADATAINAK
0	EGYESITESE A KORABBIEVAL									
	C549	91	F8		1240		STA	(#FB),Y		;AZ EREDMENYT VISSZAJRJA
0	C54B	20	95	C6	1250		JSR	CIMBE		;VISSZALLITJA A TABLAZAT INDUL
0	O CIMET									
	C54E	20	82	C6	1260		JSR	TORLO		
0	C551	20	79	00	1270		JSR	CHGET		
	C554	4C	50	C4	1280		JMP	UJVAL		;UJ VALTOZOT KERES
0	C557	A2	00		1300	VEGE	LDX	#0		
	C559	A0	00		1310	UJBET	LDY	#0		
0	C55B	B1	F8		1320	UJMAS	LDA	(#FB),Y		
	C55D	08	1E		1340		BNE	VALT		;VAN VALTOZO
0	C55F	C8			1350	VISSZ	INY			
	C560	08	25		1360		CPY	#425		;37
0	C562	08	F7		1370		BNE	UJMAS		
	C564	E8			1380		INX			
0	C565	E0	1A		1390		CPX	#41A		;26
	C567	08	06		1400		BNE	NKESZ		;NINCS VEGE A TABLAZATNAK
0	C569	20	5D	C6	1404		JSR	NLAF		
	C56C	4C	44	E5	1406		JMP	CLR		
0	C56F	A5	F8		1410	NKESZ	LDA	#4FB		



	C571	18		1420	CLC			
	C572	69	25	1430	ADC	##25		;37
	C574	90	02	1440	BCC	NINCS		;NINCS RTVITEL
	C576	E6	FC	1450	INC	##FC		
	C578	85	FB	1460	NINCS	STA	##FB	;A KOVETKEZO KEZOO BETU CIME
	C57A	4C	59	1470	JMP	UJBET		;UJ ELSO BETU
	C57D	85	02	1490	VALT	STA	##2	;MEGURZES
	C57F	46	02	1500	LSR	##2		;0.BIT A C-BEN
	C581	B0	1E	1510	BCS	VANV		;VAN VALOS VALTOZO
	C583	46	02	1520	VIV	LSR	##2	;1.BIT A C-BEN
	C585	B0	23	1530	BCS	VANS		;VAN #-VALTOZO
	C587	46	02	1540	VIS	LSR	##2	
	C589	B0	20	1550	BCS	VANEL		;VAN EGESZ TIPUSU VALT.
	C58B	46	02	1560	VIE	LSR	##2	
	C58D	B0	37	1570	BCS	VANVT		;VAN VALOS TOMB
	C58F	46	02	1580	VIVT	LSR	##2	
	C591	B0	3C	1590	BCS	VANST		;VAN #-TOMB
	C593	46	02	1600	VIST	LSR	##2	
	C595	B0	46	1610	BCS	VANET		;VAN #-TOMB
	C597	46	02	1620	VIET	LSR	##2	
	C599	B0	50	1630	BCS	VANFN		
	C59B	20	A8	1640	VIFN	JSR	SOREM	
	C59E	4C	5F	1650	JMP	VISSZ		
	C5A1	20	0D	1660	VANV	JSR	NEV	;MEGALLAPITJA A NEVET
	C5A4			1670				;OSAK AZ ELSO KET ELEMET
	C5A4	20	38	1680	JSR	HAROM		;3 URES HELY
	C5A7	4C	83	1690	JMP	VIV		
	C5AA	20	0D	1700	VANS	JSR	NEV	
	C5AD	A9	24	1710	LDA	##24		;#
	C5AF	20	D2	1720	JSR	KIIR		
	C5B2	20	38	1730	JSR	HAROM		
	C5B5	4C	87	1740	JMP	VIS		
	C5B8	20	0D	1750	VANE	JSR	NEV	
	C5BB	A9	25	1760	LDA	##25		;#
	C5B0	20	D2	1770	JSR	KIIR		
	C5C0	20	38	1780	JSR	HAROM		
	C5C3	4C	88	1790	JMP	VIE		
	C5C6	20	1C	1800	VANVT	JSR	NEVT	
	C5C9	20	43	1810	JSR	ZJEL		
	C5CC	4C	8F	1820	JMP	VIVT		
	C5CF	20	1C	1830	VANST	JSR	NEVT	
	C5D2	A9	24	1840	LDA	##24		;#
	C5D4	20	D2	1850	JSR	KIIR		
	C5D7	20	43	1860	JSR	ZJEL		
	C5DA	4C	93	1870	JMP	VIST		
	C5DD	20	1C	1880	VANET	JSR	NEVT	
	C5E0	A9	25	1890	LDA	##25		;#
	C5E2	20	D2	1900	JSR	KIIR		
	C5E5	20	43	1910	JSR	ZJEL		
	C5E8	4C	97	1915	JMP	VIET		
	C5EB	B1	FB	1917	VANFN	LDA	<#FB>,Y	
	C5ED	29	9F	1918	AND	##3F		;63
	C5EF	F0	03	1919	BEI	UGRIK		
	C5F1	20	A8	1920	JSR	SOREM		
	C5F4	20	A8	1921	UGRIK	JSR	SOREM	
	C5F7	A9	46	1922	LDA	##46		;#
	C5F9	20	D2	1923	JSR	KIIR		
	C5FC	A9	4E	1924	LDA	##4E		;#
	C5FE	20	D2	1925	JSR	KIIR		
	C601	20	1C	1926	JSR	NEVT		
	C604	20	4D	1927	JSR	RVSKI		
	C607	20	A9	1928	JSR	SOREM		
	C60A	4C	96	1929	JMP	VIFN		
	C60D	20	24	1930	NEV	JSR	NEV1	;A NEV ELSO BETUJE
	C610	20	D2	1940	JSR	KIIR		
	C613	8D	F9	1945	STA	JELZO		
	C616	20	29	1950	JSR	NEV2		;A MASODIK ELEM
	C619	4C	D2	1960	JMP	KIIR		
	C61C	A9	12	1970	NEVT	LDA	##12	;18=RVS ON
	C61E	20	D2	1980	JSR	KIIR		
	C621	4C	0D	2020	JMP	NEV		
	C624	9A		2030	NEV1	TXA		
	C625	18		2040	CLC			
	C626	69	41	2050	ADC	##41		;A
	C628	60		2060	RTS			
	C629	98		2070	NEV2	TXA		
	C62D	F0	0B	2080	BEG	URES		;NINCS MASODIK BETU
	C62E	18		2090	CLC			
	C62D	69	36	2100	ADC	##36		;54, HA 16Y AZ OSSZEG >=65, BET
	C62F	20	18	2110	JSR	BETUE		;BETUE
	C632	B0	03	2120	BCS	URES		;ILL. BETU
	C634	30		2130	SEC			
	C635	E9	07	2140	SBC	#7		;UGYANIS SZAM
	C637	60		2150	URES	RTS		
	C638	A9	20	2160	HAROM	LDA	##20	;SZOKOZ
	C63A	20	D2	2170	JSR	KIIR		
	C63D	20	D2	2180	KETTO	JSR	KIIR	
	C640	4C	D2	2190	JMP	KIIR		
	C643	A9	28	2200	ZJEL	LDA	##28	;#
	C645	20	D2	2210	JSR	KIIR		
	C648	A9	29	2220	LDA	##29		;#
	C64A	20	D2	2230	JSR	KIIR		
	C64D	A9	92	2240	RVSKI	LDA	##92	;RVSOFF
	C64F	20	D2	2250	JSR	KIIR		



```

0652 A9 20 2255 LDA #*20
0654 40 30 06 2260 JMP KETTO
0657 A5 06 2270 LAPOZ LDA #SOR
0659 09 15 2280 CMP #*15 ;HA NEM ERI EL, C=0
065B 90 0A 2290 BCC URES ;ELAGAZIK, HA A SOR>21
065D AD F9 03 2310 NLAPP LDA JELZO
0660 F0 05 2312 BEQ URES
0662 A5 08 2314 HLAPP LDA #LENY
0664 09 40 2320 CMP #*40 ;64; NINCS LENYOMOTT BILLENTYU
0666 F0 FA 2330 BEQ NLAPP
0668 A9 00 2333 LDA #0
066A 80 F9 03 2335 STA JELZO
066D A5 0B 2340 ALL LDA #LENY
066F 09 40 2350 CMP #*40
0671 D0 FA 2360 BNE ALL
0673 8A 2370 TXA
0674 48 2380 PHA
0675 98 2390 TYA
0676 48 2400 PHA
0677 20 44 E5 2410 JSR CLR
067A 20 E0 06 2415 JSR IRCIM
067D 68 2420 PLA
067E A8 2430 TRY
067F 68 2440 PLA
0680 AA 2450 TAX
0681 60 2460 RTS
0682 A9 00 2470 TORLO LDA #0 ;AZ
0684 80 FC 03 2480 STA BETU1 ;ATMENETI
0687 80 FD 03 2490 STA BETU2 ;TAR
068A 80 FE 03 2500 STA TIPUS ;TORLESE
068D 80 FF 03 2510 STA TOMB
0690 A9 00 2514 TOR LDA #0
0692 85 02 2517 STA *2
0694 60 2520 RTS
0695 A9 00 2530 CIMBE LDA #0 ;AZ
0697 A2 00 2540 LDX #*00 ;ADATTAR
0699 85 FB 2550 STA #*FB ;INDULO
069B 86 FC 2560 STX #*FC ;CIMEHEK
069D 60 2570 RTS ;BEALLITASA
069E A9 00 2600 KI LDA #0 ;KIKAPCSOLJA A SZOKOZ ATUGRST
06A0 85 83 2610 STA #*83
06A2 60 2620 RTS
06A3 A9 EF 2650 BE LDA #*EF ;VISSZAKAPCSOLJA A SZOKOZ ATUGR
RST
06A5 85 83 2660 STA #*83
06A7 60 2670 RTS
06A8 A9 0D 2700 SOREM LDA #*D ;RETURN
06AA 20 02 FF 2710 JSR KIIR
06AD 40 57 06 2720 JMP LAPOZ
06B0 A9 01 2800 IRCIM LDA #1
06B2 85 06 2810 STA #SOR
06B4 A9 0E 2820 LDA #*E
06B6 85 03 2830 STA #*032
06B8 20 6C E5 2840 JSR POZ
06BB A2 00 2850 LDX #0
06BD 8D 09 06 2860 IRAS LDA CIMIR,X
06C0 20 02 FF 2870 JSR KIIR
06C3 E8 2880 INX
06C4 E0 0E 2890 CPX #*E
06C6 D0 F5 2892 BNE IRAS
06C8 60 2900 RTS
06C9 56 2910 CIMIR .BY #56 ;W
06CA 41 2911 .BY #41 ;A
06CB 4C 2912 .BY #4C ;L
06CC 54 2913 .BY #54 ;T
06CD 4F 2914 .BY #4F ;D
06CE 5A 2915 .BY #5A ;Z
06CF 4F 2916 .BY #4F ;D
06D0 4C 2917 .BY #4C ;D
06D1 49 2918 .BY #49 ;I
06D2 53 2919 .BY #53 ;S
06D3 54 2920 .BY #54 ;T
06D4 41 2921 .BY #41 ;A
06D5 0D 2922 .BY #D ;RETURN
06D6 0D 2923 .BY #D ;RETURN
5000 .EN

```

0400 / 0607 / 0207
R FURRAS FILE: VRLT020 16.SRC
@ HIBA

ADAT	C483	ALL	C660	BE	C6A3	BEIR	C487	BETU	C4CC
BETU1	C3FC	BETU2	C3FD	BETUE	B113	CHG	007C	CHGET	0073
CHSOT	0079	CIM	C529	CIMBE	C695	CIMIR	C6C9	CLR	E544
E	C4AD	ERROR	AF08	FELO	C4F1	FELO2	C500	HAROM	C638
HIBA	C4C9	IOEZO	C472	IRAS	C680	IRCIM	C680	JELZO	C3F9
KETTO	C63D	KI	C69E	KIIR	FFD2	KOVR	C3FA	KOVF	C3FB
LAPOZ	C657	LEB	C4E7	LENY	000B	MBETU	C525	NDATA	C490
NEMFN	C4A6	NEV	C600	NEV1	C624	NEV2	C629	NEVT	C61C
NFN	C539	NIDEZ	C47F	NINCS	C578	NKESZ	C56F	NLAPP	C65D
NLAPP	C662	HONAT	C512	NREM	C46E	NSORV	C45D	NTB	C547
NULLA	C517	OS2	00D3	POZ	E56C	RYSKI	C64D	SOR	00D6
SOREM	C6A8	TIPUS	C3FE	TOMB	C3FF	TOR	C690	TOR1	C40D
TOR2	C408	TORLO	C682	TOMA	C430	TYPBE	C4DE	TYPUS	C406
UGRIK	C5F4	UJ	C44D	UJBET	C559	UJNAS	C55B	UJRA	C50B
UJSOR	C424	UJVAL	C450	URES	C637	USOR	C42A	VAL	C49A
VALT	C57D	VANE	C5B8	VANET	C500	VANFN	C5EB	VANS	C58A
VANST	C5CF	VANV	C5A1	VANVT	C5C6	VEGE	C557	VIE	C58B
VJET	C597	VIFN	C59B	VIS	C587	VISSZ	C55F	VIST	C593
VIV	C583	VIVT	C58F	VS	C53F	ZJEL	C643		

C-64 bővítések

SIMON'S BASIC C-64 cartridge

(A cikkünkhöz kapcsolódó hirdetés borítónk második oldalán, vásárlási utalvány a 34. oldalon.)

