

Az Országos Commodore Egyesület lapja

újság

1993 / 1

BUÉK



1993



SZERENCSEJÁTÉK RT

10845

Commodore



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)



Több, mint tíz éve fókusz a Rádiótechnikában, vezetem a „Zenélő hobby elektronika” fejlécsű 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	Hangdigitizáló + 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 Kbajt)	2900 Ft	4000 Ft
RT91/10	IC tesztler	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 tesztler + szoftver	5900 Ft	9800 Ft
RT91/11,12	IBM EPROM-égető		
	+ szoftver	4800 Ft	9900 Ft
RT91/6	48 csat. I/O kártya	3500 Ft	

HE92/2	User-Centronics csatlakozó	950 Ft
RT92/3	TTL IC katalógus lemezen	600 Ft
	CMOS IC katalógus lemezen	600 Ft
	Dioda 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	Kiszajító 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	Rítmusgép 32 rítmussal	2500 Ft
RT90/10	Dobszintetizátor	3500 Ft
RT88/3	Süvöltő gitáreffekt	780 Ft
RT87/11	Ringmodulátor	995 Ft
RT92/4	Oktávemelő	700 Ft
	Visszhangosító MN3005-tel	4800 Ft

VIDÉÓS TÉMÁK:

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

FÉNYTECHNIKA:

RT88/10	Diszkofény II (triak nélkül)	1400 Ft
	Diszkofény 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	Kivezérlésjelző	700 Ft
RT91/8	Sziporkázó szingyürű	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 UM66Txx	480 Ft
HE91/4	Beszéd felismerő VCP200-zal	750 Ft
	VCP200	1950 Ft
HE91/2	Kutyugatás szintetizátor	800 Ft
RT91/8	Eb-vezérlő	400 Ft
HE91/6	Szűnyogriasztó	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
böngésződjé



Állandóan változó készlet!

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

Levélcím: 1656 Budapest Pf. 50.



SZERENCSEJÁTÉK RT

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.

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

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

PÖTYÖGŐSZOLGÁLAT: 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 rendelkezésre. 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 tikkett 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 Andre 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
Levélcím: 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 tagdíj 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 és
január 17-én.

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

Már régóta szerettem volna — és szerintem mások is szerettek volna — megírni egy olyan rutint, amely egy adott szöveget nagyított karakterekkel futó szöveg formájában jelenít meg. Nekem sikerült egy ilyet elkészítenem. A rutinnak megcsináltam a megszakítással (IRQ) vezérelt formáját is. Én elküldöm Önöknek mind a két verziót, és még hozzá egy másik IRQ-rutint, melynek működésekor egy csíkot látunk a képernyőn, amely összehúzódik, majd kitágul.

Most pedig beszéljünk a rutinokról:

A futó szöveg rutin sprite-okkal dolgozik, mégpedig mind a nyolc sprite-ot használja, ezért sajnos a rutin futása alatt más sprite-ok nem lehetnek a képernyőn.

Beállítások:

Be lehet állítanunk azt, hogy a sprite-ok melyik blokkokból vegyék az adatokat. Az egyetlen kikötés az, hogy a nyolc blokk egymás után következzen, ezért elég megadni a kezdő blokkot. A kezdő blokkot meg kell adnunk a 2015-ös sorban, de vigyázat ebben a sorban nem a blokk számát kell megadnunk, hanem a blokkterület kezdőcímeinek felső byte-ját. Én a 248-as blokkot választottam kezdőnek, így a 2015-ös sorban a 6-dik szám 62 lett ($248 * 64 / 256 = 62$). Meg kell adnunk a kezdő blokkot a 2055-ös sorban (2-dik szám) és a 2160-as sorban (6-dik szám). Az előbb említett sorokban azonban már a kezdő blokk számát (ez esetben 248) kell beírni.

Lehetőségünk van megadnunk a szöveg Y koordinátáját a 2065-ös sorban (2-dik szám).

A szöveg színét a 2040-es sorban (2-dik szám) adhatjuk meg. Egy érdekesség: ha a 2045-ös sorban a 7-dik számot 0-ra írjuk át, akkor laposabb karakterekkel fut a szöveg.

A rutin képes saját karakterkészletből álló betűket vagy jeleket megjeleníteni a képernyőn, de ehhez rá kell irányítani a saját betűkészletünkre. Az új karaktergenerátor kezdőcímeinek alsó és felső byte-ját a 295-ös sorban adhatjuk meg (2-ik, ill. 9-ik szám).

A program a 4123-as címtől kezdődik. Előtte vannak (4096-tól) azok a jelzőbyte-ok, amelyet a gép a rutin futása alatt használ. Számunkra négy jelzőbyte fontos: a 4096;4097, itt kell megadnunk a kiírandó tárterület kezdetének alsó és felső byte-ját. Ezenkívül fontos még a két másik jelzőbyte is: a 4098 és a 4099, ide a kiírandó terület végének alsó és felső byte-ját kell beírunk.

```

0 rem *****
1 rem * *
2 rem * futo felirat *
3 rem * *
4 rem * nagyított betukkel *
5 rem * *
6 rem * keszitette: d. g. *
7 rem *****
10 rem ** adatok beolvasasa **
15 v=53248:pokev+32,0:pokev+33,0:print"551"
20 print"adatbeolvasas ...":print"20kis turelmet kerek!"
30 fori=4115to4644:reada:pokei,a:s=s+a:nexti
40 ifs<>58871thenprint"223hiba az adatokban!":end
45 rem ***** takaras *****
47 print"567ime itt fut egy peldaszoveg : "
50 poke53248+27,255:rem a sprite-ok prioritasanak beallitasa a takarashoz
55 print"5555555555":rem a karakterszin beallitasa feketere
60 fori=1to3:print"2 ":nexti:rem 3 sornyi inverz nyomtatasa
1000 poke4096,59:poke4097,3:poke4098,216:poke4099,3:poke828,76:poke829,75
1005 a$="Hello !! Ez itt csak egy peldaszoveg,de barmit be lehet irni ehelyett."
1006 a$=a$+" Csak el kell helyezned a szoveget,es mar hivhatod is a "
1007 a$=a$+"rutint! " :rem szokozok a sz. teljes kifutasahoz
1010 fori=1to1en(a$)
1015 b$=mid$(a$,i,1)
1020 print"3";b$:b=peek(1024)
1025 poke827+i,b:nexti:print"4":sys4123
2000 rem ***** a d a t o k *****
2005 data4,52,100,148,196,244,36,84
2010 rem * blokkok torlese *
2015 data169,0,133,251,169,62,133,252
2020 data160,0,162,0,169,0,145,251,200,192,63,208,249
2025 data160,0,232,224,8,240,16,165,251,24,105,64,133,251,176,2,144,229,230,252
2030 data24,144,224
2035 rem * szinbeallitas,nagyitas *
2040 data169,7,162,0,157,39,208,232,224,8,208,248
2045 data169,255,141,29,208,169,255,141,23,208
2050 rem * sprite adatok a blokkokbol *
2055 data169,248,162,0,157,248,7,24,105,1,232,224,8,208,245
2060 rem * elhelyezes a kepernyon *
2065 data169,100,162,1,157,0,208,232,232,224,17,208,247,169,255,141,21,208
2070 data162,0,160,0,189,19,16,153,0,208,232,200,200,192,16,208,243
2075 data169,192,141,16,208
2080 rem * a jelzo byte-ok beallitasai *
2085 data169,0,162,0,157,4,16,232,224,6,208,248
2090 data141,12,16,141,13,16,141,14,16,141,16,16,169,1,141,10,16,141,11,16
2095 data141,15,16
2100 rem ** a rutin kezdete **
2105 rem * x koor csokkentese *
2110 data174,13,16,222,0,208,189,0,208,201,255,240,45
2120 data238,13,16,238,13,16,173,13,16,201,16,240,7
2122 data238,14,16,169,1,208,223
2123 data160,0,169,0,200,208,253,105,1,201,10,208,247
2124 data169,0,141,13,16,141,14,16,169,1,208,198
2125 rem * x=0 *
2130 data174,14,16,189,4,16,201,1,208,3,76,246,17
2135 rem * sprite hatratevese *

```



```

2136 data174,12,16,201,1,208,1,96
2137 data173,14,16,201,0,208,4,169,1,208,12
2140 data174,14,16,14,15,16,202,208,250
2145 data173,15,16,24,109,16,208,141,16,208,169,1,141,15,16,169,128
2147 data174,13,16,157,0,208
2150 data174,14,16,169,1,157,4,16
2155 rem * blokk torlese *
2160 data173,14,16,24,105,248,133,253,169,0,133,254
2165 data162,6,6,253,38,254,202,208,249
2170 data160,62,169,0,145,253,136,192,255,208,249
2175 rem * karakterek beolvasasa *
2180 data238,0,16,208,3,238,1,16,173,0,16,205,2,16,208,16,173,1,16,205,3,16
2185 data208,8,169,1,141,12,16,96,181,16
2190 data160,0,173,0,16,133,251,173,1,16,133,252,177,251,141,17,16
2195 :data169,0,133,251,141,18,16,169,216,133,252,14,17,16,46,18,16
2200 data14,17,16,46,18,16,14,17,16,46,18,16,165,251,24,109,17,16,133,251
2205 data144,2,230,252,165,252,24,109,18,16,133,252
2210 rem * blokkba iras *
2213 data120,169,51,133,1
2215 data165,253,141,210,17,165,254,141,211,17
2220 data174,16,16,160,0,177,251,157,0,0,232,232,232,200,192,8,208,243
2225 data238,16,16,173,16,16,201,3,240,3,76,89,17,169,0,141,16,16
2227 data169,55,133,1,88,76,183,16
2230 rem * utolso vizsgalat *
2235 data174,14,16,169,0,157,4,16
2295 data173,14,16,201,0,208,5,173,16,208,208,12
2296 data174,14,16,14,15,16,202,208,250
2297 data173,16,208,56,237,15,16,141,16,208,169,1,141,15,16,76,196,16

```

ready.

Az IRQ-val megoldott változatra ugyanezek vonatkoznak.

Ennyit akartam mondani erről a két rutinról, de most következzen a harmadik, aminek a feladatát már ismertettem. Ezt a rutint közvetlenül az előző nem megszakításos változat után tettem, hogy ha valaki a futó szöveg rutinnal szeretné használni, akkor ne kelljen bajlódni az eltolással.

Ez a rutin is tartalmaz jelzőbyte-okat. A 2010-es sorban a második és a 2080-as sorban az utolsó szám átírásával a csík mozgásának mértékét lehet megadni. (Nyilván, mind a két helyre ugyanaz a szám kell.)

A 2065-ös és a 2080-as sorban a 16-ik szám átírásával a csík

alsó részének mozgását határozhatjuk meg. Így például ajánlom a következő értékeket: 60, 60; 30, 30; 10, 10; 5, 5.

A csík Y koordinátáját a 2010-es sorban a 12-dik szám értékével lehet megadni, vagy ha már megy a rutin, akkor POKE 695, Y koordináta.

Van két dolog, ami jól jöhet: a POKE 696, 1 utasítással megmondjuk a gépnek, hogy a csík a lehető legkisebbre szűküljön össze, a másik az, hogy ha begépeljük a POKE697, 1 utasítást, akkor a kitágulást és az összeszűkülést megállítjuk. Természetesen ezeket a rutin futása alatt kell beírni, mert amikor meghívjuk, akkor a gép az összes jelzőbyte-ot alaphelyzetbe állítja. A

```

0 rem *****
1 rem *
2 rem *      futo felirat      *
3 rem *
4 rem *      megszakitással    *
5 rem *
6 rem * es nagyított betuvel *
7 rem *
8 rem *      készítette: d. g.  *
9 rem *****
10 rem ** adatok beolvasasa **
15 v=53248:pokev+32,0:pokev+33,0:print"33i"
20 print"adatbeolvasas ...":print"33kis türelmet kerek!"
30 fori=4115to4721:reada:pokei,a:s=s+a:nexti
40 ifs<>62812thenprint"333hiba az adatokban!":end
45 rem ***** takaras *****
47 print"333ime itt fut egy példaszöveg : "
50 poke53248+27,255:rem a sprite-ok proiritasanak beallitasa a takarashoz
55 print"3333333333 ":rem a karakterszín beallitasa feketere
60 fori=1to3:print"2  ":nexti:rem 3 sornyi inverz szokoz nyomtatasa
1000 poke4096,59:poke4097,3:poke4098,233:poke4099,3:poke828,56:poke829,75
1005 a$="Hello !! Ez a rutin most éppen IRQ-val van megoldva. Sajnos ennek is "
1006 a$=a$+"van egy kis hatranya. Peldaul a sebességet nem lehet túlzottan "
1007 a$=a$+"gyorsra allítani! "

```



```

1009 poke53280,0:poke53281,0
1010 for i=1 to len(a$)
1015 b$=mid$(a$,i,1):print" ";b$:b=peek(1024)
1025 poke827+i,b:next i:poke1024,14:poke56325,51:sys4123
1030 print"XXXXXXXXXXXX":end
2000 rem ***** a d a t o k *****
2005 data4,52,100,148,196,244,36,84
2010 rem * blokkok torlese *
2015 data169,0,133,139,169,62,133,140
2020 data160,0,162,0,169,0,145,139,200,192,63,208,249
2025 data160,0,232,224,0,240,16,165,139,24,105,64,133,139,176,2,144,229,230,140
2030 data24,144,224
2035 rem * szinbeallitas,nagyitas *
2040 data169,7,162,0,157,39,208,232,224,0,208,248
2045 data169,255,141,29,208,169,255,141,23,208
2050 rem * sprite adatok a blokkokbol *
2055 data169,248,162,0,157,248,7,24,105,1,232,224,8,208,245
2060 rem * elhelyezes a kepernyon *
2065 data169,100,162,1,157,0,208,232,232,224,17,208,247,169,255,141,21,208
2070 data162,0,160,0,189,19,16,153,0,208,232,200,200,192,16,208,243
2075 data169,192,141,16,208
2080 rem * a jelzo byte-ok beallitasai *
2085 data169,0,162,0,157,4,16,232,224,6,208,248
2090 data141,12,16,141,13,16,141,14,16,141,16,16,169,1,141,10,16,141,11,16
2095 data141,15,16
2100 rem ** a rutin kezdete **
2101 data169,0,133,2,141,12,16
2102 data120,169,203,141,20,3,169,16,141,21,3,88,96
2105 rem * x koor csokkentese *
2107 data165,2,201,1,240,41
2110 data174,13,16,222,0,208,189,0,208,201,255,240,60
2120 data238,13,16,238,13,16,173,13,16,201,16,240,7
2121 data238,14,16,169,1,208,223
2122 data169,0,141,13,16,141,14,16
2123 data169,1,133,2,162,0,232,208,253,238,12,16,173,12,16,201,1
2124 data240,3,76,49,234,169,0,133,2,141,12,16,76,49,234
2125 rem * x=0 *
2130 data174,14,16,189,4,16,201,1,208,3,76,29,18
2135 rem * sprite hatratevese *
2136 data174,12,16,201,1,208,1,96
2137 data173,14,16,201,0,208,4,169,1,208,12
2140 data174,14,16,14,15,16,202,208,250
2145 data173,15,16,24,109,16,208,141,16,208,169,1,141,15,16,169,128
2147 data174,13,16,157,0,208
2150 data174,14,16,169,1,157,4,16
2155 rem * blokk torlese *
2160 data173,14,16,24,105,248,133,141,169,0,133,142
2165 data162,6,6,141,38,142,202,208,249
2170 data160,62,169,0,145,141,136,192,255,208,249
2175 rem * karakterek beolvasasa *
2180 data238,0,16,208,3,238,1,16,173,0,16,205,2,16,208,16,173,1,16,205,3,16
2185 data208,8,169,1,141,12,16,76,76,18
2190 data160,0,173,0,16,133,139,173,1,16,133,140,177,139,141,17,16
2195 :data169,0,133,139,141,18,16,169,216,133,140,14,17,16,46,18,16
2200 data14,17,16,46,18,16,14,17,16,46,18,16,165,139,24,109,17,16,133,139
2205 data144,2,230,140,165,140,24,109,18,16,133,140
2210 rem * blokkba iras *
2213 data169,51,133,1
2215 data165,141,141,250,17,165,142,141,251,17
2220 data174,16,16,160,0,177,139,157,0,0,232,232,232,200,192,8,208,243
2225 data238,16,16,173,16,16,201,3,240,3,76,130,17,169,0,141,16,16
2227 data169,55,133,1,76,209,16
2230 rem * utolso vizsgalat *
2235 data174,14,16,169,0,157,4,16
2295 data173,14,16,201,0,208,5,173,16,208,208,12
2296 data174,14,16,14,15,16,202,208,250
2297 data173,16,208,56,237,15,16,141,16,208,169,1,141,15,16,76,222,16
2298 rem **
2299 data120,169,49,141,20,3,169,234,141,21,3,169,55,133,1,88,76,49,234
3000 data120,169,49,141,20,3,169,234,141,21,3,169,55,133,1,88,76,49,234

```

ready.

693-tól a 697-ig tartó byte-ok a rutin jelzőbyte-jai. Ha esetleg nekünk ez pont nem jó, akkor meg kell adnunk a rutinban, hogy a gép a futás alatt melyik byte-okat használja jelzőként.

Végül az utolsó beállítás: a csík nagyságát a 2035-ös sorban a 12-ik számmal határozhatjuk meg (itt 3).

Még két dolgot szeretnék mondani, amit kifejejtettem: mielőtt a futó szöveg programot beírnánk, toljuk el a basic kezdetet: POKE44,32:POKE8192,0:NEW.

Erre csak azért van szükség, mert a rutin a 4096-os címtől kezdődik, és a basic betöltője ennél tovább tartana, s így a vége megsérülne. A másik dolog amit még mondanék, az az, hogy

ha a betöltőprogramokban valamilyen számot átírtunk (ahogy azt leírtam), akkor az ellenőrző összegeket is ennek megfelelően változtassuk meg (vagy töröljük ki az ellenőrzést), különben a gép kiírja, hogy hiba van az adatokban.

Úgy érzem, hogy a rutinok működéséről mindent részlete-sen elmondtam.

Ha programjaim megnyerték tetszésüket, akkor kérem ha lehet, tegyék közé őket a COMMODORE újságban.

Tisztelettel:

Diós Gábor

```

0 rem *****
1 rem * *
2 rem * kepernyocsik *
3 rem * *
4 rem * irq - rutin *
5 rem * *
6 rem * keszitette: d.g. *
7 rem *****
8 poke53280,0:poke53281,0
10 fori=4645to4826:reada:pokei,a:s=s+a:nexti
20 ifs<>21062thenprint"hiba az adatokban!":end
25 print"szoborok"
30 print"ez hat az az irq-rutin!":poke53280,0:poke53281,0
40 sys4645:print" ":end
2000 rem **** a d a t o k ****
2010 data169,241,141,181,2,169,1,141,182,2,169,100,141,183,2
2015 data120,169,81,141,20,3,169,18,141,21,3
2020 data173,26,208,9,1,141,26,208,173,14,220,41,254,141,14,220
2023 data88,96
2025 data173,17,208,41,127,141,17,208
2027 data173,25,208,41,1,141,25,208
2030 data169,2,141,32,208,141,33,208
2033 :data173,184,2,201,1,240,14
2035 data162,0,169,0,232,208,253,24,105,1,201,3,208,247
2040 data162,0,232,236,181,2,208,250
2045 data169,0,141,32,208,141,33,208
2047 data173,185,2,201,1,208,3,76,49,234
2050 :
2055 data173,182,2,201,1,208,30
2060 :
2065 data238,183,2,173,183,2,141,18,208,173,181,2,56,233,15,141,181,2,201,1
2070 data208,5,169,2,141,182,2,76,49,234
2075 :
2080 data206,183,2,173,183,2,141,18,208,173,181,2,24,105,15,141,181,2,201,241
2085 data208,5,169,1,141,182,2,76,49,234
    
```

ready.

```

10 POKE 19,1:GOTO 40
20 A=INT((10-2)*RAND(1)+2)
30 RETURN
40 GOSUB 20:X=A:GOSUB 20:Y=A
50 PRINT"X*Y="";:INPUT S$
55 S=VAL(S$):PRINT:IF S<1 OR S>100 THEN 70
60 IF S=X*Y THEN PRINT"OKOS VAGY !":GOTO 100
70 PRINT"CSACSI VAGY !!!! EZ AZ O.K. > X*Y
100 PRINT"<F-1>"
110 WAIT 198,1:GOTO 40
    
```

READY.

Tisztelt Szerkesztőség

Szorzótabla helyett

Bizonyára más „általános iskolás apukának” is gondot okoz a szorzótábla gyakoroltatása. Nagyon nehéz ötletszerűen példákat gyártani. Ezért „rittyentettem” ezt a kis rutint, amely véletlenszámok generálásával állítja elő a szorzót és a szorzandót 2 és 9 között. A beírt választ értékeli, és minősíti: „okos vagy”, „csacsi vagy”. Az utóbbi esetben megadja a helyes eredményt is.

Tisztelt szerkesztőség

Dec./hex./dec. konvertáló

Gyakorta, de különösen gépi kódú programozás során, szükségünk lehet a decimális és hexadecimális számok egymásba konvertálására. Ehhez nyújt segítséget a lista szerinti program.

Menü segítségével választhatjuk ki a megfelelő konvertálási „irányt”. A beírt és konvertált értékeket kinyomtatjuk. Ennek akkor lehet jelentősége, ha pl. egy rutint, vagy teljes programot szeretnénk konvertálni.

Ezen túlmenően lehetőség van a decimális számok 0 és 255 közötti kinyomtatására.

Az 1-es, és a 2-es üzemmódból, a „0” érték beírásával térhünk vissza a főmenühöz.

A program a #0—#65535, valamint a \$0—\$FFFF közötti értékeket tudja kezelni.

Szász Sándor

```

10 POKE 53280,9:POKE 53281,8:PRINT "CHR$(142)CHR$(8):POKE 808,234:GOTO 80
20 IF N=0 THEN RETURN
30 PRINT#4," ";:FOR L=1 TO L1:PRINT#4,"=";:NEXT L:PRINT#4:RETURN
40 GOSUB 70:PRINTTAB(9)"  GOOD BYE KONVERTALO! ":T=TI
50 IF TI-T<300 THEN 50
60 SYS 64738
70 FOR L=17 TO 20:POKE 781,L:SYS 59903:NEXT L:POKE 214,17:SYS 58640:RETURN
80 Q(1)=4096:Q(2)=256:Q(3)=16:Q(4)=1
90 N=0:PRINT " * DECIMALIS/HEXADECIMALIS KONVERTALO *"
100 PRINTTAB(14)"> VALASSZON <>:PRINTTAB(10)"1.DECIM.-ROL HEX.-RE
110 PRINTTAB(10)"2.HEX.-ROL DECIM.-RE:PRINTTAB(10)"3.HEX.TABLAZAT PRINT"
120 PRINTTAB(12)"DECIMALIS 0-255-IG:PRINTTAB(10)"4.V E G E !
130 GOSUB 70:PRINT" * * * * * "
140 PRINT" * LIMIT: #0-#65535 * $0-$FFFF *"
150 WAIT 198,1:GET W$:W=VAL(W$)
160 IF W=0 OR W>4 THEN 150
170 IF W=4 THEN 40
180 GOSUB 70:IF W=3 THEN 220
190 PRINT TAB(8)"KER NYOMTATAST? IGEN/NEM."
200 GETW$:IF W$<>"I" AND W$<>"N" THEN 200
210 IF W$="N" THEN 260
220 GOSUB 70:PRINT TAB(5)"A NYOMTATO UZEMKESZ? IGEN/NEM."
230 GETW$:IF W$<>"I" AND W$<>"N" THEN 230
240 IF W$="N" THEN 80
250 OPEN4,4:N=1
260 ON W-1 GOTO 400,550
270 IF N=0 THEN 290
280 L1=28:GOSUB 20:PRINT#4," DECIMALISROL HEXADECIMALISRA"CHR$(13)
290 GOSUB 70:POKE 19,1:INPUT" KEREM A DECIMALIS SZAMOT: # ";A$:POKE 19,0:PRINT
300 A=VAL(A$):IF A=0 THEN GOSUB 20:CLOSE4:GOTO 90
310 IF A>65535 THEN 290
320 IF N=1 THEN PRINT#4," #"RIGHT$("0000"+MID$(STR$(A),2),5)" = ";
330 A$="$":FOR L=1 TO 4:B=INT(A/Q(L)):A=A-(B*Q(L))
340 X=55:IF B<10 THEN X=48
350 A$=A$+CHR$(B+X):NEXT L
360 PRINT" ITOVABB:<F-7>"SPC(13)" "A$ "
370 IF N=1 THEN PRINT#4,A$:GOTO 390
380 GET W$:IF W$<>CHR$(136) THEN 380
390 A=0:A$="$":GOTO 290
400 IF N=0 THEN 420
410 L1=28:GOSUB 20:PRINT#4," HEXADECIMALISROL DECIMALISRA"CHR$(13)
420 GOSUB 70:POKE 19,1:INPUT" KEREM A HEXADECIM.SZAMOT: $ ";A$:POKE 19,0:PRINT
430 IF LEN(A$)>4 THEN 420
440 IF A$="0"THEN GOSUB 20:CLOSE4:GOTO 90
450 A$=RIGHT$("0000"+A$,4)
460 FOR L=1 TO 4:X=ASC(MID$(A$,L,1))
470 IF X>47 AND X<58 THEN B=48:GOTO 500
480 IF X>64 AND X<71 THEN B=55:GOTO 500
490 PRINT" IDEGEN KARAKTER !! <F-7>":L=4:ER=1
500 A=A+(X-B)*Q(L):NEXT L:IF ER=1 THEN ER=0:GOTO 530
510 PRINT" ITOVABB:<F-7>"SPC(13)" # "RIGHT$("0000"+MID$(STR$(A),2),5)" "
520 IF N=1 THEN PRINT#4," $"A$ = #"RIGHT$("0000"+MID$(STR$(A),2),5):GOTO 540
530 GET W$:IF W$<>CHR$(136) THEN 530
540 A$="":A=0:GOTO 420
550 A=10:F=100:A$="$":GOSUB 70:PRINTTAB(8)" NYOMTATOM A TABLAZATOT
560 L1=64:GOSUB 20:PRINT#4," ****";:FOR L=A TO F STEP 10
570 PRINT#4," #"RIGHT$("00"+MID$(STR$(L),2),3);:NEXT L:PRINT#4,CHR$(13)
580 FOR Q=0 TO 9:PRINT#4," #"RIGHT$("00"+MID$(STR$(Q),2),3);
590 FOR L=A TO F STEP 10:B=Q+L:IF B>255 THEN 640
600 FOR Y=1 TO 2:C=INT(B/Q(Y+2)):B=B-(C*Q(Y+2))

```




```

610 X=55:IF C<10 THEN X=48
620 A$=A$+CHR$(C+X):NEXT Y
630 PRINT#4," "A$;
640 A$=" ":NEXTL:PRINT#4:NEXT Q:PRINT#4
650 A=110:F=200:IF L<200 THEN 560
660 A=210:F=255:IF L<255 THEN 560
670 GOSUB 20:CLOSE4:GOTO 90

```

READY.