A részletes ismertetés előtt röviden arról, hogy ez a bővítés mire is jó.

A Commodore felhasználó, legyen az tanár, diák, mérnök, ha a C-64-en kívánja a saját programjait megírni, hamar rájön arra, hogy a gép által kínált lehetőségek az eredeti BASIC V.2-n csak nehezen valósítható meg. A grafika, zene, a sprite-ok vezérlése sokkal több és jobb utasításokat igényelne. Ezért, aki a számítógépén nem csak előre gyártott programokat kíván futtatni, hanem komoly feladatokat szeretne megoldani, annak feltétlenül szüksége van erre a bővítésre. Tehát ezt a bővítést a haladó programozóknak ajánlom.

A SIMON'S cartridge ismertetése

A C-64 esetében a programozást segítő bővítések két módszert alkalmaznak az új utasítások beépítésére: vagy egy különleges jellel kezdődnek a bővítés parancsai, vagy a BASIC interpreterhez hasonlóan — annak rutinjaihoz kapcsolódóan — tokenizálást alkalmaznak, azaz minden utasításhoz egy kódot rendelnek. Az előbbire példa a SUPER GRAPHICS (szalagos változat), GRAPAK vagy a HELP+. Az utóbbi reprezentánsai a GRAPHICS BASIC, PRINTER BASIC, WINDOW BASIC, SPEECH BASIC, BASIC 4.0, 7.0 és maga a SIMON'S, az ismert bővítések közül.

A tokenizálás több előnyt rejt magában. Tömörebb, így kisebb a helyfoglalása a memóriában és a lemezen, gyorsabb a végrehajtása. Mivel ugrási tábla tartalmazza az utasítások rutinjainak címeit, flexibilis, így más programokkal, pl. compilerekkel hajlandó együttműködni.

A SIMON'S több, az alap BASIC-ben elhanyagolt területet lefed. Támogatja a grafikát, a hanggenerálást, a periféria-kezelést, a strukturált programozást és

programszerkesztést. Ezért a nagyon sok és hatékony szolgáltatásért csak 8K szabad RAM-területet kell feláldoznunk, mert a program mérete ugyan 16K, de ebből 8K-t a BASIC interpreter alatti területre (A000 ... BFFF) rejt el. Ezen kívül a képernyő (C000), a definiálható karakterek (E000) és a sprite minták (F000) sem a szabad RAM rovására helyezkednek el. A program a bővítővel autostartosan jelentkezik be. A képernyőn bekapcsolás után néhány másodperccel megjelenik:

```
***EXPANDED CBM V.2 BASIC***
30 719 BASIC BITES FREE
```

A hardver rajza az 1. ábrán látható. Ez különösebb ismertetést nem igényel, hiszen hasonló áramkör már terítékre került az eddigi C-64-es témáknál. Több figyelmet kellene szentelni a szoftvernek, hiszen a mindennapi használat feltétele, hogy alaposan megismerjük az új utasításokat.

Mivel ennek az ismeretterjesztő cikknek nem lehet feladata a program részletes ismertetése, mert az a terjedelme miatt lehetetlen lenne, ezért célszerű beszerezni a Plenge—Szceppanowsky szerzőktől a Data Becker—Novotrade kiadásában megjelent SIMON'S BASIC gyakorlatok című könyvet. Ez a 355 Ft-os árával nem olcsó, de megéri, mert nagyon jól, izgalmasan, sok mintapéldán keresztül mutatja be az új utasításokat. Jőmagam és kollégáim is ezt a könyvet lapozgatva kaptunk kedvet e téma elkészítéséhez, ismertetéséhez.

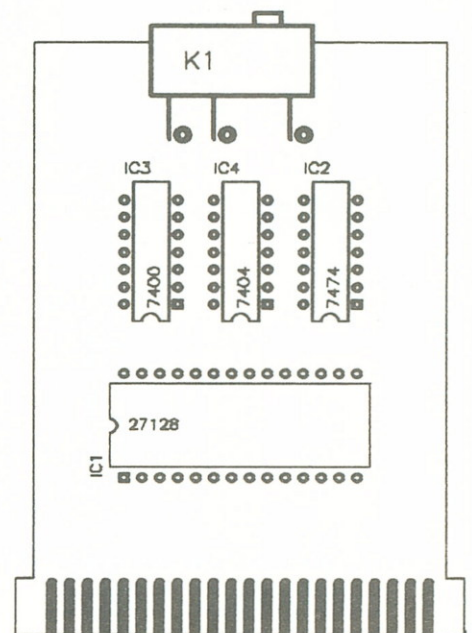
A szoftver mellett azonban ennyivel nem mehetünk el, ezért a teljesség igénye nélkül néhány sorban leírom, hogy valójában az új utasítások hogyan csoportosíthatók:

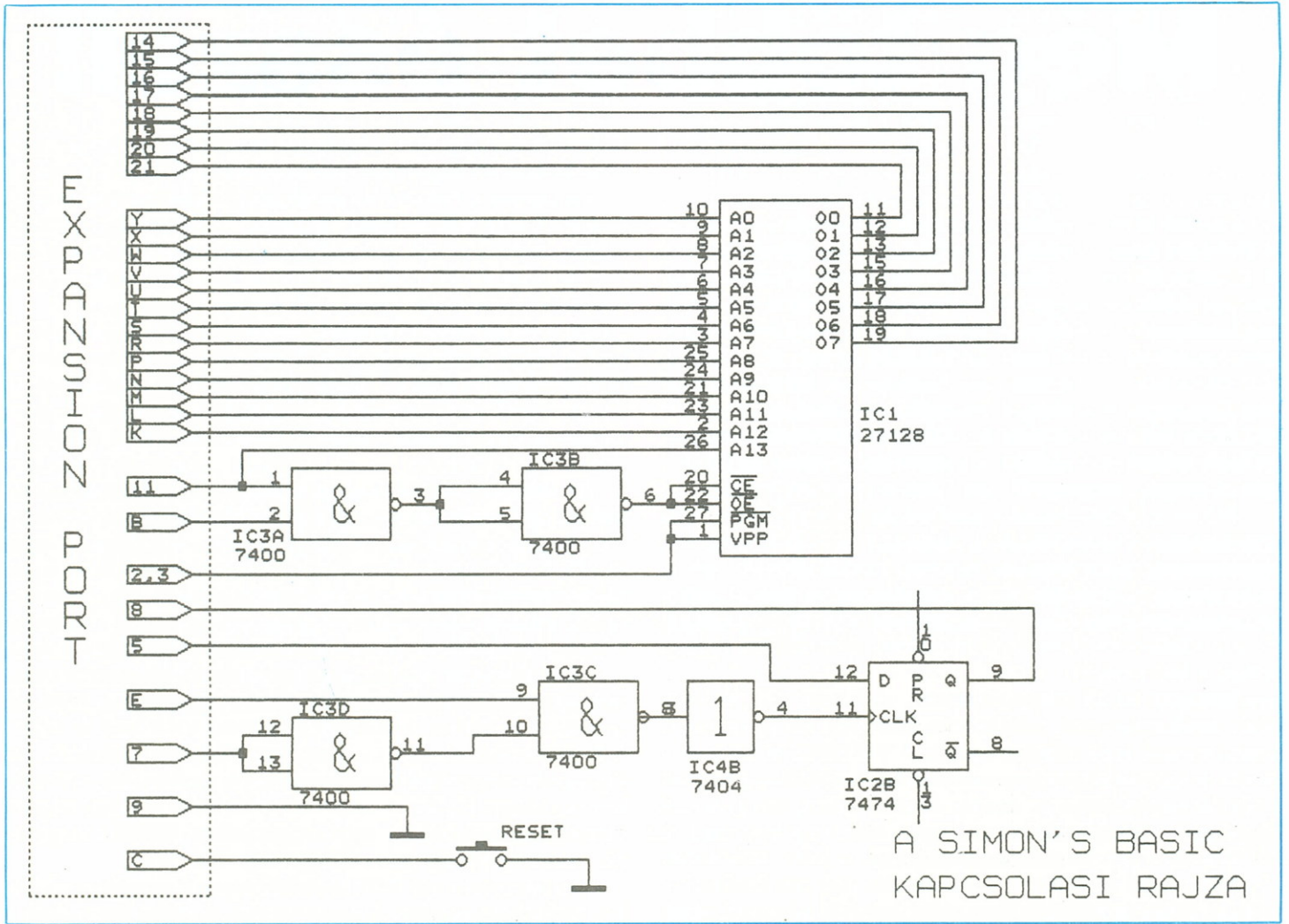
1. Grafika: két- és többszínű üzemmód, területfeltöltés, átszínezés, mozgatás, görgetés, invertálás, karakter-definiálás, kiírási koordináta, íráskép megadása, középzeigazítás stb.
2. Sprite-használat: két- és többszínű üzemmód, egyszerű definiálás, para-

méter-megadás, mozgatás, animáció, ütközésvizsgálat stb.

3. Periféria-kezelés: fényceruza, potenciométeres analóg botkormány, joystick kezelés, lemezes parancsok (ezen belül programvesztés nélküli tartalom jegyzés, olvasás) ellenőrzött, formázott beviteli utasítások stb.
4. Hanggenerátor: hangerő, hullámforma, burkológörbe, szólam, hangmagasság és -hossz könnyű programozhatósága, háttérzene üzemmód stb.
5. Strukturált programozás: kiszámított ugrás, REPET... UNTIL; LOOP... EXIT-IF ... END LOOP ciklusvezérlés, feltételvizsgálat ismétlése, alprogram-használat, lokális, globális változók szétválasztása stb.
6. Programszerkesztés, belövés: hibakezelés, átsorszámozás, auto-sorszámozás, funkcióbillentyűk használata, programok összefűzése, Data Pointer beállíthatóság, várakozás. Formázott, kiemelt SIMON'S utasítás és szabályozott sebességű listázás. Sztring keresés, nyomkövetés, változó DUMP, sorok titkosítása stb.
7. Új függvények: maradékérték, tört-rész, kizáró vagy egész osztás stb.

SIMON'S cartridge beültetési rajza





A cartridge szerelése, élesztése

A cartridge nyomtatott áramköre furat galvanizált. Három helyezőfurattal

csatlakozik a feliratozott dobozhoz, ezért erről egy helyezőcsapot le kell vágni. Az egységcsomag dobozánál ez megtörtént, csak ellenőrizni kell.

Foglalatot a cartridge doboz kis belmagassága miatt nem lehet használni, így

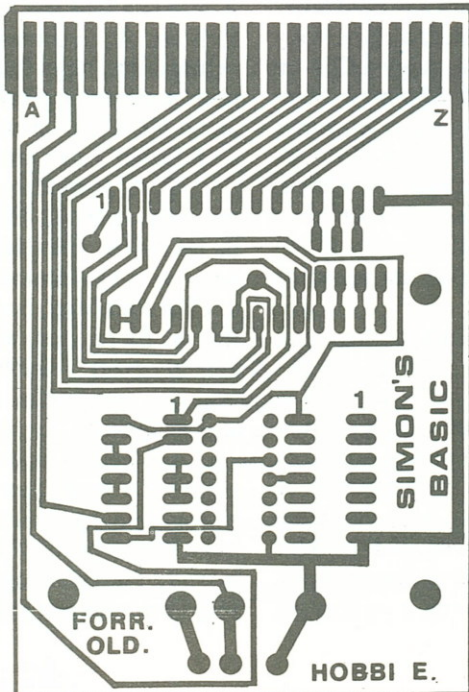
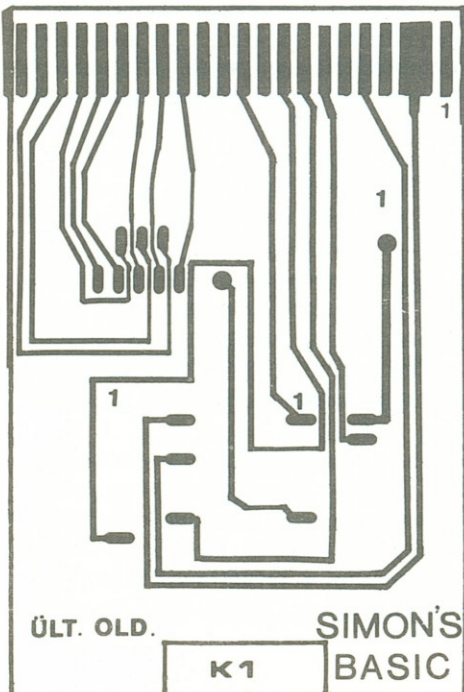
az IC, esetleges kiemelése az egy oldalon történő forrasztás miatt sokkal könnyebb.