=====
DECIMALISROL HEXADECIMALISRA

```

#65535 = $FFFF
#00666 = $029A
#00888 = $0378
#00999 = $03E7
#12345 = $3039

```

=====
HEXADECIMALISROL DECIMALISRA

```

$FFFF = #65535
$000F = #00015
$00FF = #00255
$0FFF = #04095
$FFFF = #65535

```

=====
**** #010 #020 #030 #040 #050 #060 #070 #080 #090 #100

#000	\$0A	\$14	\$1E	\$28	\$32	\$3C	\$46	\$50	\$5A	\$64
#001	\$0B	\$15	\$1F	\$29	\$33	\$3D	\$47	\$51	\$5B	\$65
#002	\$0C	\$16	\$20	\$2A	\$34	\$3E	\$48	\$52	\$5C	\$66
#003	\$0D	\$17	\$21	\$2B	\$35	\$3F	\$49	\$53	\$5D	\$67
#004	\$0E	\$18	\$22	\$2C	\$36	\$40	\$4A	\$54	\$5E	\$68
#005	\$0F	\$19	\$23	\$2D	\$37	\$41	\$4B	\$55	\$5F	\$69
#006	\$10	\$1A	\$24	\$2E	\$38	\$42	\$4C	\$56	\$60	\$6A
#007	\$11	\$1B	\$25	\$2F	\$39	\$43	\$4D	\$57	\$61	\$6B
#008	\$12	\$1C	\$26	\$30	\$3A	\$44	\$4E	\$58	\$62	\$6C
#009	\$13	\$1D	\$27	\$31	\$3B	\$45	\$4F	\$59	\$63	\$6D

=====
**** #110 #120 #130 #140 #150 #160 #170 #180 #190 #200

#000	\$6E	\$78	\$82	\$8C	\$96	\$A0	\$AA	\$B4	\$BE	\$C8
#001	\$6F	\$79	\$83	\$8D	\$97	\$A1	\$AB	\$B5	\$BF	\$C9
#002	\$70	\$7A	\$84	\$8E	\$98	\$A2	\$AC	\$B6	\$C0	\$CA
#003	\$71	\$7B	\$85	\$8F	\$99	\$A3	\$AD	\$B7	\$C1	\$CB
#004	\$72	\$7C	\$86	\$90	\$9A	\$A4	\$AE	\$B8	\$C2	\$CC
#005	\$73	\$7D	\$87	\$91	\$9B	\$A5	\$AF	\$B9	\$C3	\$CD
#006	\$74	\$7E	\$88	\$92	\$9C	\$A6	\$B0	\$BA	\$C4	\$CE
#007	\$75	\$7F	\$89	\$93	\$9D	\$A7	\$B1	\$BB	\$C5	\$CF
#008	\$76	\$80	\$8A	\$94	\$9E	\$A8	\$B2	\$BC	\$C6	\$D0
#009	\$77	\$81	\$8B	\$95	\$9F	\$A9	\$B3	\$BD	\$C7	\$D1

=====
**** #210 #220 #230 #240 #250

#000	\$D2	\$DC	\$E6	\$F0	\$FA
#001	\$D3	\$DD	\$E7	\$F1	\$FB
#002	\$D4	\$DE	\$E8	\$F2	\$FC
#003	\$D5	\$DF	\$E9	\$F3	\$FD
#004	\$D6	\$E0	\$EA	\$F4	\$FE
#005	\$D7	\$E1	\$EB	\$F5	\$FF
#006	\$D8	\$E2	\$EC	\$F6	
#007	\$D9	\$E3	\$ED	\$F7	
#008	\$DA	\$E4	\$EE	\$F8	
#009	\$DB	\$E5	\$EF	\$F9	

=====



Tisztelt Szerkesztőség

Mellékelten küldök két programot közzétételre, amennyiben azt arra érdemesnek találják.

Az első — „A háromszög köré írható kör” — programot Pethes Endre hasonló című Plus/4-re írt programja alapján írtam át COMMODORE 64-re.

A második — „A háromszögbe írható kör” — már úgy született, hogy kedvet kaptam ennek megoldására is.

Mindkét program „SIMON'S BASIC”-ben működik.

A második program számadatainak bevitelekor ügyelni kell

arra, hogy a háromszögpontok, az óramutatóval ellentétes irányúak legyenek, egyébként pedig az adatpárok ciklikusan változtathatók.

A jobb megértésre egy példa:

— bevihető ciklikusan az alábbi

15,40	→	100,110	→	145,65	vagy
100,110	→	145,65	→	15,40	vagy
145,65	→	15,45	→	100,110	értékek.

Ha tehát beírtuk a háromszög-csúcs két pontját, a harmadik csúcspont a forgásirány szerinti lehet.

Czakó János

```

10 REM *****
20 REM * ALTALANDOS HAROMSZOG KORE *
30 REM * RAJZOLHATO KOR *
40 REM * SUGARA ES KOORDINATAI *
50 REM *****
70 :
100 PRINT" (CLR) "
110 INPUT "(DOWN) (RGHT) EGYIK CSUCS (A1,A2)";A1,A2
120 INPUT "(DOWN) (RGHT) MASIK CSUCS (B1,B2)";B1,B2
130 INPUT "(DOWN) (RGHT) HARM. CSUCS (C1,C2)";C1,C2
140 HIRSI14,6:POKE53280,14
150 LINEA1,A2,B1,B2,1
160 LINEB1,B2,C1,C2,1
170 LINEC1,C2,A1,A2,1
180 V1=B1-A1:V2=B2-A2:V=SQR(V1^2+V2^2):V5=-V2:V6=V1
190 U1=C1-A1:U2=C2-A2:U=SQR(U1^2+U2^2):U5=-U2:U6=U1
200 Z1=C1-B1:Z2=C2-B2:Z=SQR(Z1^2+Z2^2):Z5=-Z2:Z6=Z1
210 F1=(A1+B1)/2:F2=(A2+B2)/2:F3=(A1+C1)/2:F4=(A2+C2)/2
220 T=(U5*(F4-F2)+U6*(F1-F3))/(V6*U5-V5*U6)
230 M1=F1+V5*T:M2=F2+V6*T
240 R1=SQR((M1-A1)^2+(M2-A2)^2)
250 O1=INT(M1*100)/100:O2=INT(M2*100)/100:R=INT(R1*100)/100
260 CIRCLEO1,O2,R,R,1
270 LINEO1,O2,10,O2,1
280 LINEO1,O2,O1,10,1
290 LINEO1,O2,309,O2,1
300 LINED1,O2,O1,189,1
310 WAIT203,63
320 PRINT"(DOWN) A KOR SUGARA : R=";R
330 PRINT"(DOWN) KOZEPPONT KOORD. : X=";O1
340 PRINT"(DOWN) : Y=";O2
350 END
    
```

READY.

PELDA

```

EGYIK CSUCS (A1,A2)? 80,100
MASIK CSUCS (B1,B2)? 170,160
HARM. CSUCS (C1,C2)? 215,65
A KOR SUGARA : R= 70.47
KOZEPPONT KOORD. : X= 150.06
: Y= 92.4
    
```

READY.

HRDCPY


```

10 REM *****
20 REM * ALTALANOS HAROMSZOGBE *
30 REM * RAJZOLHATO KOR *
40 REM * SUGARA ES KOORDINATAI *
50 REM *****
60 :
100 POKE53280,14:POKE53281,6:POKE646,14
110 PRINT"(CLR)":PRINT"(DOWN)(RGHT)MEGADHATO ERTEKEK : 0<X<319 ; 0<Y<199
120 C$="ERTEKPAROK BEVITELE ORAJARASSAL ELLEN- "
130 A$=" TETESEN ES CIKLIKUS SORRENDEN !"
140 PRINT"(DOWN)(RGHT)"C$+A$
150 INPUT"(DOWN)(RGHT) X1,Y1 :";X1,Y1
160 INPUT"(DOWN)(RGHT) X2,Y2 :";X2,Y2
170 INPUT"(DOWN)(RGHT) X3,Y3 :";X3,Y3
180 IFX1<0ORX1>319ORX2<0ORX2>319ORX3<0ORX3>319THEN210
190 IFY1<0ORY1>199ORY2<0ORY2>199ORY3<0ORY3>199THEN210
200 GOTO 220
210 PRINT"(CLR)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)ROSSZ
AZ ADAT !!!":END
220 A=SQR((X1-X3)^2+(Y1-Y3)^2)
230 B=SQR((X1-X2)^2+(Y1-Y2)^2)
240 C=SQR((X2-X3)^2+(Y2-Y3)^2):S=(A+B+C)/2
250 U=SQR((S-B)*(S-C)/(B*C)):AL=2*ATN(U/SQR(1-U^2))
260 V=SQR((S-A)*(S-C)/(A*C)):BE=2*ATN(V/SQR(1-V^2))
270 W=SQR((S-A)*(S-B)/(A*B)):GA=2*ATN(W/SQR(1-W^2))
280 M1=(Y3-Y1)/(X3-X1):M2=(Y2-Y1)/(X2-X1):M3=(Y3-Y2)/(X3-X2)
290 F1=ATN(M1)+GA/2:FE=TAN(F1)
300 F2=ATN(M2)+AL/2:FM=TAN(F2)
310 F3=ATN(M3)+BE/2:FH=TAN(F3)
320 X=((FE*X1+Y2)-(FM*X2+Y1))/(FE-FM)
330 Y=FE*(X-X1)+Y1
340 YB=FM*(X-X2)+Y2
350 YC=FH*(X-X3)+Y3
360 R=SQR((S-A)*(S-B)*(S-C)/S)
370 PRINT"(DOWN)(RGHT)A BELSO KOR SUGAR : R=";INT(R*100)/100
380 PRINT"(DOWN)(RGHT)KOZEPPONT KOORDINATAK: X=";INT(X*100)/100
390 PRINT"(DOWN) Y=";INT(Y*100)/100
400 PRINT"(DOWN)(RGHT)(RVS) ELLENORZES : YB ES YC LEKERDEZESE. (OFF)"
410 WAIT 203,63
420 HIRES 14,6
430 LINE X1,Y1,X2,Y2,1
440 LINE X1,Y1,X3,Y3,1
450 LINE X2,Y2,X3,Y3,1
460 CIRCLE X,Y,R,1
470 LINE 5,Y,315,Y,1
480 LINE X,5,X,195,1
490 WAIT 203,63

READY.

```

PELDA

MEGADHATO ERTEKEK : 0<X<319 ; 0<Y<199

ERTEKPAROK BEVITELE ORAJARASSAL ELLEN-
TETESEN ES CIKLIKUS SORRENDEN !

X1,Y1 :? 15,40

X2,Y2 :? 100,110

X3,Y3 :? 145,65

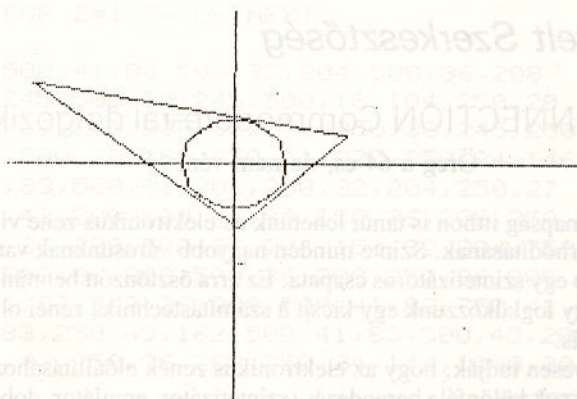
A BELSO KOR SUGAR : R= 22.78

KOZEPPONT KOORDINATAK: X= 98.51

Y= 79.26

READY.

HRDCPY



COPY

Tisztelt Szerkesztőség

Kisfelbontású grafika Commodore számítógépekhez

Durvább rajzok, egyszerűbb függvényábrázolások megjelenítéséhez a 80×50-es grafika is elfogadható eredményt ad. Használatánál a számítógép karakteres üzemmódban dolgozik. Ez azzal az előnnyel jár, hogy rajzaink mellett a gép teljes karakterkészletét a szokásos módon használhatjuk, továbbá a bittérképes üzemmódbhoz képest közel 8 kbyte-al több RAM áll rendelkezésünkre, amely különösen a C-16 típusnál jelentős.

A C-64-hez Liesert PEEK-ek és POKE-ok a C-64-esen c. könyve ismerteti egy kisfelbontású grafikát lehetővé tevő szubrutint. Hiányossága, hogy a képernyőkódok átalakításához szükséges terjedelmes táblázatot tömbben helyezi el, amely természetesen azzal a hátránnyal jár, hogy RUN parancsra vagy a főprogram szerkesztése esetén az adatok elvesznek. Ilyenkor kezdhetjük előlrol a tömb feltöltését, amely a programkészítés munkáját lassúvá és nehézkesé teszi.

A következőkben ismertetendő szubrutin az említett hibától mentes. A 60060 sz. sor végén található szám — a képernyőmemória kezdőcíme — értelemszerű változtatásával a C-64-en kívül a többi géptípusra is alkalmazható. A kódszámmodosításhoz csupán 16 adatot használ, ezek a szubrutin első hívását követően a memória valamely szabad területére kerülnek, esetünkben a kazettapufferbe, a 828. címtől kezdődően. Ha a táblázatot más memóriaterületen kívánjuk elhelyezni, akkor a 60000 sz. sorban a BB változó értékadását kell annak megfelelően módosítani.

A működés azon alapul, hogy az eredeti grafikus képernyőkódszámokat az említett táblázat segítségével egy olyan közbülső kódba alakítja (0...15), amelyben a szükséges módosítások egyszerű logikai műveletekkel megoldhatók. Az ily módon kapott új közbülső kódszámokhoz tartozó eredetit visszakeresve előáll az a szám, amelyeket POKE utasítással a képernyőmemória megfelelő címére helyezve a kívánt változás a monitoron megvalósul. A főprogram elkészítésekor vegyük figyelembe, hogy a 0, 0 koordinátájú pont a képmező bal alsó sarkába kerül. Egy képpont megjelenítéséhez a VV változónak 1. törléséhez 0 érté-

```

○ 59910 REM *****
59920 REM * 80*50 GRAFIKA SZUBRUTIN *
59930 REM * C 64 (16,PLUS4) *
○ 59940 REM * *
59950 REM * MEGJELENITES:VV=1 *
○ 59960 REM * TORLES: VV=0 *
59970 REM * X: 0...79 Y: 0...49 *
59980 REM * HIVASA: GOSUB 60000 *
○ 59982 REM * *
59984 REM * PA'L L. 1991. *
59986 REM *****
○ 59990 :
60000 BB=828:IF PEEK(BB)=32 THEN 60030
60010 FOR I=BB TO BB+15
60020 READ AA:POKE I,AA:NEXT
○ 60030 IFX>79 OR X<0 THEN RETURN
60040 IFY>49 OR Y<0 THEN RETURN
60050 X=INT(X):Y=INT(Y):FF=0
○ 60060 CC=INT(X/2)+40*INT((49-Y)/2)+1024
60070 REM CSERE:C 16,PLUS/4-NEL:3072↑
60080 DD=X AND 1:EE=Y AND 1:HH=PEEK(CC)
○ 60090 IF DD AND EE THEN UU=8:GOTO60130
60100 IF NOT DD AND EE THEN UU=4:GOTO60130
60110 IF DD AND NOT EE THEN UU=2:GOTO60130
○ 60120 UU=1
60130 FORI=0 TO 15
○ 60140 IF PEEK(I+BB)=HH THEN GOSUB 60210
60150 NEXT
60160 IF VV=0 AND FF=0 THEN RETURN
○ 60170 FOR I=0 TO 15
60180 IF UU=I THEN POKE CC,PEEK(I+BB):I=15
○ 60190 NEXT:RETURN
60200 :
60210 IF VV>0 THEN UU=I OR UU:GOTO60230
○ 60220 UU=I AND NOT UU
60230 I=15:FF=1:RETURN
○ 60240 :
60250 DATA 32,123,108,98,126,97,127,252
60260 DATA 124,255,225,254,226,236,251
○ 60270 DATA 160

```

ket kell adni. Programjaink futtatása gyorsabbá, mutatósabbá tehető, ha az ábrázolás fix részeit pl. a koordináta-tengelyeket nem pontonként a szubrutinnal rajzoljuk, hanem a főprogramban elhelyezett PRINT utasításokkal hozzuk létre.

Pál László

Tisztelt Szerkesztőség

A CONNECTION Commodore-ral dolgozik

Öreg a 64-es, de nem vén!

Manapság itthon is tanúi lehetünk az elektronikus zene viharos térhódításának. Szinte minden nagyobb városunknak van legalább egy szintetizátoros csapata. Ez arra ösztönzött bennünket, hogy foglalkozzunk egy kicsit a számítástechnika zenei oldalával is.

Kevesen tudják, hogy az elektronikus zenék előállításához használt sok különféle berendezés (szintetizátor, emulátor, dob-gép, szekvenszer... stb.) mind-mind egy speciális számítógép, melynek a zenei oldalát fejlesztették elég magas szintre. Ezekben is ugyanúgy megtalálhatjuk a számítógép alap elemeit: a processzort, a memóriákat, illesztő egységet és a (speciálisan ki-

képzett) billentyűzetet. Sőt, némelyikhez lemezegység és kijelző monitor is tartozik. Éppen ezekből adódóan remekül tudnak egymással és más számítógépekkel kommunikálni. Ezt a kommunikációs rendszert MIDI névvel illették meg tervezői.

A fenti kijelentésünket igyekezünk alátámasztani az a tény is, miszerint egyre több MIDI-interface és program kerül forgalomba C64-re, AMIGA-ra és egyéb más géptípusokra. Jó-jó, mondhatja a kedves olvasó, de ezek csupán a száraz, puszta tények, megállapítások. Mi a helyzet a valóságban? Kik és hogyan alkalmazzák a házi számítógépet a zenében? Ennek igyekeztünk utánajárni.

Végül is a zalaegerszegi—nagykanizsai illetőségű ELECTRIC CONNECTION névre hallgató szinti-pop zenekarnál kaptunk kielégítő választ kérdéseinkre.

— Te vagy a csapat egyik alapítótagja és — sajátos elnevezésekkkel élve — billentyűzatorosa, Szok István alias: Smog. Ha jól tudom, 1990 őszén alakultatok. Akkoriban már igen népszerű volt Magyarországon is a számítástechnika, a számítógépek al-

kalmazása. Mégis, hogyan jutott eszetekbe, hogy „computer” zenét játsszatok? Már régebben is foglalkoztatok számítógépekkel?

— Szerencsére Zalaegerszeg zenei aktivitás terén nem egy passzív városka. Sok ember foglalkozik zenével és számítástechnikával és egynéhány párosította a kellemeset a haszonnal. Elsősorban rajtuk keresztül ismerkedtem meg a számítógép zenében való alkalmazásával. Persze a gyakorlati alkalmazás még várattott magára. Az is igaz, hogy még zenei hajnalom idejétaját a MICRO-sorozat dobprogramjait használtuk, de a gépeink hangzása véget vetett szerencsétlen programcsalád karrierjének.

— Mikor, mivel és hol kezdted? Ma mire és hogyan használjátok a computer technikát? Milyen a hangszer- és gépparkotok jelenleg?

— Mint mondtam néhány éve kezdtük (a fantasztikus játéktékprogram mellett) használni a computert. Akkoriban a számítógép hangchipseinek produkcióját alkalmaztuk, ma már „csak” vezérlésre ún. sequencerezésre használjuk. Otthon mindössze egy C64-gyel + KAWAI K4-gyel és külön egy ROLAND D20-szal dolgozunk. Koncertekre ezeket egészítjük még ki egy-két hangszerrel.

— Milyen irányú és milyen konkrét fejlesztéseket terveztek a jövőt illetően?

— Elsősorban újabb programok beszerzése a célunk, hogy ezzel is bővítsük a lehetőségeinket a zene komponálásban.

— Mi a software-oldala, igénye a ti zenétekben? Tíetek a programozás feladata is vagy csak felhasználók vagytok?

— Általában ezek a programok nagyon jól meg vannak szerkesztve, ezért sem foglalkozunk a programozással. Legfőbb elvárásaink egy programmal szemben például a kezelhetőség, nagy adatkapacitás és persze a gyorsaság.

— Az anyagiaktól most elvonatkoztatva, milyen számítógéppel, -gépekkel dolgoznátok szívesen és miért?

— Nem Commodore-ral, ezért ne feszegezzük a témát!

— Az ELECTRIC CONNECTION-féle zenék rögzítésére leginkább egy teljesen digitális stúdió a legoptimálisabb. Mi erről a véleményed, álláspontod és ti hogyan csináljátok?

— Tul'képpen a mai technikai szinten minden előadót egy tökéletes hangrögzítés orientál, hogy művét megpróbálja olyan vagy jobb minőségben rögzíteni, hogy aztán a hallgató az előadás eredeti mivoltában élvezhesse a produkciót. Ennek egyik gyakorlata egy DIGSTÚDIÓ. A mi lehetőségeink nem elegendők (egyelőre), hogy ilyen körülmények között dolgozhassunk. Tehát egyelőre marad az az egy pár módszer, amivel azért lehet valamit kezdeni.

— Akkor hát miért választottátok mégis ezt a — szó szerint — megrázó stílust? Annyira olcsók a szintik?

— Hm.

— Attól tartok, igencsak sokat kell még dolgoznotok ahhoz, hogy egy saját digitális stúdiót tudjatok fölállítani és fenntartani, de biztos vagyok benne, hogy az idő és a rajongók titeket fognak igazolni! Sok sikert! Köszö a beszélgetést!

—Maci—

It's a small world

Mindenki ismeri a fenti című dalocskát? Ha nem, az se baj, az alábbi kis program aránylag elfogadható minőségben lejátssza.

```

0 REM KIS VILAG
10 S=54272:FOR SW=S TO S+24:POKE SW,0:NEXT
20 POKE S+24,15:POKE S+2,255:POKE S+3,0:POKE S+5,9:POKE S+6,0
30 READ N1,N2,DR
40 IF N1=0 THEN PRINT"VEGE !":END
50 POKE S+1,N1:POKE S,N2:POKE S+4,65:FOR Z=1 TO DR:NEXT
60 POKE S+4,64:GOTO 30
70 DATA 20,169,250,21,227,250,24,146,500,41,83,500,32,204,500,36,208
80 DATA 250,32,204,250,32,204,500,30,245,500,30,245,500,18,104,250,20
90 DATA 169,250,21,227,500,36,208,500,30,245,500,32,204,250,30,245,250
100 DATA 27,148,500,24,146,500,24,146,500,20,169,250,21,227,250,24,146
110 DATA 500,32,204,250,36,208,250,41,83,500,36,208,250,32,204,250,27
120 DATA 148,500,36,208,250,41,83,250,43,200,500,41,83,250,36,208,250,24
130 DATA 146,500,43,200,500,41,83,500,36,208,500,32,204,150,32,204,750
140 DATA 32,204,250,41,83,500,43,200,500,36,208,500,36,208,250,36,208
150 DATA 1000,36,208,750,36,208,250,41,83,500,36,208,500,41,83,750,41,83
160 DATA 250,41,83,1000,41,83,750,41,83,250,49,162,500,41,83,500,43,200
170 DATA 750,43,200,250,43,200,500,41,83,250,36,208,250,24,146,1000,30
180 DATA 245,1000,32,204,1500,0,0,0

READY.

```


Tippek-trükkök C-64-hez

Színváltás soronként

Rendes körülmények között egy 64 pixelből (8×8) álló mezőben maximum négy színt használhatunk. Csak abban az esetben nőhet ez a szám, ha a mező fölé egy sprite-ot fektetünk. Vajon más megoldás nincs? Persze hogy van. Egy trükk segítségével, amely azonban a C64 esetében mintegy 64 százalékos futási sebességsökkenést eredményez, egy 8×8-as mátrixban akár tizenhat színt is használhatunk. Ezt az úgynevezett FLI mód (lásd a magyarázatot lentebb) használatával érhetjük el.

Ennek a rutinnak az időviszonyai azonban olyan kritikusak, hogy az alkalmazásnál a képernyőn egy szem sprite sem állhat! Nem használhatjuk így egy sor első három karakterét sem, mert a VIC-nek ennyi idő kell, míg átáll az új video RAM-ra.

A multicolor módban egy szín minden sorban egy 8×8-as mezőben konstans. A többi színt a multicolor módhoz hasonlóan használhatjuk. Az FLI plusz színeit kiválóan használhatjuk főcímek, feliratok vagy egyéb, idő szempontjából „stabil” alkalmazásokban.