A panelt erős fényvel átvilágítva vizsgáljuk meg, hogy nincs-e rajta zárlat vagy szakadás.

A kontúrok mentén reszeljük méretre a 3. ábra alakjának megfelelően. A csatlakozósor méretre igazításánál fokozottan ügyeljünk a pontosságra. Először az egyik oldalát reszeljük meg, majd illesztjük a C-64 csatlakozójához. Az eltérésnek megfelelően korrigáljuk a pozíciót és reszeljük be a csatlakozósor másik végét is.

Forasszuk be az IC-eket, majd a K₁-et. Ez utóbbit fektetve helyezük a panelre, és a lábait egy-egy ellenállásból levágott huzaldarab segítségével forrasztuk be. Ez megfelelő rögzítést is ad. K₁ helyére kétféle nyomógomb is használható, a magyar Pm 2-III és a szovjet MP-7 típus. A méretük nem azonos, ennek megfelelően a beforrasztáshoz több forrpont áll rendelkezésre.

K₁ helyét a műanyagházon sliccelni kell, a feliratozott oldalon. A bővítőt a feliratozott oldalával felfelé nézve csatlakoztassuk a C-64-hez, *kikapcsolt állapotban!*



Még többet ésszel!

VII. rész

Az első feladvány főleg azoknak lehetett könnyű, akik már az assembly programozást is kóstolgatták. Ez a művelet ugyanis ott gyakran felbukkan, és önálló utasítás is van hozzá, az EOR (itt-ott a XOR alakkal is találkozhatunk). A kódszó a „kizáró vagy” kifejezés rövidítése. A művelet szabályait $X \text{ eor } Y$ eredménye pontosan akkor 1, ha X és Y különbözik, egyébként 0. Más szóval akkor 1, ha vagy csak X , vagy csak Y értéke 1, a két lehetőség egymást kizárja. Szembeállítható ezzel az OR művelet, amely mindkét esemény együttes bekövetkezését is megengedi. Ha szükség van a különbség kiemelésére, akkor az OR műveletet emiatt „megengedő vagy”-nak is szokás nevezni.

Sajnos a BASIC-be ezt a gyakran használt műveletet nem vették fel önálló kódszóval. Az 1. példa az $A \% \text{ EOR } 52$ megvalósítását mutatja be két eltérő módszerrel. Az első változat a művelet meghatározásából következik, ez a klasszikus alak. A második, elegánsabb formula nemrég szerepelt a C-Újságban (92/5. 29. o.); tökéletesen beválik, de bevallom, nem tudnám bizonyítani a helyességét.

Tartozom az 52 megindoklásával. A feladat a 2., 4. és 5. bitek átbillentése volt, készítettem tehát egy olyan maszkot, amelyben csak ezek a bitek magasak. A $\%00110100$ pedig pontosan 52. Lehet próbálgatni, ezzel a maszkkal az EOR tényleg csak a kijelölt biteket váltja át, mindig ellenkezőjükre, és egymástól függetlenül. Egy apróság: a csupa egyes bittel (egy byte-on 255, két byte-on —1) való eorolás megegyezik a NOT művelettel.

Most, hogy az EOR-t is megismertük, már van értelme beszélni a WAIT utasításról. Nem sokan ismerik a használatának módját, még kevesebben merik alkalmazni, pedig néha bizony jól jönne. A 2. példán látható egy kis várakozó ciklus, utána ugyanennek a megvalósítása WAIT-tel. Az első változatban az olvashatóság kedvéért alkalmaztam az EOR operátort, a valóságban persze ezt az előbb mutatott egyik kifejezéssel kell helyettesíteni. Természetesen mindhárom változó helyett bármilyen numerikus kifejezés állhat. A „B” tag elhagyható, ekkor a gép nullával helyettesíti.

A sorozat 5. részének vége felé foglalkoztam egy nem teljesen tökéletes WAIT utasítással (3. pl.), amelynek az a célja, hogy egy billentyű leütésére várjon. Ehhez tudni kell, hogy a C64-en a 197-es rekeszben az éppen nyomva tartott billentyű (nem ASCII) kódja van, amely 0 és 63 közé eshet. Ha nincs lenyomva semmi, akkor 64 a tartalma. Ha 63-mal maszkolom ezt a byte-ot, akkor a 64 és a 0 — ami a Del kódja — egyaránt nullát ad vissza, vagyis a 3. példa gépi ciklusát a Del nem szakítja meg. Jobb megoldás, ha inkább a 6. bitet figyelem, amely 64 alatti számok esetén alacsony. Az elmondottak alapján, gyakorlásként javaslom a 4. példa elemzését.

Térjünk át a második feladványra (5. pl.)! Ha a megfogalmazásban ezt nem is emeltem ki, de lényegében a ciklus megszakítása a feladat. Akik kellő körültekintés nélkül ezt egyenértékűnek tekintik a ciklusból való kiugrással (6/Éa. pl.), azok a strukturálás elve ellen vétének. Korábban már elmondtam, hogy egy ciklusból csak egy ponton illik kilépni, és egy FOR esetében ez a pont a NEXT utasítással már el van foglalva. Úgy kell tehát kilépünk, hogy ez a NEXT-en keresztül történjen, akár a ciklus végértékének elérése előtt. Kicsit körülményes, de megoldható.

Hivatkoznék most a program olvashatóságára, a logikus vo-

nalvezetésre, egyebekre. Mindez persze most is érvényes, csak hogy komolyabb baj is származhat az ilyen húzásból, mint egy csúnya program. Két oka van ennek: az interpreter rugalmassága és a FOR ciklus eltérő volta. Izgalmas? Ha nem eléggé, akkor kérlek, próbáld ki a 7. példát a saját gépeden. Remélem, nálad is hibával áll le. Ha pontosan érted a hiba okát, akkor a jövő számban találkozunk.

A FOR abban különbözik az összes többi, BASIC-ben előírható ciklustól, hogy magának a ciklusnak a végrehajtása, szervezése gépi kódban történik. Úgy értem, hogy egy másfajta ciklus BASIC-szintű értékadásokból, elágazásból, ugrásból áll, azaz minden a szemünk előtt, általunk kiadott utasítások alakjában zajlik.

A szubrutin fogalmának ismertetésekor (92/9. szám) említettem, hogy a gépben van egy elkülönített társzakasz, a verem. Az interpreter itt tárolja a GOSUB utasítás visszatérési címét, de a FOR ciklusok szervezéséhez szükséges adatokat is, a ciklusváltozó nevét (címét), a végértéket és egyebeket. Ha a verembe egy A, majd egy B elemet teszünk, akkor az A-hoz csak úgy férhetek, ha a B-t előbb kivesszem. Erre rövidesen szükségünk lesz.

Az interpreter sokkal rugalmasabban kezeli a programot, mint egy hagyományos compiler. Például egy ciklus a FOR-nál kezdődik, és bárhol, bármelyik megfelelő NEXT-nél ér véget. A 8. példán látható iszonyatos program hibátlanul működik, pedig a ciklus vége előbb van, mint az eleje; ráadásul lóg a levegőben egy idegen NEXT is, amelyre sosem kerül sor. Egy szigorú compiler már ott leállna, ahol a „gazdátlan” NEXT I megjelenik, mondván, hogy nincs előtte FOR.

Az interpreter egy FOR hatására beteszi a verembe a ciklus-hoz szükséges adatokat, és megy tovább. A ciklus lényegében a NEXT-től működik. A NEXT végrehajtásakor a gép megkeresi a veremben az ugyanerre a ciklusváltozóra (!) kiadott FOR adatait. Ha kell, módosítja a ciklusváltozót és ismételi. Ha nem, akkor törli a ciklus adatait a veremből, és megy tovább. Veremből kivenni valamit csak a rajta levőkkel együtt lehet, igaz?

Megtehetjük, hogy egy FOR-hoz több NEXT-et is elhelyezünk, csak gondolni kell arra, hogy a ciklus után a program mindig ott folytatódik, amelyik NEXT-nél a ciklus éppen lejárt. A ciklus addig tart, amíg a végérték elérésekor a NEXT nem törli azt. Az interpreter nem köti meg a kezünket, de ezért nem veheti észre, hogy futás közben átlépte azt a pontot, ahol szerintünk a ciklus logikai vége van. Márpedig ha nem találkozott annyi NEXT-tel, amennyi ismétlést a FOR-ban előírtunk (mert a ciklusból kiugrottunk), akkor a ciklus nem szűnik meg.

Na jó, mondhatod erre, bent marad egy ciklus, de van elég hely, kit érdekel? Hát, egy FOR a kb. 240 byte-os veremből, ha jól tudom, elfoglal 24 byte-ot, és a verem sok egyéb célra is kell. De sokkal fontosabb, hogy az interpreter a FOR végrehajtása előtt ellenőrízi, hogy azzal a változóval van-e már ciklus megnyitva. És ha igen, akkor a régit törli, persze minden későbbi adattal együtt. Tehát ha az előző, lezáratlan ciklusban megnyitottunk egy másikat is, akkor ennek adatai megszünnének az újbóli megnyitáskor!

A 7. példán akkor most már megnézhetjük, hogyan okozza a galibát a lezáratlan ciklus. Most az nem számít, hogy ennek a programnak így nincs semmi értelme. Az 1. sorban levő ciklusból a feltétel NEN teljesülése esetén kilépünk. Ez még nem hiba, hiszen a ciklust befejezhetjük máshol is, ha úgy van értelme. Az interpreter szépen fejben tartja, hogy az I értéke annyi, és ezzel

még tart egy ciklus. Aztán nyitunk egy másikat, J-vel. Nincs akadálya, a régibe ágyazunk egy újat, hétköznapi ügy. Aztán hirtelen megnyitunk egy másik I ciklust. A gép észleli, hogy ilyen már van. Nyilván az interpreter írói csökkenteni akarták annak esélyét, hogy félbehagyott ciklusokkal megtölthessük a vermet, ami egyébként „OUT OF MEMORY” hibaüzenetet eredményezne. Ezért a gép a veremből előbb kidobja a J-ciklus adatait, aztán a befejezetlen I-ciklusét, majd beteszi az új ciklust. A NEXT I rendben megy. A NEXT J-nek nyoma sincs, hibaüzenet, leállítás, a programozóban döbbenet. Azaz most már remélhetőleg nem.

Hogyan előzzük meg az ilyen hibát? Azt nem mondhatom, hogy a ciklust soha ne hagyjuk félbe, mert arra bizony gyakran szükség van. Van trükkös és nagyon egyszerű megoldás is. Az egyszerű az, hogy a ciklusváltozókat szigorúan meghatározott rendben, csak egymásba ágyazva használjuk. Én megszoktam, hogy I,J,X,Y sorrendben nyitok meg ciklusokat, vagyis X-ciklust csak akkor használok, ha közben nyitva tartok már egy I- és J-ciklust. Érdeemes végiggondolni, így csak olyan ciklus adatai ürülhetnek ki a veremből, amelyekre már nincs szükség. (Persze néha megszegem ezt a szabályt, de akkor jól megnézem, hogyan.)

Való igaz, ez a megoldás nem mindig alkalmazható. Vagy szeretnénk más ciklusváltozót is használni; vagy éppen a szokásokat eltérő sorrendben, tömbök kezelésekor gyakori az ilyen igény. Nincs más, a ciklus elhagyásakor rá kell bírni a gépet, hogy a ciklust szüntesse meg, tekintse befejezettnek. Hát akkor

fejezzük be! Egy FOR-ciklus akkor jár le, ha a gép a NEXT végrehajtásakor úgy találja, hogy a változó átlépte a végértéket. A ciklusváltozó olyan változó, mint a többi, azaz felülírható. Kilépéshez tehát bele kell írni a végértéket (6/b. pl.). Csak figyelni kell arra, hogy utána a NEXT-re fusson a program, akár egy GOTO árán is.

Előfordul, hogy szeretnénk a ciklusváltozó értékéből értesülni arról, hogy a ciklust szabályosan fejeztük-e be. Ekkor a változó értéke a végérték és a lépésköz összege, az 5. példában 1001. A 6/b. példa sajnos ugyanezt eredményezi megszakításakor is. Kiegészíthető az előbbi húzás tehát azzal, hogy megszakításakor nagyobb értékkel lépünk ki (6/c. pl.). Vagy éppen beállítható egy általunk választott érték, itt éppen nulla, de a kilépés marad (6/d. pl.). Persze itt gondolni kell arra, hogy az újrainytással kárt okozhatunk, ahogy már elmondtam.

Ha valamiért egyik megoldás sem tetszik, még mindig marad dobásunk: törölnetni kell a ciklust, közvetlenül. A BASIC nem képes a verem kezelésére, de kihasználhatjuk a FOR-ral kapcsolatos manőverét erre a célra. A törléshez elég egy rövid üres ciklus végrehajtása a kívánt helyen (6/e. pl.)