De hogy működik az FLI? Ahhoz hogy minden sort külön-külön kiszínezhessünk, a VIC-nek minden sorban hivatkoznia kell a video RAM-ra: A multicolor módban a színeket a video RAM-ból és a szín RAM-ból olvassuk. Persze igen nehéz volna a video RAM-ot minden esetben kimásolni, arról nem is szólva, hogy ehhez nincs elegendő idő sem. Van azonban megoldás! Nem másoljuk be az adatokat a video RAM-ba, hanem a video RAM-ot helyezzük át. Ezt a \$D018-as regiszter manipulálásával

```

1 REM *****
2 REM *
3 REM * C64-HAM.OBJ *
4 REM *
5 REM *****
6 :
7 :
8 PRINT CHR$(147)"DATA BEOLVASAS ES ELLENORZES ...":J=49152:VE=49356:P=J
9 FOR B=0 TO 7:READ A#
10 L=ASC(MID$(A#,2,1))
11 H=ASC(MID$(A#,1,1))
12 L=L-48:IF L>9 THEN L=L-7
13 H=H-48:IF H>9 THEN H=H-7
14 PRINT"#####":P=P+1
15 IF H>15 OR L>15 THEN 17
16 A=H*16+L:POKE J+B,A:T=T+A:NEXT B:READ A:IF A=T THEN 18
17 PRINT:PRINT"DATA HIBA ... SOR:"PEEK(64)*256+PEEK(63):END
18 T=0:J=J+8:IF J<VE THEN 9
19 PRINT"###KESZ":END
20 DATA 78,A2,00,8A,18,69,01,29,0591
21 DATA 07,09,38,9D,00,C5,8A,69,0669
22 DATA 01,0A,0A,0A,0A,09,08,29,0099
23 DATA 7F,9D,00,C4,E8,E0,C8,D0,1344
24 DATA E2,A9,70,8D,C7,C5,A9,08,1221
25 DATA 8D,C7,C4,A9,86,8D,14,03,1003
26 DATA A9,C0,8D,15,03,A9,1B,8D,0863
27 DATA 11,D0,A9,18,8D,16,D0,A9,0958
28 DATA 00,8D,15,D0,8D,21,D0,8D,0893
29 DATA 20,D0,8D,0E,DC,A9,4C,8D,1001
30 DATA 05,DC,A9,38,8D,04,DC,AD,1068
31 DATA 11,D0,10,FB,AD,11,D0,30,0938
32 DATA FB,A9,30,CD,12,00,00,FB,1358
33 DATA A2,11,A9,18,8D,11,D0,3E,0880
34 DATA 0E,DC,A9,38,8D,11,D0,AD,1001
35 DATA 0D,DC,AD,00,DD,29,FC,09,0929
36 DATA 02,8D,00,DD,58,60,A9,9F,0876
37 DATA 38,ED,04,DC,C9,0C,90,03,0877
38 DATA 4C,BF,C0,4A,90,00,8D,9A,0972
39 DATA C0,10,FE,EA,EA,EA,EA,EA,1632
40 DATA EA,A9,30,8D,11,D0,EA,EA,1285
41 DATA EA,A2,00,8D,00,C4,8D,18,0946
42 DATA D0,8D,00,C5,8D,11,D0,E8,1192
43 DATA E0,C8,D0,EF,4C,31,EA,A9,1399
44 DATA 4C,8D,05,DC,A9,C7,8D,04,0955
45 DATA DC,4C,BC,C0,00,00,00,00,0676

```

READY.

érjük el. Azt tudnunk kell azonban, hogy a VIC ebben az esetben is csak arra a szín RAM-ra fog hivatkozni, amely az első sorban érvényes volt. Ha mást akarunk, minden sorban még külön közölni kell vele, hogy azt tegye, amit mi akarunk. És ez is lehetséges az n AND 7 utasítással a \$D011 regiszter alsó három bitjénél. Ezt a VIC cellát szigorúan egy meghatározott időben kell fölülrírni, ellenkező esetben a képet nem lehet korrektül ábrázolni. Sajnos ez az időpont ahhoz későn van, hogy egy sor mind a 40 karakterénél kihasználhassuk ezt: A bal szélén a VIC a \$FF színértéket kapja. Ahhoz hogy a FLI-t a teljes képernyőn használni lehessen, a felső és az alsó keretet ki kell kapcsolni. Csak így használható mind a 200 lehetséges FLI sor.

Egy FLI kép ezek szerint egy szín RAM-mal és a bitmaphez tartozó plusz nyolc video RAM-mal kell rendelkezzen. A mi rutinunkban a \$4000–\$8000 területet használjuk erre a célra az alábbiak szerint:

\$4000–\$43FF a 0, 8, 16...192 rasztorsorok video RAM-ja
 \$4400–\$47FF az 1, 9, 17...193 rasztorsorok video RAM-ja
 \$4800–\$4BFF a 2, 10, 18...194 rasztorsorok video RAM-ja
 \$4C00–\$4FFF a 3, 11, 19...195 rasztorsorok video RAM-ja
 \$5000–\$53FF a 4, 12, 20...196 rasztorsorok video RAM-ja
 \$5400–\$57FF az 5, 13, 21...197 rasztorsorok video RAM-ja
 \$5800–\$5BFF a 6, 14, 22...198 rasztorsorok video RAM-ja
 \$5C00–\$5FFF a 7, 15, 23...199 rasztorsorok video RAM-ja
 \$6000–\$7FFF multicolor vagy HiRes bitmap
 \$D800–\$DCFF szín RAM



Ha egy 8×8-as blokkot szeretnénk fölülírni a megfelelő színadatokkal, nyolc értéket kell a hozzá tartozó video RAM-ba írni (minden sorhoz egyet). A szín RAM-ot és a tulajdonképpeni grafikát (bitmap) úgy kell használni, mint kikapcsolt FLI-nél. Az adott tárolócellák pozícióit az alábbi táblázat alapján számolhatjuk ki:

line	szín RAM	video RAM	bitmap	színkód
0	\$d800+n:06	\$4000+n	\$6000+n*8	11 10 10 11 3 2 2 3
1	\$d800+n:06	\$4400+n	\$6001+n*8	10 01 01 10 2 1 1 2
2	\$d800+n:06	\$4800+n	\$6002+n*8	10 01 01 10 2 1 1 2
3	\$d800+n:06	\$4c00+n	\$6003+n*8	10 01 01 10 2 1 1 2
4	\$d800+n:06	\$5000+n	\$6004+n*8	10 01 01 10 2 1 1 2
5	\$d800+n:06	\$5400+n	\$6005+n*8	10 01 01 10 2 1 1 2
6	\$d800+n:06	\$5800+n	\$6006+n*8	10 01 01 10 2 1 1 2
7	\$d800+n:06	\$5c00+n	\$6007+n*8	11 01 01 11 3 2 2 3

Arra figyeljünk, hogy az első három karaktert nem lehet használni. Az „n” változó értéke tehát háromnál mindig nagyobb legyen.

A C64—HAM.OBJ programot a SYS 49153-vel indíthatjuk.

Ha valaki használná az FLI rutint saját programjában, gondoskodjon róla hogy az I flag nulla (kikapcsolt) legyen. Ellenkező esetben az FLI nem működik. Bemutatásként mellékeljük a HAMDEMI programot, amelyet a SYS 36864 utasítással indíthatunk. Ez kizárólag a C64—HAM.OBJ programmal együtt használható. A demó igyekszik képet adni a kis helyen is létrehozható színekavalkádról.

Mi az az FLI?

Az FLI a „Flexible Line Interpretation” rövidítése. Emögött az alábbi hatás rejtezik: Ebben a módban a VIC a képernyő minden egyes sorát egyenként értékeli ki, így egy 8×8-as mezőben mind a 16 szín megjeleníthető. Hogy ezt megvalósítsuk, az egyes sorok között át kell kapcsolni a video RAM-ot. Sajnos a dolognak van egy buktatója. Ha ugyanis elvégezzük ezt az átkapcsolást egy sorban (egy meghatározott helyen), a sor első három karakterét nem tudjuk megjeleníteni. Tehát csak 296 pixel áll a rendelkezésre. Azonban általában ez a három karakter nem zavar, mivel ott a háttér és az előtér színe azonos.

```

0 10 REM *****
1 20 REM *
2 30 REM * C64—HAMDEMO1 *
3 40 REM *
4 50 REM *****
5 60 REM
6 70 FOR I=35840 TO 37577
7 80 READ A:POKE I,A:S=S+A
8 90 NEXT I
9 100 IF S<>164410 THEN PRINT "HIBA A DATA-SOROKBAN..." :END
10 110 PRINT "KESZ !":END
11 120 DATA234,0,43,23,52,70,96,96,100,67,113,234,0,32,64,234,0,2,70,70,234,0
12 130 DATA2,64,234,0,32,96,234,0,1,70,62,62,70,234,0,1,96,234,0,32,70,228,55
13 140 DATA23,23,115,228,70,96,234,0,27,96,234,0,3,96,234,0,1,70,67,52,70,234
14 150 DATA0,1,96,234,0,32,224,234,0,2,70,70,234,0,2,224,234,3,48,16,234,0,2,96
15 160 DATA96,234,0,2,16,234,0,32,64,234,0,2,70,70,234,0,2,64,234,0,32,96,234
16 170 DATA0,1,70,62,62,70,234,0,1,96,234,0,32,96,228,62,23,23,62,228,96,234,0
17 180 DATA32,96,234,0,1,96,52,52,96,234,0,1,96,234,0,32,224,234,0,2,70,70,234
18 190 DATA0,2,224,234,3,48,112,96,96,96,96,234,0,2,112,234,0,32,96,234,0,1,224
19 200 DATA70,70,234,0,2,96,234,0,32,96,96,70,227,62,70,234,0,1,96,234,0,32,96
20 210 DATA70,67,49,19,62,70,96,234,0,32,96,234,0,1,96,52,52,96,234,0,1,96,234
21 220 DATA0,32,48,234,0,2,70,70,234,0,2,48,234,3,48,112,64,234,0,1,96,96,234
22 230 DATA0,2,112,234,0,32,96,234,0,1,224,70,70,234,0,2,96,234,0,32,96,100,70
23 240 DATA227,48,70,96,96,234,0,32,96,96,64,19,19,230,96,96,234,0,32,96,234,0
24 250 DATA1,96,70,70,96,234,0,1,96,234,0,32,48,234,0,2,96,96,234,0,2,48,234,3
25 260 DATA48,48,224,234,0,1,96,96,234,0,2,48,234,0,32,96,234,0,1,96,70,70,96
26 270 DATA234,0,1,96,234,0,32,96,100,64,49,19,64,96,96,234,0,32,96,96,70,48,51
27 280 DATA30,96,96,234,0,32,96,234,0,2,70,70,234,0,2,96,234,0,32,112,234,0,2
28 290 DATA96,96,234,0,2,112,234,3,48,48,48,234,0,1,70,70,234,0,2,48,234,0,32
29 300 DATA96,234,0,1,96,52,52,96,234,0,1,96,234,0,32,96,70,67,19,19,52,70,96
30 310 DATA234,0,32,96,234,0,1,70,62,62,230,234,0,1,96,234,0,32,96,234,0,2,70
31 320 DATA70,234,0,2,96,234,0,32,112,234,0,2,96,96,234,0,2,112,234,3,48,224,112
32 330 DATA234,0,1,70,70,234,0,2,224,234,0,32,96,224,96,52,52,96,234,0,1,96,234
33 340 DATA0,32,96,228,62,23,23,62,228,96,234,0,32,96,234,0,1,70,62,62,96,234
34 350 DATA0,1,96,234,0,32,64,234,0,2,70,70,234,0,2,64,234,0,32,16,234,0,2,96
35 360 DATA96,234,0,2,16,234,3,48,224,19,70,70,70,234,0,2,224,234,0,32,96,224
36 370 DATA70,67,52,70,234,0,1,96,234,0,32,70,228,55,23,23,115,228,70,96,234,0
37 380 DATA31,96,234,0,1,70,62,62,96,234,0,1,96,234,0,32,64,234,0,2,70,70,234
38 390 DATA0,2,64,234,0,32,23,52,100,96,96,100,67,113,234,4,93,90,64,64,64,64
39 400 DATA64,64,64,126,234,0,7,106,234,0,7,85,1,1,1,1,9,9,41,85,64,64,64,64,96
40 410 DATA96,104,86,234,0,7,126,234,0,7,90,1,1,1,1,1,1,1,234,1,0,64,64,64,64
41 420 DATA64,64,64,64,234,0,12,1,1,5,9,37,39,167,151,159,173,189,122,88,216,218
42 430 DATA214,246,122,126,94,234,0,4,64,64,80,96,234,0,8,1,1,1,1,1,1,1,1,234
43 440 DATA1,0,64,64,64,64,65,85,165,234,0,3,1,21,165,169,149,41,37,167,151
44 450 DATA95,126,245,90,167,167,175,94,189,245,213,218,218,214,245,122,126

```



460 DATA95,87,104,88,218,214,245,126,95,90,234,0,3,64,84,90,106,86,1,1,1,1
 470 DATA1,65,85,90,234,1,0,165,85,65,64,64,64,64,64,149,169,165,21,1,234,0
 480 DATA3,90,245,126,95,151,167,37,41,213,245,126,173,87,151,167,167,87,95
 490 DATA126,122,213,214,218,218,90,90,107,95,126,122,244,212,86,106,90,84,64
 500 DATA234,0,3,90,85,65,1,1,1,1,1,234,1,0,64,64,64,64,64,64,64,234,0,8
 510 DATA9,5,1,1,234,0,4,122,189,173,159,151,167,39,37,94,126,122,246,214,218
 520 DATA216,88,96,80,64,64,234,0,12,1,1,1,1,1,1,1,1,234,1,0,64,64,64,64,64
 530 DATA64,64,90,234,0,7,126,234,0,7,149,41,9,9,5,1,1,1,85,104,96,96,80,64
 540 DATA64,64,85,234,0,7,86,234,0,7,126,1,1,1,1,1,1,1,90,234,24,40,234,0,44
 550 DATA14,14,234,0,3,14,6,234,0,34,3,14,14,234,0,37,14,7,7,14,234,0,36,14
 560 DATA7,7,4,234,0,37,14,14,234,0,35,6,14,234,0,4,14,6,234,3,5,88,32,0,192
 570 DATA162,0,134,99,134,101,169,64,133,100,169,140,133,102,160,0,177,101,201
 580 DATA234,208,53,200,177,101,133,98,200,177,101,133,97,169,0,129,99,230,99
 590 DATA208,2,230,100,198,97,165,97,201,255,208,2,198,98,165,98,208,232,165
 600 DATA97,208,228,24,165,101,105,2,133,101,144,3,230,102,24,76,86,144,145
 610 DATA99,230,99,208,2,230,100,230,101,208,2,230,102,165,101,201,255,208,175
 620 DATA165,102,201,143,208,169,162,0,189,3,128,157,3,216,189,3,129,157,3,217
 630 DATA189,3,130,157,3,218,189,3,131,157,3,219,232,208,229,169,0,141,135,146
 640 DATA141,136,146,141,137,146,169,3,133,111,165,111,10,170,24,189,119,146
 650 DATA109,135,146,133,99,189,120,146,105,0,133,100,189,127,146,109,135,146
 660 DATA133,101,189,128,146,105,0,133,102,189,111,146,109,136,146,133,97,189
 670 DATA112,146,109,137,146,133,98,32,146,145,198,111,16,198,24,173,136,146
 680 DATA105,64,141,136,146,144,4,238,137,146,24,24,173,135,146,105,8,141,135
 690 DATA146,201,32,144,166,169,255,141,15,212,169,128,141,18,212,169,25,133
 700 DATA110,169,3,133,97,169,64,133,98,169,3,133,99,169,216,133,100,160,0,162
 710 DATA8,177,97,32,85,145,145,97,24,165,98,105,4,133,98,202,208,239,56,165
 720 DATA97,233,0,133,97,165,98,233,32,133,98,177,99,32,85,145,145,99,200,192
 730 DATA37,208,212,24,165,97,105,40,133,97,144,3,230,98,24,24,165,99,105,40
 740 DATA133,99,144,3,230,100,24,198,110,208,182,76,244,144,134,112,133,111
 750 DATA74,74,74,74,32,114,145,10,10,10,10,133,101,165,111,41,15,32,114,145
 760 DATA5,101,166,112,96,170,173,27,212,41,3,240,12,201,1,240,12,201,2,240
 770 DATA12,189,186,146,96,189,138,146,96,189,154,146,96,189,170,146,96,169
 780 DATA6,133,110,169,88,133,103,169,97,133,104,169,43,133,105,169,64,133,106
 790 DATA169,43,133,107,169,216,133,108,162,8,160,7,177,103,145,97,136,16,249
 800 DATA160,0,169,8,133,109,177,105,145,99,24,165,106,105,4,133,106,165,100
 810 DATA105,4,133,100,198,109,208,235,177,107,145,101,56,165,105,233,255,133
 820 DATA105,165,106,233,31,133,106,56,165,99,233,255,133,99,165,100,233,31
 830 DATA133,100,230,101,208,2,230,102,230,107,208,2,230,108,24,165,97,105,8
 840 DATA133,97,144,3,230,98,24,24,165,103,105,8,133,103,144,3,230,104,24,202
 850 DATA208,151,24,165,103,105,0,133,103,165,104,105,1,133,104,24,165,97,105
 860 DATA0,133,97,165,98,105,1,133,98,24,165,101,105,32,133,101,165,102,105
 870 DATA0,133,102,24,165,107,105,32,133,107,165,108,105,0,133,108,24,165,105
 880 DATA105,32,133,105,165,106,105,0,133,106,24,165,99,105,32,133,99,165,100
 890 DATA105,0,133,100,198,110,240,3,76,174,145,96,88,97,216,104,88,112,216
 900 DATA119,43,64,27,65,11,66,251,66,43,216,27,217,11,218,251,218,0,0,0,0,1
 910 DATA4,15,2,8,9,7,5,11,14,9,2,7,10,15,0,1,12,3,4,10,11,13,10,6,8,9,4,13
 920 DATA5,3,0,1,4,15,12,5,6,7,14,11,5,6,2,7,8,3,0,1,2,3,12,14,9,13,8,9,14,11
 930 DATA12,13,8,15
 READY.

C 128

MERGE PEEK ÉS POKE NÉLKÜL

Ha két BASIC programot össze szeretnénk fésülni, MERGE parancs híján valami nehézségre mindig számíthatunk. Vagy be kell tölteni egy MERGE helyettesítő rutint, vagy egy sor POKE parancsot, tárolócímet kell megjegyeznünk. Rövid programok esetében (kb. 20 sor 80 karakterrel) azonban egy egyszerű trükk könnyíthet a helyzetünkön.

Listázzuk ki a rövid programot a 80 karakteres képernyőre, majd váltunk (ESC) (X)-szel a negyvenesre. Hívjuk be most itt LOAD-dal a másik programot. Ekkor ugyan törlődik az első, de a 80 karakteres képernyőn megmaradt! Váltunk tehát oda ismét (ESC) (X)-szel, és a (RETURN)-nel vegyük át a sorokat.

Ha csak a 80 karakteres képernyőt akarjuk használni, akkor az (ESC) (T) és az (ESC) (B) segítségével rögzítsünk egy, a kiírást nem érintő ablakot. Ezen keresztül bonyolítjuk azután a másik program betöltését. A művelet után a kijelölt ablakot a (HOME) kétszeri megnyomásával törölhetjük, és az ott álló sorokat átvehetjük. A trükk azonban csak akkor használható, ha rövid részek illesztését végezzük. Hosszabb programoknál a MERGE rutin vagy a PEEK/POKE-ok használatát nem tudjuk elkerülni.

C-64 bővítések

Univerzális EPROM-égető C-64-hez

EPROM-mal előbb-utóbb szinte minden amatőr és számítógép-tulajdonos találkozhat, még az is, aki a gép hardverét nem szereti, csak programozza a masinát.

Nehéz ma már ebben a témában újat alkotni, különösen ha C-64-hez illesztjük a hardvert. Nem is ez a célom, hiszen számos külföldi és hazai szaklap ismertette már EPROM-égetőt. Az általam ismert cikkek mindegyike magán hordoz egy-két apró hiányosságot, ami a kevésbé gyakorlott felhasználót bizonytalanná teszi az építés, alkalmazás tekintetében. Két lényegesebb ilyen probléma a következő:

A beégetők egy része, különösen amelyek a hazai irodalomban jelentek meg, kevés kivétellel csak egy típus égetésére alkalmasak. Ez problémát jelent az alkatrész beszerezhetősége, az ára és a szükséges memória-kapacitás optimális egyeztetése miatt. Egy példa: a 2 K-s 2716 ma drágább, mint a 8 K-s 2764, ezért azt a beégetőt, ami csak egy típust tud égetni, nem szabad megépíteni, mert azt hamar „kinőjük”.

A másik probléma az átlagos képességű amatőr előtt az, hogy a szoftver és a hardver teljesen elválik egymástól. Ez alatt azt értem, hogy a szerzők a jól megtervezett hardver mellé írják egy szoftvert, ami a kapcsolási rajz mellett jelenik meg a lap hasábjain. Ennek begépelése, kimentése szinte biztos hibaforrás. Kérdéssé válik ezután, hogy a drágán vásárolt EPROM-ot feláldozza-e valaki a próbaégetésre.

Ezeket a gondolatokat az ország különböző részeiből hozzám érkezett olvasói levelek sugallták, és kész tettem arra, hogy közlök ezt az EPROM-égetőt, amely kiküszöböli a fenti hiányosságokat. 2716-tól 27256-ig mindenféle EPROM-ot éget, a működtető szoftvert egy cartridge tartalmazza, így a gép ennek csatlakoztatása után rögtön EPROM-égetőként jelenkezik be.

Ezen kívül semmiféle szoftvert nem kell hozzá alkalmazni. A kezelése csak

némi szakértelmet igényel, a képernyőn megjelenő menü pedig mindent megmagyaráz.

Az EPROM-égető működése

Az áramkör kapcsolási rajza az 1. ábrán látható. Az égetőpanel a C-64 felhasználói portjára csatlakozik. Innen kapja az információt, amivel kiválasztja az égetendő EPROM paramétereit, és innen jönnek ugyanezen a 8 kivezetésen a beírandó adatok is. Az EPROM címezését IC₁, IC₂ számláló végzi.

IC₁ az órajelet a port 5. pontjáról kapja. A számláló a beírt adat visszellenőrzése után lép egyet. Az IC₃ az égetendő EPROM paramétereit állítja be a Q_n kimenetére kapcsolódó inverterek, diódák, tranzistorok segítségével. A mellékelt táblázatban ismertetett lábkiosztás alapján követhető ennek működése. Mivel sokféle típus égetünk ugyanabban a foglalatban, a táblázat szerint szükség van arra, hogy egyazon lábra többféle információt tudjunk kapcsolni. Ezeket csillaggal jeleztem. N₃, T₂, D₁ a 2716 esetében a V_{pp} = 25 V-os feszültséget kapcsolja, a többi típusnál ugyanerre a lábra A₁₁ cím kerül. N₁, n₂, T₁ a 2732-nél és a 2716-nál a tápfeszültséget, a többi típusnál az A₁₃ címet kapcsolja. T₁ az N₂ nyitott kollektoros tranzistorával, ami a token belül van, komplementer fokozatot alkot, így biztosítja a 24 lábú tokoknak tápellátást. N₄, T₃, D₃, 2732-nél V_{pp}-t a Többi típusnál OE jelet kapcsol. T₄, T₅ és T₆ a V_{pp} megfelelő értékét állítja be a Zener-diódák rövidrezárásával. A V_{pp} értékét a táblázat mutatja.

Ennél az értéknél a ±5% tűrés a megengedett, ezért a zenerek megválasztásánál ezt figyelembe kell venni! A V_{pp}-t az IC₄ állítja elő a bemenetére kötött egyenfeszültségből. Ez a feszültség 30–35 V lehet. Ez jöhet egy önálló külső tápegységből, amit nem kell külön stabilizálni.

Előállítható a C-64 felhasználói portjának 10. és 11. pontján megjelenő 9 V-os váltófeszültségből is, feszültségszorzó segítségével. Ezt a megoldást mutatja a 2. ábra.

Az áramkör szerelése, élesztése

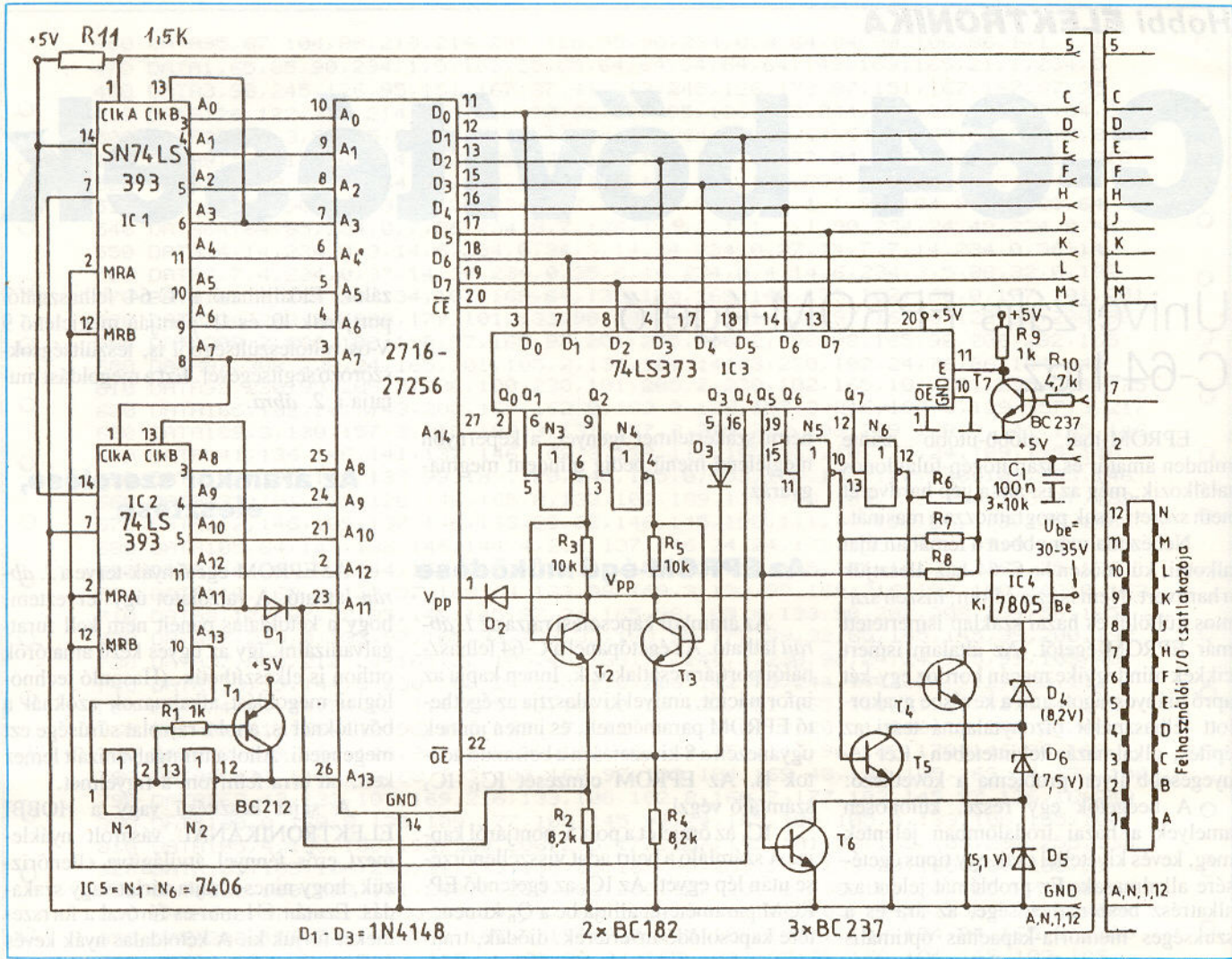
Az EPROM-égető nyák-terve a 3. ábrán látható. A rajzolatot úgy terveztem, hogy a kétoldalas panelt nem kell furatgalvanizálni, így az ügyes kezű amatőrök otthon is elkészíthetik. (Hasonló technológiai megoldást alkalmazok azoknál a bővítőknél is, ahol a rajzolat sűrűsége ezt megengedi. Ahol a furatgalvanizált lemez kell, ott arra felhívom a figyelmet.)