Bőven van még megbeszélnivalónk ebben a témában, például a változó nélküli NEXT-ről, vagy a GOSUB és a FOR kölcsönhatásáról. Legközelebb folytatjuk. Addig is ajánlom nézegetésre a 9. példát, ami hibásnak látszik és 10.-et, ami talán jónak.

Hódi Gyula

<p>(1.)</p> <p>○ A%=(NOT(A%) AND 52) OR (A% OR NOT(52))</p> <p>○ A%=(A% OR 52)-(A% AND 52)</p>	<p>(3.)</p> <p>○ WAIT 197,63</p>	
<p>(2.)</p> <p>○ 1 IF ((PEEK(C) EOR B) AND A)=0 THEN 1</p> <p>○ 1 WAIT C,A,B</p>	<p>(4.)</p> <p>○ WAIT 197,64,64</p>	
<p>(5.)</p> <p>○ 1 FOR I=1 TO 1000</p> <p>○ 2 GET V\$: IF V\$>" " THEN ...</p> <p>○ 3 NEXT I</p> <p>○ 4 ::</p>	<p>(6.)</p> <p>○ A) 2 ... GOTO 4</p> <p>○ B) 2 ... I=1000</p> <p>○ C) 2 ... I=1001</p> <p>○ D) 2 ... FOR I=-1 TO -1</p> <p>○ E) 4 FOR I=1 TO 1: NEXT I</p>	
<p>(7.)</p> <p>○ 1 FOR I=1 TO 100: IF 10*RND(1)<3 THEN NEXT I: END</p> <p>○ 2 FOR J=3 TO 5: FOR I=1 TO 8: PRINT A(I,J);</p> <p>○ 3 NEXT I: PRINT : NEXT J</p>		
<p>(8.)</p> <p>○ 1 GOTO 3</p> <p>○ 2 NEXT I: GOTO 5</p> <p>○ 3 FOR I=1 TO 3: GOTO 2</p> <p>○ 4 NEXT J</p> <p>○ 5 END</p>	<p>(9.)</p> <p>○ 1 FOR I=1 TO 3</p> <p>○ 2 GOSUB 8</p> <p>○ 3 NEXT I: END</p> <p>○ 8 FOR I=1 TO 2</p> <p>○ 9 NEXT I: RETURN</p>	<p>(10.)</p> <p>○ 1 FOR I=1 TO 5</p> <p>○ 2 GOSUB 8</p> <p>○ 3 END</p> <p>○ 8 NEXT I</p> <p>○ 9 RETURN</p>

Képernyővarázslat — a TED

Harmadik rész: A megszakítások programozása

Vágjunk hát bele a TED programozásába! Azért a megszakításokkal kezdem, mert a későbbiekben igen sok hasznukat vehetjük, és előbb-utóbb minden programozó szembetalálkozik egy olyan feladattal, ami kényelmesen csak ezek használatával oldható meg.

Mi is az a „megszakítás” (interrupt)? Nos, ez a processzor által nyújtott lehetőség arra, hogy programunk folyamatos (majdnem folyamatos) futása közben (!) más apróbb-nagyobb rutinok is lefussanak anélkül, hogy ezekkel törődnünk kellene. Hogy érthetőbbé tegyem a dolgot, gondoljunk egy kicsit a PLUS/4 bekapcsolás utáni állapotára! A képernyőn villog a kurzor, és a gép várja parancsainkat. Írjunk be egy nagybonyolultságú BASIC programot, például:

```
10 PRINT "EN VAGYOK A TELAPO":GOTO10
Indítsuk most el! A kép rövidesen tele lesz elmés szövegeinkkel. Most nyomjuk le a SHIFT és a COMMODORE gombokat egyszerre! Az történik, amit vártunk: a gép kapcsolgatni kezd a nagybetű/grafikus és kisbetű/nagybetű módok között, és ezt folytatja a BASIC program folyamatos futása közben (a kiírás a COMMODORE gomb lenyomása miatt lassul le).
```

Gondolkozzunk csak egy kicsit! Mi is történt most? Futott egy program (ez lehetett volna gépi kódú is), és eközben a gép más is csinált anélkül, hogy erre utasítást adtunk volna (vagyis figyelte a billentyűzetet). Hogy történt ez? Nos, a billentyűzetet a megszakítórutin figyelte. A megszakítás ugyanis nem más, mint a futó program időnkénti megállítása (felfüggesztése), amelyet egy kiegészítő rutin (a megszakítórutin) futtatása, majd a „félbehagyott” program folytatása követ. Mi is történik ilyenkor?

Mint azt már az első részben említettem, a TED adja a processzornak az IRQ jelet. Megszakítás akkor kérelmeződik, ha ez a jel logikai nulla szintbe megy (ez csak az SEI utasítással letiltható megszakításokra érvényes), vagy a processzor BRK (BRaK) utasításra futott (természetesen ezt a processzor maga veszi észre, ilyenkor a TED IRQ vezetéke nem vált ál-

lapotot; a BRK által kiváltott megszakítás SEI-vel nem tiltható le). Ha az IRQ jel vételekor az "I" (interrupt) bit értéke nulla, a processzor befejezi az éppen végrehajtás alatt álló műveletet, majd automatikusan elmenti a verembe az állapotregiszter, valamint a programszámláló értékét, majd elugrik az \$FFFE-FFFE vektor által mutatott címre. Itt a megszakítórutin található, ezt végrehajtja, majd az annak végén található RTI utasítás hatására visszaolvassa a programszámlálót és az állapotregisztert a veremből, majd folytatja a program végrehajtását.

Nézzünk bele a megszakítórutinba! Az \$FFFE-FFFF vektor a \$FCB3 címre mutat. Itt a regisztereket menti el a gép a verembe (hogy visszatéréskor zavartalanul folytathassa a programot, melynek futását félbeszakította), belapozza a rendszer-ROM-ot, majd elugrik a tulajdonképpeni megszakítórutinra (hiszen ez eddig csak az előkészítés volt), amely a \$CE00 címen kezdődik. Itt kiolvassa az állapotregiszter "B" (Break) bitjét, amely azt jelzi, hogy a megszakítást BRK utasítás váltotta-e ki. Ha a bit magas, a gép a \$0316-0317 vektor által mutatott címre ugrik, azaz alapesetben jellegzetes üzenettel MONITORba lép. Alacsony "B" bit esetén a \$0314-0315 vektor adja meg a folytatás címét. Itt az úgynevezett „rasztermegszakítás” ág kezdődik (erről mindjárt szöveg). Természetesen a vektorok átírásával kiegészíthetjük vagy teljesen kicserélhetjük az egyes rutinokat. A raszter-ág végrehajtása után a gép \$FCBE címre ugrik. Itt belapozza azt a ROM-ot, amelyik aktuális volt a megszakítás vételekor, visszaolvassa a regisztereket, majd az RTI utasítás hatására visszatér a vezérlés a főprogramra.

A raszter-ág tovább tagolható: az egyik ág másodpercenként 100-szor fut le (ez a fő raszter-ág, de kevésbé fontos funkciókat lát el, nagyrészt az osztott grafikus képet vezérli), a másik csak 50-szer, mely a gép működéséhez alapvető feladatokat lát el (kezeli a billentyűzetet, az RS-232 interface-t stb.). A magyarázat az, hogy egy képernyő-megjelenítő ciklus

alatt (mely másodpercenként 50-szer hajtódik végre) a főág kétszer hajtódik végre: először ott, ahol az osztott képernyős üzemmódban a grafikus/karakteres képernyő határa van (161. képernyősor), másodsor pedig a képernyő/alsó keret határán (204. sor), míg a mellékág csak a 204. sorban. Mikor küld a TED megszakításkérést a processzor felé? A TED három regisztere foglalkozik ezzel: az \$FF09 — ez érzékeli, ha valamelyik forrás megszakítást kér; az \$FF0A, melyben külön engedélyezhetjük vagy letilthatjuk ezeket; valamint az \$FF0B, melyben a rasztermegszakítás sorát adhatjuk meg (valójában az \$FF0A 0. bitje is ide tartozik). Az egyes megszakításforrások, és a hozzájuk tartozó bitek:

\$FF09 — megszakítás engedélyezése

(a bitek „1”-ben aktívak)

- b.7 — a processzor felé menő IRQ vezeték negáltja
- b.6 — harmadik időzítő
- b.5 — nem használt
- b.4 — második időzítő
- b.3 — első időzítő
- b.2 — fényceruza (Light-pen)
- b.1 — raszter
- b.0 — nem használt

\$FF0A — megszakítás érzékelése

- b.7 — nem használt
- b.6 — harmadik időzítő
- b.5 — nem használt
- b.4 — második időzítő
- b.3 — első időzítő
- b.2 — fényceruza
- b.1 — raszter
- b.0 — rasztermegszakítás sora 8. bit.

\$FF0B — b.7 — b.0 a rasztermegszakítás sorának alsó 8. bitje

Az időzítők használata egyszerű: egyesével számolnak lefelé 884 kHz frekvenciával, és megszakítást kérnek, ha értékük eléri a nullát. Ezután a második és harmadik időzítő \$FFFF-ról (65535-ről), míg az első az utoljára beléírt értékről kezdi újból a számlálást. Címeik:

Első időzítő: \$FF00-FF01

Második időzítő: \$FF02-FF03

Harmadik időzítő: \$FF04-FF05

Használatukkor először az alsó bájtot írjuk be, erre az időzítő megáll, és csak a felső bájt beírásának hatására indul újra. Ezen kívül csak TED-stop idején állnak, de ilyenkor a képgenerálás is szünetel. TED-stop az \$FF07-es vezérlőregiszter 5. bitjének 1-be állításával váltható ki (erről a regiszterről a grafikánál fogok részletesebben beszámolni). Az időzítők írása előtt érdemes letiltani a megszakításokat az SEI utasítással.

A fényceruza használata még egyszerűbb: nem használható. A TED-nek



```

○
○
○ mexak; 1.....page # 1
○
○ Line#   Addr Code           Source
○
○ 00001  0000           ;mexak.1.
○ 00002  0000           * = $2000.
○ 00003  2000          intl   = $0314
○ 00004  2000          inth   = intl+1
○ 00005  2000          kep    = $0c00           ;kepernyo teteje
○ 00006  2000          int    = $ce0e           ;megszakito-rutin
○ 00007  2000          ;
○ 00008  2000          ;
○ 00009  2000          ;
○ 00010  2000          ;
○ 00011  2000 78          sei     ;mexak: ki
○ 00012  2001 a2 0d          ldx #<mexak
○ 00013  2003 a0 20          ldy #>mexak
○ 00014  2005 8e 14 03          stx intl
○ 00015  2008 8c 15 03          sty inth
○ 00016  200b 58          cli     ;megszak. be
○ 00017  200c 60          rts     ;vissza
○ 00018  200d          ;
○ 00019  200d          ;
○ 00020  200d          ;
○ 00021  200d a2 03          mexak  ldx #3           ;4 kar.
○ 00022  200f bd 1b 20          cikl   lda kofa,x           ;felmasoljuk
○ 00023  2012 3d 00 0c          sta kep,x           ;a kepre
○ 00024  2015 ca          dex
○ 00025  2016 10 f7          bpl cikl           ;nincs vege
○ 00026  2018 4c 0e ce          jmp int           ;ugras a gepi rutinra
○           0b 0f 06 01
○ 00027  201b          kofa   .byt $0b,$0f,$06,$01 ;"kofa"
○
○ end of assembly, error count = 00000
○
○ cikl   200f   int   ce0e   inth   0315   intl   0314
○ kep    0c00   kofa  201b   mexak  200d
○
○
○

```

nincs fényceruza bemenete, az tehát nem is válthat ki megszakítást. Valószínűleg később akarták kifejleszteni ezt az eszközt (a géphez néhányan megpróbálták fényceruzát illeszteni, de ehhez nem ezt a megszakítást használják).

Rasztermegszakítás: akkor váltódik ki, ha a megjelenítés alatt álló sor száma (a rasztorsor számláló értéke) megegyezik az \$FF0A-FF0B párosba beállított értékekkel. A számtartomány:0—311-ig, ennél nagyobb beírt érték nem vált ki megszakítást.

Az \$FF0A adott bitjének 1-be állításával az adott forrás engedélyezhető, nullázásával letiltható. Alapállapotban csak a rasztermegszakítás aktív, mivel a gép rendszerprogramja csak ezt használja. Ez azonban csak az \$FF09 7. bitjére és a processzor felé menő IRQ vezeték állapotára

van hatással, ugyanis az \$FF09 adott bitje mindig jelzi az adott eszköz megszakításkérélmét, még ha azt az \$FF0A-ban le is tiltottuk. Így szokás figyelni programból az időzítőket, hiszen azok alacsony beírt értéknél olyan gyakran váltódnának ki megszakítást, hogy az a gépet kezelhetlenné tenné.

Ennyi elméleti bevezetés után elkezdhetünk programozni. A cikk elején említettem, hogy a vektorok segítségével beavatkozhatunk a megszakítórutinba. Próbálkozzunk meg először a legegyszerűbb megoldással! Írjuk át a \$0314-0315 vektort! Vigyük be az 1. listán látható programot, és futtassuk le!

A \$0314-0315 vektor alapállapotban a \$CE0E címre mutat. Ha ezt saját rutinunkra irányítjuk, akkor az másodpercenként 100-szor végrehajtódik (ezért

nem tudjuk a KOFA feliratot — mely egyébként a szerző beceneve — letörölni a képről). Rutinunk ezután elugrik a \$CE0E címre, hogy az eredeti megszakítórutint is végrehajtsuk. FIGYELEM!!! A vektort csak GÉPI KÓDBAN, SEI utasítás UTÁN írogassuk át, mert a „félíg beállított” vektor a gépet rossz címre irányíthatja (szinte bizonyos a gép „fagyhalála”), ugyanis ha olyankor jön megszakítás, mikor a vektornak még csak az egyik bájtyát írtuk át, a vektor rossz címre mutat, tehát a gép hibásan ugrik. Ugyanezt produkálja a rutin végén álló hibás cím (próbáljunk csak ki egy véletlenszámot a \$8000-\$FFFF intervallumban!)

Módosítsuk a rutint! Használjuk a \$0312-0313 vektort! Ehhez nem elég a \$0314-et \$0312-re, a \$0315-öt \$0313-ra cserélni, módosítani kell az ugrási címet



is a rutin végén (\$CE42-re). Futtassuk a módosított programot! A hatás nem módosul. Mi hát a különbség? Az, hogy mint már említettem, a \$0314-es ág egy képmegjelenítés alatt kétszer hajtódik végre, míg a \$0312-es ág csak egyszer. A felirat mindig kiíródik, itt ez nem okoz gondot. Írjuk most be a 2. lista programját! A keret és a háttérszín a képernyő egy részén felcserélődik (a 161. és 204. sorok között, ritkábban ezen kívül, attól függően, hogy hol tart a vektor átirásakor a képmegjelenítés). Oka: a második csere (hiszen egy ciklus alatt kétfő van!) a színeket természetesen visszacsereéli. Próbálkozzunk meg a rutin átalakításával úgy, hogy az a \$0312-0313 vektort használja!

Mi van akkor, ha nekünk nem felelnek meg ezek a fixen tartott helyek? Próbáljuk meg a programot kibővíteni azzal, hogy minden megszakításkor magunk állítjuk be a következő megszakítás helyét (pl. 10-zel növeljük az \$FF0B értékét)! Kelemetlen csalódás fog érni... a gép ugyanis

saját maga is beleír ebbe a regiszterbe, és ezzel visszaállítja az a saját \$A1 (161) vagy \$CC (204) értékére. Nyilvánvaló, hogy a megszakítás sorának áthelyezéséhez „meg kell szabadulnunk” a gép beavatkozásától. A következő lépés a 3/a listán található. Telecsíkozunk a képernyőt — természetesen szerint! első ránézésre furcsa (és felesleges) az \$FF09 cím olvasása, majd a kiolvasott érték azonnali visszairása, de nem szabad megfedkezünk arról, hogy ezen a címen nem egy egyszerű memóriabájt, hanem a TED egyik regisztere található. Ez a regiszter ugyanis íráskor törlődik, ilyenkor az 1-be váltott bitek nullába mennek, és ezzel a TED felkészül egy újabb megszakításkérelem fogadására, és egy esetleges újabb megszakítás kiváltására. Próbálkozzunk meg az írás elhagyásával, de erre a gép „fagyizással” reagál. Eddig ezt a törlést a \$CE0E rutin végezte el. A JSR \$DBII-gyel hívott rutin nem más, mint a billentyűzet figyelése, ezt használja az eredeti megszakítórutin \$CE42-es ága. A végén az ugrási cím vál-

tozott: a \$CE0E vagy \$CE42 \$CC3-ra. Az eredeti (gépi) rutin FCBE-re tér vissza, ahol visszalapozzák a hívó ROM-ot. Mivel a mi programunk nem lapozta ki a rendszer-ROM-ot, ez a rész a mi számunkra elhagyható.

Sajnálatos tény, hogy a csíkok a kéretben kissé remegnek. Ennek oka az, hogy a gép nem mindig akkor írja át a színeket, mint az előző alkalommal, ugyanis a megszakítás-jelzés vételekor éppen végrehajtott utasítás időigénye változhat. Ezt kiküszöbölhetjük, ha a megszakítást egy sorral előbb váltjuk ki, és itt a vízszintes videópozíció-számláló (\$FF1E) figyelésével várjuk meg a sor végét, és akkor váltunk keretszínt. Ezt a 3/b. lista mutatja be.

Vegyük most azt a sajnálatos esetet, amikor le kell mondanunk a ROM által nyújtott kényelemről, és mindent magunknak kell csinálnunk. Ilyenkor a \$0314-0315 vektort nem használhatjuk, hiszen oda csak a ROM-ban lévő megszakítórutin ugrik el. Marad az \$FFFE-FFF

```

○   mexak: 2.....page # 1
○
○   Line#   Addr Code           Source
○
○   00001   0000           ;mexak: 2.
○   00002   0000           * = $2000
○   00003   2000           int1    = $0314
○   00004   2000           inth    = int1+1
○   00005   2000           int     = $ce0e           ;megszakito-rutin
○   00006   2000           keret   = $ff19           ;keretszin
○   00007   2000           hatter  = $ff15           ;hatterszin
○   00008   2000           ;
○   00009   2000           ;
○   00010   2000           ;
○   00011   2000 78           sei     ;mexak ki
○   00012   2001 a2 0d           ldx #<mexak:
○   00013   2003 a0 20           ldy #>mexak
○   00014   2005 8e 14 03           stx int1
○   00015   2008 8c 15 03           sty inth
○   00016   200b 58           cli     ;megszak. be
○   00017   200c 60           rts    ;vissza
○   00018   200d           ;
○   00019   200d           ;
○   00020   200d           ;
○   00021   200d ae 19 ff           mexak  ldx keret           ;keret-hatter
○   00022   2010 ac 15 ff           ldy hatter           ;csere
○   00023   2013 8e 15 ff           stx hatter
○   00024   2016 8c 19 ff           sty keret
○   00025   2019 4c 0e ce           jmp int           ;megszakito-rutinhoz
○
○   end of assembly, error count = 00000
○
○   hatter  ff15           int     ce0e           inth    0315           int1    0314
○   keret   ff19           mexak   200d

```




vektor. Nézzük meg, mit csinál a gép, ha a ROM-ból veszi az információt, és másoljuk le nyugodtan! Szépen PHA-val a verembe kerül az akku, majd rajta keresztül a másik két regiszter. Itt garázdálkodhatunk tovább, mint eddig, majd vissza kell olvasnunk a regisztereket, és RTI-vel visszatérnünk a főprogramhoz.

Bizonyára már mindenki gépében volt „demo” vagy egyéb hasonló program. Itt — jó esetben — gyönyörű szín-effektusokat láthatunk, hasonlókat, mint a mienk, de a színváltás minden sorban megtörténik. Ez nem érhető el azzal, ha

minden sorban új megszakítást generálunk, hiszen ez olyan sűrűséget jelent, amellyel a gép már nem képes megbirkózni (nincs idő minden sorban kiváltani a megszakítást, végrehajtani a rutint, majd visszatérni). Ilyenkor a módszer: az első kívánt sor előtt megszakítunk, itt időzítünk, váltjuk a színt, és ciklusban visszatérünk addig, amíg az utolsó kívánt sort el nem érjük. Itt nagyon pontos időzítés kell, ezt esetenként csak kísérletezéssel lehet megállapítani. Erre példát a 3/d. lista szolgáltat. Fontos, hogy ne rakjuk a kezdősort karakter-kezdősorra, itt ugyanis a

TED késlelteti a processzort (mint az első részben leírtam, ilyenkor hozza el a TED a képelőállításához szükséges információkat).

Most már majdnem mindent tudunk a megszakításokról. Próbálkozzunk meg olyan program írásával, amely letiltott megszakítás mellett is elvégzi a megszakítórutin összes teendőjét. Egy lehetséges megoldás a 4. listán látható, de bizonyos vagyok benne, hogy az Olvasó ettől különböző módon is meg tudja valósítani a kitűzött célt.

Találkozunk a következő számban!

```

○ mexak 3./a.....page # 1
○
○ Line# Addr Code Source
○
○ 00001 0000 ;mexak 3./a
○ 00002 0000 * = $2000
○ 00003 2000 int1 = $0314
○ 00004 2000 inth = int1+1
○ 00005 2000 irqend = $fcc3 ;megszakitas vege
○ 00006 2000 keret = $ff19 ;keretszin
○ 00007 2000 irq = $ff09 ;hatterszin
○ 00008 2000 rasz = $ff0b ;mexak sora
○ 00009 2000 bill = $db11 ;billentyuzet
○ 00010 2000 ;
○ 00011 2000 78 sei ;mexak ki
○ 00012 2001 a2 12 ldx #<mexak ;vektor
○ 00013 2003 a0 20 ldy #>mexak ;beallitas
○ 00014 2005 8e 14 03 stx int1
○ 00015 2008 8c 15 03 sty inth
○ 00016 200b a9 04 lda #$04
○ 00017 200d 8d 0b ff sta rasz ;első mexak sora
○ 00018 2010 58 cli ;megszak. be
○ 00019 2011 60 rts ;vissza
○ 00020 2012 ;
○ 00021 2012 ad 09 ff mexak lda irq ;jelzo
○ 00022 2015 8d 09 ff sta irq ;torlese
○ 00023 2018 ad 19 ff lda keret ;szinvaltas
○ 00024 201b 49 36 eor #$36
○ 00025 201d 8d 19 ff sta keret
○ 00026 2020 18 clc
○ 00027 2021 ad 0b ff lda rasz ;kovetkezo
○ 00028 2024 69 10 adc #$10 ;mexakitas
○ 00029 2026 8d 0b ff sta rasz ;sora
○ 00030 2029 c9 f0 cmp #$f0 ;kep alja?
○ 00031 202b 90 03 bcc vege ;meg nem
○ 00032 202d 20 11 db jsr bill ;billentyuzet-kezeles
○ 00033 2030 4c c3 fc vege jmp irqend
○
○ end of assembly, error count = 00000
○
○ bill db11 inth 0315 int1 0314 irq ff09
○ irqend fcc3 keret ff19 mexak 2012 rasz ff0b
○ vege 2030

```



mexak 3./b.....page # 1

```

Line#  Addr Code          Source
00001  0000          ;mexak 3./b
00002  0000          * = $2000
00003  2000          .intl      = $0314
00004  2000          inth       = intl+1
00005  2000          irqend     = $fcc3          ;megszakitas vege
00006  2000          keret      = $ff19          ;keretszin
00007  2000          irq        = $ff09          ;hatterszin
00008  2000          rasz       = $ff0b          ;mexak sora
00009  2000          bill      = $db11          ;billentyuzet
00010  2000          pont      = $ff1e
00011  2000 78          sei        ;mexak ki
00012  2001 a2 12      ldx #<mexak ;vektor
00013  2003 a0 20      ldy #>mexak ;beallitas
00014  2005 8e 14 03  stx intl
00015  2008 8c 15 03  sty inth
00016  200b a9 03      lda #$03
00017  200d 8d 0b ff  sta rasz          ;első mexak sora
00018  2010 58          cli          ;megszak. be
00019  2011 60          rts          ;vissza
00020  2012          ;
00021  2012 ad 00 ff  mexak lda irq          ;jelzo
00022  2015 8d 00 ff  sta irq          ;torlese
00023  2018 a9 a0      lda #$a0
00024  201a cd 1e ff  hason cmp pont          ;varunk
00025  201d b0 fb      bcs hason
00026  201f ad 19 ff  lda keret          ;szinvaltas
00027  2022 49 36      eor #$36
00028  2024 8d 19 ff  sta keret
00029  2027 18          clc
00030  2028 ad 0b ff  lda rasz          ;kovetkezo
00031  202b 69 10      adc #$10          ;mexakitas
00032  202d 8d 0b ff  sta rasz          ;sora
00033  2030 c9 f0      cmp #$f0          ;kep alja?
00034  2032 90 03      bcc vege          ;meg nem
00035  2034 20 11 db  jsr bill          ;billentyuzet-kezeles
00036  2037 4c c3 fc  vege jmp irqend

end of assembly, error count = 00000

bill      db11      hason      201a      inth      0315      intl      0314
irq       ff09      irqend     fcc3      keret     ff19      mexak     2012
pont      ff1e      rasz       ff0b      vege     2037

```

mexak 3./c.....page # 1

```

Line#  Addr Code          Source
00001  0000          ;mexak 3./c
00002  0000          * = $2000
00003  2000          intl      = $fffe
00004  2000          inth       = intl+1
00005  2000          keret      = $ff19          ;keretszin
00006  2000          irq        = $ff09          ;hatterszin
00007  2000          rasz       = $ff0b          ;mexak sora