A saját készítésű vagy a HOBBI ELEKTRONIKÁNÁL vásárolt nyáklemzet erős fényrel átvilágítva ellenőrizzük, hogy nincs-e rajta zárlat vagy szakadás. Ezután Ø1 mm-es fúróval a forrasztókat fúrjuk ki. A kétoldalas nyák kevés átkötéssel rendelkezik. Ezek az átkötések jórészt IC lábhoz esnek, így kétoldalról forrasztható IC foglalatot használva, könnyen elkészíthetők. Ilyen típusú a Preci clip foglalat, amelyik nem fekszik rá műanyag részével a panelre.

Megjegyzem, hogy az EPROM-helyének kivételével nem kell foglalatot használni, mert a TTL tokok nyugodtan beforraszthatók kétoldalról is. Kétoldalról kell beforrasztani néhány ellenállást, diódát, tranzisztort, kondenzátort. Csúpan két olyan átkötés van, ahol egy kis huzaldarabkával kell kontaktust létrehozni.

Összegezve: a szerelésnél az a lényeg, hogy az ültetési oldalon minden forrasztást be kell forrasztani. Egyszerűsíti a helyzetet, hogy ezen az oldalon csak annyi forrasztás van, amennyi az átkötéshez kell. A forrasztás nagy figyelemmel végezzük, és gondosan ellenőrizzük, mert egy forrasztás hiánya miatt már nem működik az áramkör. Az átkötések helyét az ültetési rajzon (4. ábra) fekete pont jelöli.

A beméréshez, hidegindításhoz nem feltétlenül kell a számítógép. Ezt helyettesíti az 5. ábrán látható célműszer és egy



1. ábra

stabilizált tápegység, amelyről +5 V-ot adunk a táppontokra (2., illetve A, N, 1. és 12. pontok). Az órajel kimenetét az 5. ponton át az IC₁ CLK A bemenetre visszük, a tapogatócsúccsal pedig végignézzük az A0...A13 kimeneteket. Az ellenőrző LED villogása jelzi a frekvenciaosztás mértékét. Ha szabad szemmel nem követhető, az 1 k×-os trimmer-potenciométerrel álljunk utána.

IC₁ és IC₂ csak akkor működik, ha az IC₃ Q₄ kimenete logikai „L” szinttel ezt engedélyezi. Ezért teszteljük az IC₃ működését. A 7-es csatlakozó pontot egy huzaldarabkával kössük a GND-re. A D0...D7 bemenetet egy mérőszinórral egymás után érintsük „L”, majd „H” szintre, a tapogatócsúcs a Q0...Q7 kimeneteken ugyanazt jelzi, amit a bemenetre adtunk. Jegyezzük fel egy papírra, hogy mit adtunk a bemenetre. A 7. pontot kössük át +5 V-ra. Most már bármilyen szintet is adunk a bemenetre, a kimenet azt őrzi, amit a papírra jegyeztünk. Ez alap-

Táblázat

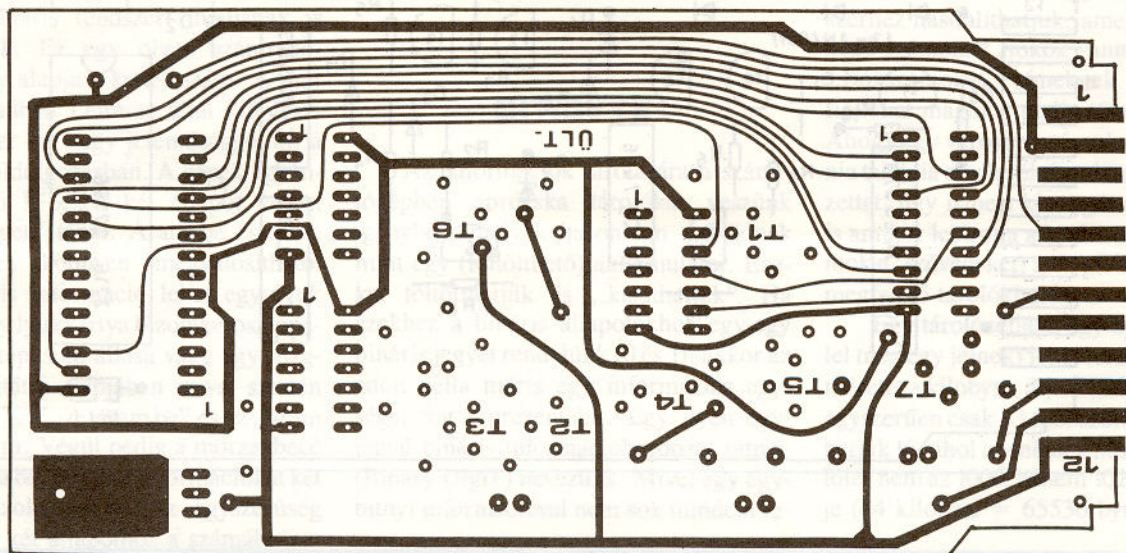
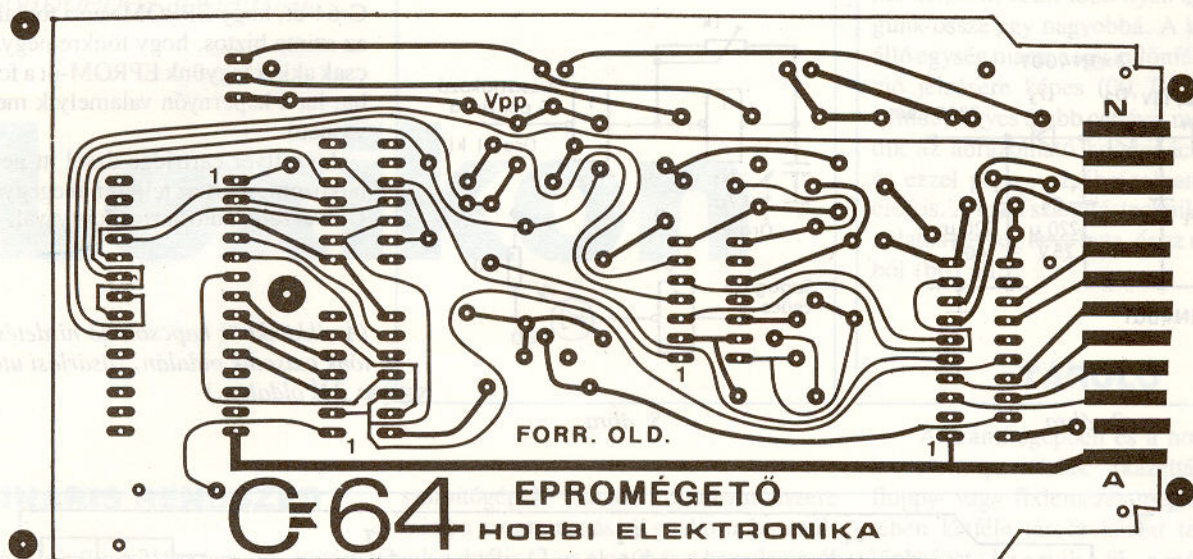
27256	27128	2764	2732	2716	I		2716	2732	2764	27128	27256	
V _{pp}	V _{pp}	V _{pp}			1	28		+5V	+5V	+5V		
A ₁₂	A ₁₂	A ₁₂			2	27		PGM	PGM	A ₁₄	*	
A ₇	A ₇	A ₇	A ₇	A ₇	3	24	26	+5V	+5V	NC	A ₁₃	*
A ₆	A ₆	A ₆	A ₆	A ₆	4	23	25	A ₈	A ₈	A ₈	A ₈	
A ₅	A ₅	A ₅	A ₅	A ₅	5	22	24	A ₉	A ₉	A ₉	A ₉	
A ₄	A ₄	A ₄	A ₄	A ₄	6	21	23	V _{pp}	A ₁₁	A ₁₁	A ₁₁	*
A ₃	A ₃	A ₃	A ₃	A ₃	7	20	22	OE	OE/VS	OE	OE	*
A ₂	A ₂	A ₂	A ₂	A ₂	8	19	21	A ₁₀	A ₁₀	A ₁₀	A ₁₀	
A ₁	A ₁	A ₁	A ₁	A ₁	9	18	20	CE	CE	CE	CE	
A ₀	A ₀	A ₀	A ₀	A ₀	10	17	19	Q ₇	Q ₇	Q ₇	Q ₇	
Q ₀	Q ₀	Q ₀	Q ₀	Q ₀	11	16	18	Q ₆	Q ₆	Q ₆	Q ₆	
Q ₁	Q ₁	Q ₁	Q ₁	Q ₁	12	15	17	Q ₅	Q ₅	Q ₅	Q ₅	
Q ₂	Q ₂	Q ₂	Q ₂	Q ₂	13	14	16	Q ₄	Q ₄	Q ₄	Q ₄	
⊥	⊥	⊥	⊥	⊥	14	13	15	Q ₃	Q ₃	Q ₃	Q ₃	

EPROM típus

- 2716
- 2732
- 2732A
- 2764
- 2764A
- 27128
- 27128A
- 27256

Égető feszültség

- 25V
- 25V
- 21V
- 21V
- 125V
- 21V
- 125V
- 125V (V_{pp} = 6V)



3. ábra

ján beállíthatjuk Q_5 , Q_6 és Q_7 -tel a V_{pp} -t is, a táblázat szerint, amit egy voltmérővel ellenőrizhetünk.

Hiba esetén értelemszerűen vizsgáljuk meg a NYÁK-ot, a forrasztásokat, az IC-eket. Ezt az óvatos vizsgálatot elsősorban a kevés tapasztalattal rendelkezőknek ajánlom, mert a cartridge-ben levő szoftver tartalmaz egy ellenőrző rutint, melynek segítségével ezt automatikusan elvégezhetjük. Fontos, hogy minden mérést az EPROM foglalat kontaktusain végezzünk, hogy az ide érkező vezetékek hibáit is ki lehessen mutatni.

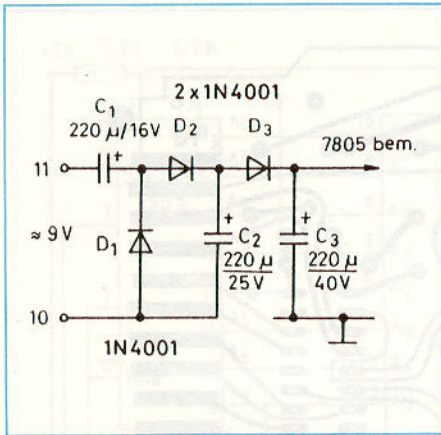
Az élesztésnél legfontosabb a V_{pp} ellenőrzése. A 2. ábrán látható feszültség-szorzó rákerült a panelre, így a C_3 -on a gép bekapcsolása után kb. 32 V mérhető. *Figyelem, nagyon fontos!* Bármilyen műszerrel mérünk, azt a C-64 bekaposo-

lása előtt stabilan földeljük össze az égetővel. A „meleg” ponttal ezután bekapcsolt állapotban nyugodtan végigtapogathatjuk a szükséges pontokat. Ez a túlzott óvatosságnak tűnő intelm minden C-64 bővítő mérésére vonatkozik. Mérés közben a laza, leszakadó földvezeték szinte biztos, hogy hibát okoz a C-64-en belül. Különösen a USER portra menő 6526 szeret „elrepülni”.