```



```

0
0
00008 2000      bill      = $db11          ;billentyuzet
00009 2000      rom        = $ff3e
00010 2000      ram        = $ff3f
00011 2000      pont      = $ffe          ;pontszamlalo
00012 2000      ;
00013 2000      ;
00014 2000      ;
00015 2000      ;
00016 2000 78          sei          ;mexak ki
00017 2001 8d 3f ff    sta ram          ;ram be
00018 2004 a2 17      ldx #<mexak      ;vektor
00019 2006 a0 20      ld y #>mexak     ;beallitas
00020 2008 8e fe ff    stx intl
00021 200b 8c ff ff    sty inth
00022 200e a9 03      lda #$03
00023 2010 8d 0b ff    sta rasz        ;első mexak sora
00024 2013 58          cli          ;megszak. be
00025 2014 4c 14 20    hopp jmp hopp      ;vegtelen ciklus
00026 2017      ;
00027 2017      ;
00028 2017      ;
00029 2017      ;
00030 2017      ;
00031 2017 40          mexak pha        ;akku mentes
00032 2018 8a          txa          ;x-regiszter
00033 2019 48          pha          ;mentes
00034 201a 08          ty a         ;y-regiszter
00035 201b 48          pha          ;mentes
00036 201c ad 09 ff    lda irq         ;jelzo
00037 201f 3d 09 ff    sta irq         ;torlese
00038 2022 a9 a0      lda #$a0
00039 2024 cd 1e ff    hason cmp pont     ;varunk
00040 2027 b0 fb      bcs hason
00041 2029 ad 19 ff    lda keret       ;szinvaltas
00042 202c 49 36      eor #$36
00043 202e 3d 10 ff    sta keret
00044 2031 18          clc
00045 2032 ad 0b ff    lda rasz        ;kovetkezo
00046 2035 69 10      adc #$10        ;mexakitas
00047 2037 8d 0b ff    sta rasz        ;sora
00048 203a c9 f0      cmp #$f0        ;kep alja?
00049 203c 90 00      bcc vege        ;meg nem
00050 203e 8d 3e ff    sta rom         ;rom be
00051 2041 20 11 db    jsr bill        ;billentyuzet-kezeles
00052 2044 8d 3f ff    sta ram         ;ram be
00053 2047 68          vege pla
00054 2048 a8          tay          ;y vissza
00055 2049 68          pla
00056 204a aa          tax          ;x vissza
00057 204b 68          pla          ;akku vissza
00058 204c 40          rti          ;mexak vege

end of assembly, error count = 00000

bill      db11      hason      2024      hopp      2014      inth      ffff
intl      fffe      irq        ff09      keret     ff19      mexak     2017
pont      ffe       ram        ff3f      rasz      ff0b      rom       ff3e
vege      2047

```





mexak 3./d.....page # 1

```

Line#  Addr Code      Source
00001  0000          ;mexak 3./d
00002  0000          * = $2000
00003  2000      intl      = $0314
00004  2000      inth      = intl+1
00005  2000      irqend    = $fcc3          ;megszakitas vege
00006  2000      keret     = $ff19          ;keretszin
00007  2000      irq       = $ff09          ;hatterszin
00008  2000      rasz     = $ff0b          ;mexak sora
00009  2000      bill     = $db11          ;billentyuzet
00010  2000      pont     = $ffe          ;
00011  2000      sor      = $ffd          ;
00012  2000          ;
00013  2000          ;
00014  2000          ;
00015  2000          ;
00016  2000 78          sei          ;mexak ki
00017  2001 a2 12      ldx #<mexak    ;vektor
00018  2003 a0 20      ldy #>mexak    ;beallitas
00019  2005 8e 14 03  stx intl
00020  2008 8c 15 03  sty inth
00021  200b a9 03      lda #3
00022  200d 8d 0b ff  sta rasz          ;első mexak sora
00023  2010 58          cli          ;megszak. be
00024  2011 60          rts          ;vissza
00025  2012          ;
00026  2012 ad 09 ff  mexak lda irq          ;irq
00027  2015 8d 09 ff  sta irq          ;torles
00028  2018 a0 00      ldy #0          ;mutato nullazasa
00029  201a ad 1d ff  lda sor          ;kovetkezo
00030  201d cd 1d ff  varsor cmp sor          ;sorra
00031  2020 f0 fb      beq varsor      ;varunk
00032  2022 b9 3d 20  olvas lda tabla,y    ;szin be
00033  2025 a2 96      ldx #96          ;var
00034  2027 ec 1e ff  hason cpx pont
00035  202a b0 fb      bcs hason
00036  202c 8d 10 ff  sta $ff10        ;szin ki
00037  202f 8d 15 ff  sta $ff15
00038  2032 c8          iny
00039  2033 c0 10      cpy #410
00040  2035 d0 eb      bne olvas          ;ciklushoz
00041  2037 20 11 db  jsr bill          ;billentyuzet
00042  203a 4c c3 fc  jmp irqend      ;vege
00043  203d          tabla .byt $01,$11,$21,$31
00044  2041          .byt $41,$51,$61,$71
00045  2045          .byt $81,$91,$a1,$b1
00046  2049          .byt $c1,$d1,$e1,$f1

end of assembly, error count = 00000

bill    db11    hason    2027    inth    0315    intl    0314
irq     ff09    irqend   fcc3    keret   ff19    mexak   2012
olvas   2022    pont     fffe    rasz    ff0b    sor     ffd
tabla   203d    varsor   201d

```

```

O      3. rész 4. lista
O
O      *=$2000
O      ;
O      ;
O      MEXAK=$FFFE          ;MEXAKITAS VEKTOR
O      IRQ  =$FF09          ;MEXAKITAS ERZEKELES
O      KERET=$FF19
O      ;
O      ;
O      ;
O      START  SEI          ;INTERRUPT KI
O      CSIK   INC KERET    ;KERET CSIKOZAS
O              BIT IRQ     ;TORTEENT MEXAKITAS?
O              BPL CSIK    ;NEM
O              LDA #>TOVABB ;PC MAGAS BYTE
O              PHA         ;VEREMBE
O              LDA #<TOVABB ;PC ALSO BYTE
O              PHA         ;VEREMBE
O              PHP         ;SR VEREMBE
O              PLA         ;VISSZAOLVASAS AKKUBA
O              AND #$EF    ;BREAK-BIT TORLESE
O              PHA         ;MODOSITOTT SR VEREMBE
O              JMP (MEXAK) ;VEKTOR ALTAL MUTATOTT CIMRE UGRUNK
O      ;
O      ;
O      ;ITT FOLYTATODIK A FUTAS A MEXAKRUTIN LEFUTASA UTAN
O      TOVABB JMP START
O      ;
O      .END
    
```

Superbooter C64

Ez a kis rutin nem egyszerűen egy autoboot program készítő a C64-eshez, egyúttal jelszóvédelemmel is ellátja a programot.

Saját programjainkhoz úgy használhatjuk, hogy begépeljük és a 10-es sorban A\$-ba tetszőleges jelszót írunk be az ismert szintaktikával: A\$ = "JELSZO". Ha beírtuk és elmentettük, futtassuk le majd töltsük be az autoboot-osítani kívánt

programot. Ezután nem kell mást tenni, mint ezt a programot elmenteni, lehetőleg új néven. Az így elmentett programot LOAD "PROGRAMNE",8,1-el betöltve a végrehajtás azonnal elindul.

Ha a program már fut, nyomjuk meg a RUN/STOP gombot, és a program megáll. A Superbooter ekkor az A\$-ban megadott jelszóra vár. Ha ezt beírjuk, a RUN/STOP gomb szabályosan fog működni.

```

O      0 REM SUPERBOOTER
O      10 A$="* IDE IRD BE A JELSZÓDÁT *":REM *** VALTOZTASD MEG A JELSZÓT ***
O      20 FOR I=828 TO 998:READ D:C=C+D:POKE I,D:NEXT I
O      30 IF C<>20705 THEN PRINT "HIBA A DATA-SOROKBAN...":END
O      40 IF LEN(A$)>25 THEN PRINT "JELSZO TUL HOSSZU !":END
O      50 IF A#<>" " THEN FOR I=1 TO LEN(A$):POKE 997+I,ASC(MID$(A$,I,1)):NEXT I:POKE 997+I,0
O      60 POKE 804,60:POKE 805,3:POKE 818,125:POKE 819,3
O      70 DATA 165,184,240,58,169,237,141,50,3,169,245,141,51,3,169,87,141,36
O      80 DATA 3,169,241,141,37,3,169,1,133,43,169,8,133,44,169,143,141,40,3
O      90 DATA 169,3,141,41,3,133,198,169,82,141,119,2,169,117,141,120,2,169
O     100 DATA 13,141,121,2,108,2,3,76,87,241,169,147,32,210,255,6,157,169,36
O     110 DATA 133,193,169,3,133,194,76,237,245,165,145,201,127,208,71,162,7
O     120 DATA 189,1,128,157,52,3,202,208,247,189,223,3,157,2,128,232,224,7
O     130 DATA 208,245,169,230,133,36,169,3,133,37,160,0,132,38,32,228,255,240
O     140 DATA 251,164,38,209,36,208,15,200,177,36,208,238,169,237,141,40,3
O     150 DATA 169,246,141,41,3,162,7,189,52,3,157,1,128,202,208,247,160,88,96
O     160 DATA 92,3,195,194,205,56,48,0
O
O      READY.
    
```

Lexikon

X. rész

SZENZOR

(Latinul sensus: érzékelés, felfogás.) Egy szenzor, amelyet érzékelőnek is neveznek, az a része egy mérő és regisztráló berendezésnek, amely közvetlenül érintkezésbe kerül a mérendő vagy a kijelző egységgel. A termoelemek hőmérsékletingadozásokat fognak föl. Azok az elektromos áramot jobban vezetnek magasabb (forróvezetés) vagy alacsonyabb (hidegvezetés) hőmérséklet mellett. Optoelektromos építőelemek a fényre reagálnak. Piezoelektromos építőelemeket a nyomás befolyásolja. A szenzor a fizikális egységeket azok mértékének arányában váltja át kiértékelhető (legtöbbször elektromos) jellé. (Például a fotocella a sorompóknál.)

A beszéd- és képfelismeréshez ilyen szenzorokon alapuló észlelőrendszereket fejlesztenek ki, amelyek fel tudják fogni és elemezni tudják a kimondott szavakat vagy a beolvasott képet.

ANALOG/DIGITÁLIS

Ez a műszaki nyelvben használatos ellentétes értelmű szópár az információfeldolgozás egyik alapfogalma. Jól szemlélteti a kétfajta eljárás közötti különbséget a kétféle hőmérő. Az analóg hőmérőknél a hőmérsékletet egy higanyoszlop magassága alapján állapítjuk meg, amelyet egy nullaponthoz viszonyítunk. A számjegyekkel dolgozó digitális hőmérő a hőmérsékletet egy bizonyos egységben (mondjuk tized fokokban) adja meg. Mialatt az analóg hőmérőknél elméletileg bármely tetszőleges köztes értéket lehet olvasni, a digitális hőmérőnél erre nincs mód. A digitális információ pontossága tehát a használt alapegység mértékétől

függ. A digitális szó eredete a latinra vezethető vissza: Digitus = ujj, azaz az ujjakon való számolásra utal.

ROBOT

(Csehül robota, azaz beszolgáltatás, munkarobot). Az 1920-ban először használt „robot” kifejezéssel önállóan mozgó automatákat illették, amelyek alakja többé-kevésbé az emberi formára emlékeztetett. Az automaták mozgását eleinte egy komplikált mechanikus óramű vezérelte, amelyek egyszerű emberi manuális tevékenységeket végeztek. Ma robotnak nevezünk mindazokat a programozható gépeket, amelyek a munkákat egy bizonyos fokig önállóan tudják elvégezni, illetve korlátozott mértékben tanulni is képesek. Ezeket a gépeket számítógépek irányítják és vezérlik. A robotok fő alkalmazási területei az iparban található (lásd hegesztőrobotok).

Mikroszámítógépek alkalmazásával, megfelelő technikával és programozással amatőr szinten is lehet kísérletezni.

VEZÉRLÉS

Egyszerűen mondva vezérlésnek nevezünk egy bizonyos állapotra való behatást. Például ahhoz, hogy este a szobában kellemes hőmérséklet legyen, reggel a (termosztát nélküli) fűtést egy tapasztalati értékre állítjuk be. Sajnos azonban elfelejtettük bezárni az ablakot. A meleg levegő egy része tehát elszökik, a hőmérséklet nem fog egy adott érték fölé emelkedni. A vezérlés elve tehát a vezérlendő egységre (itt a hőmérséklet) irányuló hatáson alapszik, egy állítólag (itt a fűtésszabályzó) segítségével. Az esetleges zavarjelek

(az ablak) a szabályzandó egységet úgy befolyásolhatják, hogy a kívánt hatást nem érzük el. Az itt említett „eset” nyitott hatásláncnak is nevezzük, ugyanis nincs visszacsatolás a szabályozott egységről a szabályzóra. A föllépő zavart az ember egyenlíti ki.

SZABÁLYOZÁS

Szabályozásról akkor beszélünk, ha egy adott egységet (pl. a hőmérsékletet) adott határértékeken belül kell tartani. Az egység állandóan kontroll alatt áll, és azt eltéréseknél visszavisszük az alapértékre (célérték). Ezt a zárt hatásláncot szabályozási körnek is nevezzük. A szobahőmérsékletet például termosztáttal szokás szabályozni. A szabályzóban egy mérőberendezés össze tudja hasonlítani a beállított értéket (alapjel, célérték) a valóságos értékkel (azaz a tényleges értékkel). Ha ezek nem egyeznek, akkor megváltozik egy szelep állása: több vagy kevesebb hőmennyiséget adunk le.

Szabályozókörre tehát ott van szükség, ahol a kívánt egységet nyitott hatásláncú vezérléssel nem lehet konstans értékben tartani, mivel zavaró hatás befolyásolja azt.

AD/DA ÁTALAKÍTÓ

Ezek az elektronikus építőelemek az analóg információt digitalizálják, a digitálisat analóggá tudják alakítani. A C64-nek is szüksége van egy AD-átalakítóra, hogy a potenciométerről érkező jeleket (paddle) a mikroporcesszor számára olvashatóvá tegye. A paddle esetében a jobb és a bal határérték között tetszőlegesen sok beállításra van mód, tehát ez egy analóg jelet ad le. Az AD-átalakító processzor felé vezető úton átalakítja ezt a jelet 0 és 255 közé eső számokká, s így az azt értelmezni és feldolgozni tudja. A nullától a maximális kitérést tehát a számítógép 256 lépésben tudja regisztrálni. A hangok digitalizálásánál a rezgést egyenlő „nagyságú” időintervallumokra osztjuk (például 1/10000 másodperces egységekre), s átváltjuk az időpillanatok amplitúdóját számértékekre.

Lexikon

SP 180 UC nyomtató + 1 doboz papír + tartozék festék-kazetta 14 000 Ft. Kiss György, 6000 Kecskemét, Bajza u. 13. IV/12.

576-ban megjelent és egyéb C-64-es programok lemezen, kazettán. A lehető legolcsóbb áron! Lemezes programok kazettán. Deutsch Szabolcs, 8749 Zalakaros, Fő u.19.

Akarsz szuper programot olcsón? Írj nekem! Vadi új

programok kazettán, lemezen. Kérj listát, bővebb információt. Deutsch Szabolcs, 8749 Zalakaros, Fő u.19.

C-64-re és PC-re 5000-es választékból 92-es programok eladók. A legjobb és legújabb! 3630 Putnok, Pf.: 25. K&C

Amiga 500-plusz + 1 MB RAM, 80 MB-os Winchester egyben vagy külön, valamint 1.3-as ROM-ok eladók. Tel.: 34/83-977.

ÚJDONSÁG! TOM SZÁMOLNI TANÍT - képes, rajzok számlálást tanító-gyakoroltató programcsomag TVC-re és C+4-re. Témakörök: számlálás, összehasonlítás, összeadás, kivonás, bontás 10-es, ill. 20-as számkörben. A 10 program ára mindössze 1200 Ft. Megrendelhető: SULI-SOFT 1327 Újpest 3, Pf.: 91.

C-64 konfiguráció eladó! Új alapgép + RCA kábel + lapkapcsolós joy-ok + szakiro-

dalom + havilapok (8900), magnó (1800), MPS 801 printer (8900), programkazetták (400), Action plusz cartridge (2500). Érdeklődni lehet a 176-2944-es telefonszámon, vagy a FRIEND TWO CREW levélcímén.

Megjelent C-64-re az ETNA SOFTWARE Shoot' em up játéka a GHASTLE. Megrendelhető utánvétellel (csak lemezen): 120 Ft + postaköltség. Badenszky Csaba, 2800 Tatabánya, Fő tér 9. 7/1.

KARÁCSONYI VÁSÁR — KARÁCSONYI VÁSÁR — KARÁCSONYI VÁSÁR

NOVOTRADE—2C Kft. ÁRLISTA

Hardverek

C-64 alapgép	13 990 Ft
VC 1541/II floppy drive	15 990 Ft
Datasette	2 990 Ft
Amiga 500 alapgép	39 990 Ft
Amiga 500 Plus alapgép	48 000 Ft
1351 Mouse (C-64-hez)	2 990 Ft
A 520 RF modulátor	3 900 Ft
Amiga tárbővítő	6 000 Ft
RocHard Hard disk illesztő	22 500 Ft
RocHard Hard disk illesztő + 40 Mbyte Winchester	50 000 Ft
RocGen Genlock RG300	10 750 Ft
Ethernet Card (A2000-hez)	35 000 Ft
MPS 1230 nyomtató	24 500 Ft
Philips 8833/II. (Amiga, C-64)	33 000 Ft

C64 játékok

	kazetta	lemez
Chamonix Challenge	499 Ft	549 Ft
Eddie Edwards Super Ski	499 Ft	549 Ft
Hostages	549 Ft	599 Ft
Impossible Mission II	581 Ft	668 Ft
Rolling Twins/I want more...	—	549 Ft
Operation Neptune	—	599 Ft
Prohibition	499 Ft	—
Sim City	—	599 Ft
Space Racer	—	599 Ft
Tin Tin on the Moon	549 Ft	599 Ft
Warlock's Quest	499 Ft	549 Ft
Xonox	399 Ft	499 Ft
Import járékkazetták	549 Ft	—

Hardverkiegészítők

2 RCA kábel	410 Ft
3 RCA kábel	575 Ft
Hálózati kábel	480 Ft
Adat kábel	360 Ft
Antenna kábel	340 Ft
Antenna váltókapcsoló	490 Ft
C64/128 Euro kábel	685 Ft
Amiga Euro kábel	1 250 Ft
Ékezetes Eprom SP180 VC	2 545 Ft
Ékezetes Eprom MPS 1230	3 500 Ft
Joy (Quick Shot II Plus)	850 Ft
Tápegység C64	3 000 Ft
Tápegység VC 1541/II	3 000 Ft
12" mono filter	700 Ft
12" color filter	750 Ft
14" mono filter	800 Ft
14" color filter	850 Ft
Disk Box 5'25" 3 db	99 Ft
Disk Box 307 5'25" 50 db	700 Ft
Disk Box 309 5'25" 100 db	800 Ft
Disk Box 310 5'25" 120 db	850 Ft
Disk Box 313 3'5" 10 db	150 Ft
Disk Box 318 3'5" 40 db	550 Ft
Disk Box 319 3'5" 80 db	750 Ft
Disk Box 320 3'5" 140 db	800 Ft
Mouse Pad 581	250 Ft
Mouse Pad + Holder	500 Ft
Mouse Set	1 000 Ft
Cleaner 562 5'25"	150 Ft
Cleaner 563 3'5"	150 Ft
Mouse Holder	250 Ft
Mouse takaró	250 Ft
Copy Holder 411	750 Ft
Copy Holder 412	1 250 Ft
Display szemüveg	750 Ft

Olcsón Commodore IC-k elektronikus alkatrészek beszerzése, C-64 és perifériáinak javítása. Tel.: 1-731-783 (üzenet).

C-64 konfiguráció eladó! Új alapgép (10 000), magnó (1500) + programkazetták, MPS 801 printer (10 000) + új festékszalagok, szakirodalom, Action plusz cartridge (3000), joystick (2 db 800), lemeztartó (500) stb. Érdeklődni lehet a 176-2944 telefonszámon, vagy a FRIEND TWO címén (1399 Budapest, Pf.: 701/55).

C-64-re új játék- és felhasználói programok eladók. Amiga, IBM PC játékprogramok is megrendelhetők. Kérjük, tüntesse fel a gépe típusát, és a hozzá tartozó perifériákat! Bélyeggel ellátott válaszborítékért listát küldünk. (Vízhatlan csomagolás, kedvezmények!) FRIEND TWO CREW, 1399 Budapest, Pf.: 701/55.

Figyelem! Majdnem komplett. Eladó: C-64/II + 1541/II + Philips monochrom monitor + 1541 datasette + 1541 Mouse + 2 joy + 19 kazettányi program (kb. 700) + 40 lemez szuperprogram + szakirodalom. Irányár: 42 000 Ft. Amíg a készlet tart! Joó Sándor, 1194 Budapest, Újvidék u. 28. Tel.: 1-285-722.

Keresek jó állapotban lévő IBM AT-t olcsón 386-os 40 MHz-ig. Kázmér Attila, 6000 Kecskemét, Gizella tér 2.

C-64 játékprogramok eladók kazettán! Lista: felbélyegzett borítékért: Németh Csaba, 1399 Budapest, Pf.:701/893.

C+4-hez! Új, japán 1551-es floppy eladó! 9800 Ft. Hosszú Kálmán, 3300 Eger, Szabó Sebestyén u. 9.

Olcsón keresek C=1551-es flopyt és nyomtatót (külön is)! Berényi László, 4461 Nyírtelek, Vasvári P. u. 112.

Megvételre vagy másolásra keresem a Citizen 120-d magyar nyelvű kezelési leírását. Kovács Gábor, 1046 Budapest, Sporttelep u. 3. II/6.

Eladó! C-64 + magnó + játékok 12 000 Ft-ért. Telefon: 1-755-240. Cím: Staub Ferenc, 1016 Budapest, Naphegy u. 22.

Eladó Amiga 500 tv modulátorral 35 000 Ft-ért. Oravecz Attila, Budapest XI., Irinyi J. u. 42. 1609-es szoba. Tel.: 1665-422/46

C+4, 1551 + 100 db lemez + tartó + 2 joy + könyvek 23 000 Ft. Balogh Sándor, 8200 Veszprém, Jutasi u. 67/a. II/2.

Eladó C+4-es, 1551 és 1541 drive, magnó, joy, játékok, könyvek, V 7.0 Basic-bővítő. 9-16-ig. Tel.: 112-9665

5000 + 4 programhoz cserepartnert keres. Kéri: Fanta (animáció) leírást. Marcsányi Z., 1143 Budapest, Ilka u. 56.

IBM számítógép, 2 váltós - AT és XT - billentyűzettel, XT floppyval eladó. Irányár: 35 000 Ft. Tel.: 158-4498. Monitor is megoldható!

Végre itt az iskola mindenki öröme, de ne felejtse el: a géped még mindig működik, és természetesen nálam továbbra is beszerezheted a legújabb 64 magnós programokat. Egy program csak 70 Ft. Minden 20. megrendelést ingyen teljesítem. Minden 5. program után 1 programot ingyen adok. Pl.: CREATURES 2, BUDOKAN, sokak kívánságára TURTLES 2, VIZ. A postaköltség téged terhel. Gégeny László, 4700 Mátészalka, Kézy László út 6. Tel.: (44) 12-811.

C-64-es programok cseréje, illetve eladása lemezen. Listát kérek és küldök. Cím: Csépanyi László, 3630 Putnok, Vásártéri út 8.

C+4-es kazettáimat eladom. Válaszborítékért listát küldök. Cím: Csépanyi László, 3630 Putnok, Vásártéri út 8.

Keresem a printfox 1987-es vagy fiatalabb változatát és az Ega V 3.2-es programot leírással. Földvári László, 5420 Túrkeve, Vásártér 22/c. II/5.

C-64 magnósok! Olcsó és vadonatúj játékprogramok, árendeményes egységcsomagok csak nálam! Ugyanitt utántöltős programcsere (cse-realap: 170 db).

Horváth Lajos, 8500 Pápa, Vajda Péter ltp. 45. III/10. Tel.: 89/24-239.

C-64 gépikódú programozás? Assembler Editor! Leírást kérj! Paczolay Dénes, 6400 Kiskunhalas, Kölcsey u.1.Eladó: C-64 13 + magnó + floppy + 60 lemez + tartó + 5 kazetta + 2 joystick + 3 cartridge + 3 könyv + mouse + pad. Ár: 40 000 Ft. Tel.: 120-2716.

Amiga tulajdonosok figyelem! Eladó 1 db infrás joystick. Mauer Péter, 1180 Budapest, Marosvásárhely u. 48.

Totókuleszkészítő programok C-64-re (8 Kbyte-osak). Válaszborítékért tájékoztató. Nagy Miklós, Újfehértó, Pf.: 41.

Amiga 500 (1 MB, V 1.3-2.0 joy, egér, ACTION III, lábjoy, pf1.; Bootkapcsoló, versionkapcsoló) eladó. Tel.: 1887-827.

C-128/64 gépre programokat adok-veszek-cserélek. 10 000 programom van. Cím: Járóka László, 1148 Budapest, Adria sétány 6. L/I.2.

C-64-es játék és felhasználói programok kazettán eladók. Kazettával együtt csak 380 Ft. 35-féle kazetta. Lemezes programok 140 Ft lemezzel együtt. 2500 program. Válaszborítékért listát küldök. Grásl János, 2800 Tatabánya, II., Réti út 32. IV/3.

Megjelent C-64-re az ETNA SOFTWARE Shoot'em up játéka a GHASTLE. Megrendelhető: csak lemezen, utánvétellel 120 Ft + postaköltség. Cím: Badenszky Csaba, 2800 Tatabánya, Fő tér 9. 7/1.

Fedezd fel a Ewing birodalmat, Dallas, C-64, lemez. Ár: 129 Ft, lemez: 58 Ft. Postaköltség az árban. Egerszegi B., Pécs, Kisszkókó.

C-64 diskre PROGRAMHEGYEK eladók! Régi és új játékok és felhasználói programok, szakleírások, GEOS-ok eredeti állapotban. Olcsó áron! Tájékoztatóhoz és listához szükséges 1 db lemez és 20 Ft-os bélyeg, vagy lemez nélkül 50 Ft. Holecz Ildikó, 1045 Budapest, Rózsa u. 21. VI/24.

C-64 Játék- és felhasználói programok lemezen. Adás-vétel—csere! Többezernyi programkinálat! Ysoft, 1701 Budapest, Pf.: 229.

C-64, 128 programok Évátó! Ár: 20 Ft/lemezoldal. Új No-Name lemezzel együtt 70 Ft/2 lemezoldal. Tájékoztatóhoz és listához szükséges 1 db lemez és 20 Ft-os okmánybélyeg, vagy lemez nélkül 50 Ft. Kopácsi Lajosné, 1031 Budapest, Vízimolnár u. 2. X/95.

Amiga programok Évátó! Ár: 25 Ft/lemez. Új No-Name lemezzel együtt 80 Ft/lemez. A tájékoztatóhoz és listához szükséges 1 db lemez és 20 Ft-os bélyeg, vagy lemez nélkül 70 Ft. Kopácsi Lajosné, 1031 Budapest, Vízimolnár u. 2. X/95.

Keresem kazettán, vagy cartridgen a Flimbo's Quest című programot. Cím: Párkányi Zsolt, Pécs, Hajdu Gy. u. 11.

Sürgősen eladó egyéves C128 + floppy + 2 joy + 60 lemez játékprogram + tartó + 11 könyv. Balázs Tamás, 2500 Esztergom, Kaán u. 5/a. Tel.: 33/12-498.

ACOMP Kft.

DECEMBERI

60 Ft-os

vásárlási utalványa

Beváltható készpénzes vásárlás esetén a 1141 Budapest, Álmos vezér útja 17. szám alatti üzletben.

Érvényes: 1992. december 31-ig.

Egy személy részére egyszeri vásárláshoz egy utalvány használható fel!

A NOVOTRADE SZERVÍZ Kft. az alább felsorolt szervízeiben mindenféle szervízszerelés munkadíjából 10% kedvezményt ad az egyesületi tagoknak.

1053 Budapest, Magyar u. 12—14	Telefon: 117-3551
1083 Budapest, Szigony u. 9.	Telefon: 134-3153
1191 Budapest, Gábor Á. sétány 3.	Telefon: 127-4763
3525 Miskolc, Fazekas u. 1—3.	Telefon: 46-17-011
4034 Debrecen, Holló L. u. 14.	Telefon: 52-32-863
5600 Békéscsaba, Bartók B. u. 37.	Telefon: 66-27-195
6724 Szeged, Csongrádi sugárút 76.	Telefon: 62-13-377
7624 Pécs, Jurisics M. u. 17.	Telefon: 72-11-812
8000 Székesfehérvár, Széchenyi u. 15/a.	Telefon: 22-12-711
9700 Szombathely, Szalonok u. 31.	Telefon: 94-13-419
Felvevőhelyek:	
9024 Győr, Babits M. 75.	
6000 Kecskemét, Széchenyi tér 1—3.	Telefon: 76-23-720

Igazolás: a javítandó berendezés leadásakor egyesületi igazolvánnyal. A kedvezmény többször is igénybe vehető.

NOVOTRADE
 SZERVÍZ Kft.

MAKROVILÁG utazási iroda

Beváltható utazás megrendelése esetén

az Üllői úti főirodában az alábbiak szerint:
 5 000 Ft-ig — 200 Ft kedvezmény
 10 000 Ft-ig — 400 Ft kedvezmény
 20 000 Ft-ig — 500 Ft kedvezmény
 20 000 Ft felett — 1000 Ft kedvezmény
 Csoportok jelentkezése esetén további kedvezményekről az irodában lehet tárgyalni

Az Országos Commodore Egyesület szolgáltatásai

Egyesületi tagoknak 20% kedvezmény:

VC—20 memóriabővítés 3—27 kByte-os:	kiépitéstől függő
C—16, C—116 memóriájának bővítése 64 kByte-ra:	3500 Ft
C—16 belső 16 kByte-os EPROM bővítés:	1450 Ft
C—16 belső 32 kByte-os EPROM bővítés:	2900 Ft
C—16 belső 8 kByte-os SOFT—ROM bővítés:	2800 Ft
C—16 belső 32 kByte-os SOFT—ROM bővítés:	4000 Ft
C—16 8 kByte-ról 32 kByte-ra átalakítás:	2000 Ft
C—16 és 1541 kompatibilis lemezegység párhuzamosítása:	3200 Ft
SOFTROM modul 32K, kikapcsoláskor sem felejt C-16, C-116, +4	5000 Ft
FÉK C—16, C—116, +4 potméteres sebességváltoztatás	
0%-tól 100%-ig fokozatmentesen	2000 Ft
TTL IC-teszter (Cartridge+lemezen a program)	4300 Ft
+4, C—16, C—116 UNI—ROM modul különféle kiépitésekben:	
— 8 kByte SOFT—ROM	3400 Ft
— 16 kByte SOFT—ROM	4000 Ft
— 8 kByte SOFT—ROM 16 kByte EPROM	4400 Ft
— 16 kByte SOFT—ROM 16 kByte EPROM	5000 Ft
— 16 kByte EPROM	2200 Ft

Egyesület tagoknak 30% kedvezmény:

Speeddos (átkapcsolható) operációs rendszer beépítése (C64 átalakítás, lemezegység átalakítás + párhuzamos kábel)	5000 Ft
1541 kompatibilis lemezegységbe elektronikus lemezlyukasztó beépítése	900 Ft
PAGEFOX magyar ékezetes kiadvány-szerkesztő cartridge (a teljes A/4-es oldal kinyomtatásához 640 pont/soros nyomtató szükséges minimum, pl. Citizen 120D)	5500 Ft
FASTLOAD cartridge (lemezes gyorsító, másoló, monitor)	1500 Ft
TTL IC-teszter cartridge + program	4300 Ft
288/256 Kbyte-os eprombank (vezérlő eprommal)	5000 Ft
Epromégető (2716-tól 27512-ig)	5000 Ft
C64-hez tároló oszcilloszkóp	8000 Ft
C64-bővítő-port elosztó (egyszerre 4 db cartridge lehet a gépben, melyeket gombnyomásra lehet kapcsolni)	7500 Ft
C64 USER — CENTRONICS nyomtatókábel (GEOS kábel)	1500 Ft
256 K RAM-diszk (256 Kbyte RAM-mal)	14000 Ft
256 K RAM-diszk (64 Kbyte RAM-mal)	9000 Ft
2×64 Kbyte-os cartridge igény szerinti programokkal feltöltve	4.300 Ft
64 Kbyte-os cartridge igény szerinti programokkal feltöltve	3000 Ft
Epromok programozása meglévő programokkal, vagy saját hozott programok beégetésével 2716-tól 27512-ig az eprommal együtt egységesen	700 Ft
Árainkat az alkatrészárak változásai befolyásolhatják.	
A fenti bővítések megrendelhetők levélben az O.C.E. címén, valamint személyesen a havonta rendezendő klubdéllelőttön, ahol rendszeres bemutató is tartunk.	

Postázás esetén 100 Ft postaköltség kerül felszámításra.

A kedvezmény igénybeviteléhez az O.C.E. tagsági igazolvány bemutatása szükséges.

DECEMBERI 60 Ft-os vásárlási utalvány

Beváltható készpénzes vásárlás esetén a 2C Áruházban. Bp. XIII., Balzac u. 35.

Érvényes: 1992. december 31-ig.

HOBBI ELEKTRONIKA

DECEMBERI

vásárlási utalványa

Értéke:

**5000 Ft-ig 80Ft,
5000 Ft felett 10%**

Beváltható a Hobbi Elektronika Kft.-nél. Budapest VII., Dózsa György u. 16. Telefon: 122-8892

Egy személy részére egyszeri vásárláshoz egy utalvány használható fel!

Cserélhető lemezes winchester!

A SyQuest Technology
hivatalos magyarországi disztribútora:

NOVOTRADE
SZERVIZ Kft.

Cím: 1053 Budapest, Henszlmann I. u. 9.
Telefon: 117-4144 Telefax: 117-9692



SQ 555 (meghajtó)	39 900 Ft
SQ 400 (44 MB/lemez)	8 400 Ft
SQ 5110 (meghajtó)	61 400 Ft
SQ 800 (88 MB/lemez)	11 900 Ft

Áraink végfelhasználói árak és ÁFA-t nem tartalmaznak.

Meghatalmazott dealereink:

Microteam Kft., 1145 Budapest, Róna u. 127.
Tel./fax: 184-1226

Professzionál Kft., 1033 Budapest, Kaszásdűlő u. 5.
Tel.: 167-0024 fax: 167-0289

Professzionál Kft., Miskolci Kirendeltség,
3525 Miskolc, Szabó L. u. 37.
Tel.: 46/56-079

Professzionál Kft., Békéscsabai Kirendeltség,
5600 Békéscsaba, Andrássy u. 75.
Tel.: 66/28-584

Korall Kft., 2800 Tatabánya, Március 15. út 3.
Tel.: 34/11-714

3S Computer Kisszövetkezet, 6723 Szeged, Kemes u. 6.
Tel.: 62/26-277, Fax: 62/26-347

Volánelektronika Vercomp Kft., 9024 Győr, Dr. Petz Lajos u. 7.
Tel./fax: 96/12-520

Novotrade Miskolc Kft., 3530 Miskolc, Vörösmarty u. 51.
Tel./fax: 46/49-489

Elektrosoft Kft., 5000 Szolnok, József A. u. 6-8.
Tel.: 56/42-880, Fax: 56/44-222

Számadó Kft., 6000 Kecskemét, Dózsa Gy. u. 29.
Tel.: 76/21-455, Fax: 76/21-462

BX-Next Kft., 3434 Mályi, Bercsényi út 50.
Tel.: 46/91-117

Onyx Szoftver Kft., 1118 Budapest, Mányoki u. 14/B.
Tel.: 165-3325

Novotrade PC Kft., 1136 Budapest, Sallai u. 25.
Tel.: 149-0798 Fax: 131-0734

Almárium Kft., 1137 Budapest, Pozsonyi u. 21-23.
Tel.: 111-2830 Fax: 112-3647

Microchip Kft., 8000 Székesfehérvár, Élmunkás u. 47.
Tel.: 22/25-514

Navigátor Kft., 4400 Nyíregyháza, Kórház út 26/B.
Tel./fax: 42/41-972

Alfadat Kft., 2803 Tatabánya, Tóth-Bucsoki út 12.
Tel.: 34/10-234, 10-405, Fax: 34/10-729

Digitech Kft., 7101 Szekszárd, Rákóczi u. 6.
Tel.: 74/16-874

Interface Kft., 1116 Budapest, Hunyadi J. út 162.
Tel.: 166-5322/58, 55, Fax: 226-3793

Netrend Rt., 1089 Budapest, Elnök u. 1.
Tel.: 113-8217, Fax: 113-9537

Omnis Kft., 2840 Oroszlány, Münnich F. u. 23.
Tel.: 34/60-832

2R Periféria Kft., 1071 Budapest, Peterdy u. 30.
Tel.: 122-3034 Fax: 142-3308

ProComp Kft., 8900 Zalaegerszeg, Bíró M. út 8. Pf.: 275.
Tel.: 92/11-373

A magyar olimpiai csapat arany fokozatú támogatója





KARÁCSONYI AKCIÓ!

Amiga 500

38 900 Ft

Amiga 500+

44 900 Ft

Amiga 600

46 900 Ft

Amiga 2000

64 999 Ft

Commodore 64 II.

12 900 Ft

ACOMP Számítástechnikai Kft.

1141 Budapest, Álmos vezér útja 17. Telefon: 183-1817. Fax: 251-2523

Áraink az ÁFA-t tartalmazzák!

Nyitvatartás: 9—18 óráig. Szombaton: 9—14 óráig.
Vidéki vásárlóinknak utánvételes csomagküldő szolgálat!

A Computer Karácsonyon meglepetésekkel várjuk kedves vásárlóinkat!

Pinball Dreams Competition!!!

1992. december 12—13-án megrendezzük

az Első Országos Flipper Versenyt Amigán.

A fődíj egy eredeti Pinball Dreams II. Részletekről a helyszínen!