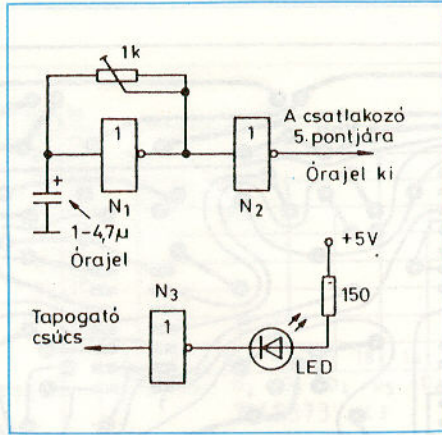
Lépjünk kéziműszerrel a V_{pp} helyére (EPROM foglalat 1. láb). A menüből válasszuk ki a V_{pp} tesztet. A kurzormozgató nyílal jelöljük ki az egyik értéket. Return után ezt mutatja a műszer. Space-szel száll ki. Ezt mindhárom értékre meg kell csinálni. Ezután válasszuk ki a menü 3. pontját, a címtesztet. Logikai ceruzával vagy szkóppal a táblázat szerint ellenőrizzük a címeket. A13...A0-ig. Itt mindig dupla

frekvenciájú jelet mutat a műszer az előző ponthoz képest. Következő mérés a 4. pont, az adatterszt. Először 00-át írjunk be, returnnel ez jelenik meg a D0...D7 adatbuszon. Space után ismét az adatterszt hívjuk és FF-et írjunk be. Ha ez is rendben megjelenik a foglalat lábain, hibátlan a gép és az égető is. A teszt többi pontja chip-engedélyazást és V_{pp} értéket ellenőriz. Értelem szerint ezt is ellenőrizzük. Csak ezek után tegyünk EPROM-ot a készülékbe.

Még egy fontos megjegyzés: Az 1. lábon a V_{pp} értéke a ki- és bekapcsolási tranziensek alatt 5–25 V között változhat aszerint, ahogy az elektronika az IC₄-et véletlenszerűen beállítja. Ez azt jelenti,



2. ábra

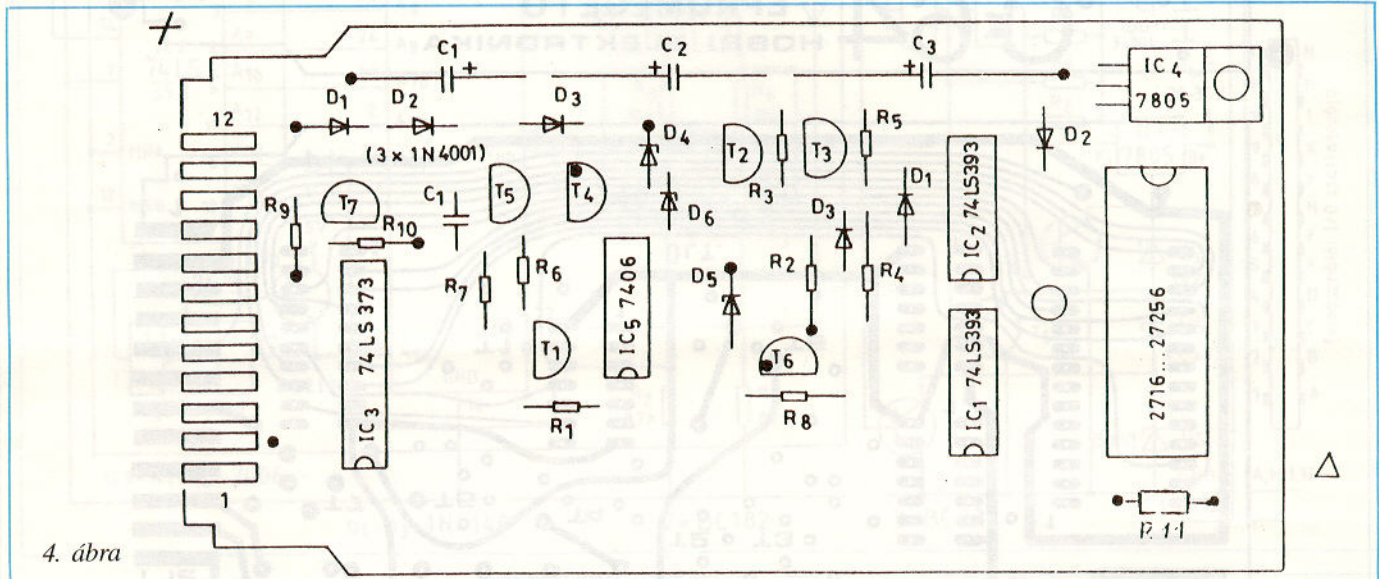


5. ábra

hogy ha úgy kapcsoljuk be vagy ki a C-64-et, hogy EPROM van a foglalatban, az szinte biztos, hogy tönkremegy. Ezért csak akkor tegyünk EPROM-ot a foglalatba, ha a képernyőn valamelyik menü olvasható.

A szoftver cartridge-dzsel itt nem foglalkozom, mert az teljesen megegyezik az Univerzális autostartos kártyával.

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



4. ábra

C 64 + FLOPPY + 50 db lemeztár, joystick 22 500 Ft-ért eladó. Valamennyi 1 éves, keveset használt. Tel.: 129-2604.

Hi! C-64-eladó! C-64 10 000 Ft, 410 db lemez 30 Ft/db, MK7, Mikrovilág hegyek stb. Érdeklődni: Kiss Tamás, 8000 Székesfehérvár, Benke F. u. 1.

4 MB-ig bővíthető 1,5 MB-ig kiépített Amiga 500-as RAM bővítő eladó vagy elcserélhető C 1541C menedzserkalkulátorra. Más műszaki cikk cseréje is érdekel. Major Nándor, 9300 Csorna, Kossuth u. 58/c. Mh.tel.: 64.

C-64-es programokat cserélek és eladok. Lemezen 20 Ft/oldal, kazettán 50 Ft/oldal. Felbélyegzett válaszborítékért listát küldök és kérek. Kázmér Attila, 6000 Kecskemét, Gizella tér 2.

Eladó Commodore VC-20 + C-64 tápegység + 1531 magnó + Junoszt 402-es tv + maxell és TDK lemezek. 1224 Budapest, VII. u. 4. Tel.: 227-22-60.

Üzemképtelen C-64-es és 1541-es floppyt vásárolnék. Eladó: motoros autóantenna vagy lemezekre cserélem. Tel.: 227-22-60.

Eladó 1 db C-64-es számítógép + magnó + 1 joystick + játékprogramok. Ár: 10 000 Ft. Cím: Kocsis András, 6098 Tass, Damjanich u. 4.

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.

Totókulcskészítő programok C-64-re (8 kbytesok). Válaszborítékért tájékoztató. Nagy Miklós, Újfehértó, Pf.: 41.

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

C-64 diszke 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.

Lexikon

XI. rész

BINÁRIS RENDSZER

Angolul Binary System. A bináris szó azt jelenti, hogy „két részből álló”. Ezért a bináris rendszert duálisnak is mondhatjuk. Ez egy olyan számrendszer, amely alapja a kettes szám. A már a 17. században Leibnitz által kifejlesztett rendszer ma nagy jelentőséggel bír a gépi adatfeldolgozásban. A gépek ugyanis könnyen képesek két állapot között „különbséget” tenni: Áram be és áram ki, mindez könnyen megvalósítható. Más bináris információ lehet egy lyuk megléte egy lyukkártya bizonyos oszlopában. Egy kapcsoló állása vagy egy mágnes polaritása alapján véve szintén visszavezethető az „áram be” és az „áram ki” állapotra. Végül pedig a morzeábécé bizonyította be, hogy az információkat két jellel ábrázolni lehet. Az egyszerűség kedvéért a két állapothoz a számábrázolásban a 0 és az 1-es jegyeket rendeltük hozzá.

SZÁMRENDSZER

Azokat a rendszereket, amelyek a halmazszerű egységek jegyekkel történő ábrázolását szolgálják, számrendszereknek hívjuk. A legismertebbek ezek közül a római, illetve a XI.—XII. században bevezetett arab számrendszerek. A nem túl gyakorlatias írásmód miatt — egy számot az M, D, C, L, X, V, I jegyek összege képezi — a római rendszert igen gyorsan fölváltotta az arab. Ez egy bázis számot használ (a decimális rendszerben a tízest), és az egyes helyiértékeket a bázisszám hatványaival értékeli. Egy ily módon leképezett szám értékét úgy kapjuk meg, hogy összeadjuk az egyes jegyek és a helyiértékek szorzatát. A bázisszám bármely egyenél nagyobb természetes szám lehet. A

számítógépek bináris munkamódszere miatt a programozásnál gyakorta használjuk a duális (2-es alapú) és a hexadecimális (16-os alapú) rendszert.

BIT/BYTE

Az információk tárolására a számítógépben aprócska tárolókat veszünk igénybe, amelyek hasonlóan működnek mint egy (föltölthető) akkumulátor. Ezeket föltölthetjük és „kisüthetjük”. Ha ezekhez a bináris állapotokhoz egy-egy bináris jegyet rendelünk (0 és 1), akkor az adott cella máris egy információs egységet fog reprezentálni. Egy ilyen egyjegyű bináris információhordozót bitnek (Binary DigIT) nevezünk. Mivel egy egybitnyi információval nem sok mindent le-

het kezdeni, ezért több ilyen egységet fogunk össze egy nagyobbá. A két cellából álló egység máris négy különféle információ jelzésére képes (00, 0,1, 10 és 11). Minden egyes újabb cellával megduplázódik az ábrázolható kombinációk száma, és ezzel persze az ábrázolható információé is. A ma a számítástechnikában használatos egység neve byte, és ez nyolc cellából (bit) áll.

TÁROLÓ

A számítógépben és a hozzá csatlakoztatott perifériák (kazettás egység, floppy- vagy fixlemezes meghajtók) esetében kétféle tároló között teszünk különbséget. Az egyik a fő-, a másik a munkatároló. A fő tárolót egy nagy polrendszerhez hasonlíthatjuk, amelyben különböző számozott fiókok vannak. Ezeket a „fiókszámokat” címeknek nevezzük, a fiókokat magukat pedig tárolócelláknak. Ahogy egy városi postásnak meg kell tudnia találnia egy levélcímzés alapján a címzettet, úgy ismeri ki magát a számítógép is amikor letárolja a kívánt adatokat (számokat, szövegeket) a megfelelő címetek a megfelelő tárolókba.

Egy tárolócellába egy byte-ot (ez felel meg egy jelnek) lehet lerakni. A tároló méretét a kilobyte, (ejtsd „kilobájt”, vagy egyszerűen csak K) többszöröseivel fejezhetjük ki, ahol azonban a használt kilo jelölés nem az 1000, hanem 1024 megfelelője (64 kilobyte = 65536 byte).

Duális rendszer, kettesalapú bázis

128	64	32	16	8	4	2	1	a decimális helyiérték
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	a kettő hatványai
1	0	1	0	0	1	1	0	a duális szám

Az eredmény:

$$0 \times 1 + 1 \times 2 + 1 \times 4 + 0 \times 8 + 0 \times 16 + 1 \times 32 + 0 \times 64 + 1 \times 128$$

$$0 \quad 2 \quad 4 \quad 0 \quad 0 \quad 32 \quad 0 \quad 128 = 166$$

Decimális rendszer, tízesalapú bázis

1000	100	10	1	a decimális helyiérték
10^3	10^2	10^1	10^0	a kettő hatványai
1	9	8	7	a decimális szám

Az eredmény:

$$7 \times 1 + 8 \times 10 + 9 \times 100 + 1 \times 1000$$

$$7 \quad 80 \quad 900 \quad 1000 = 1987$$

Képernyővarázslat — a TED

Negyedik rész: A grafika programozása

Nos, ismét találkozunk az újság hasábjain. A grafika a PLUS/4 programozásának egyik legérdekesebb — és leglátványosabb — része. Gépünk ismét eltér a C-64-től. Sokkal több színt (illetve árnyalatot) használhatunk (15 szín 8—8 árnyalatát, plusz a feketét), de a hardver nem támogatja a SPRITE-kezelést. Hosszabb-rövidebb csatoláson bizonyára már mindenki részt vett a grafika területén a BASIC V3.5 segítségével, de ott nem produkálhatott igazán érdekes dolgokat. A BASIC alapban a kétszínű karakteres és a grafikus (bittérképes) módot támogatja, de a multicolor (többszínű) és a bővített háttérszín módot nem, és nem ad lehetőséget a kép áthelyezésére. Ezeket természetesen gépikódban — illetve a TED regiszterek közvetlen hozzáféréssel POKE-kal BASIC-ból is — könnyen kezelhetjük, és a gép grafikájának új képességeit fedezhetjük fel. Vágujnk hát bele! A grafikával kapcsolatos regiszterek:

\$FF06 — TED vezérlő 1.

- b7. — I.C. teszt, értéke 0
- b6. — bővített háttérszín, 1-ben aktív
- b5. — bittérkép engedélyezve, 1-ben aktív
- b4. — képernyő engedélyezve, tőzse 1-ben aktív
- b3. — 24/25 soros képernyő, 0/1 állapotban
- b2-0 — függőleges finom scroll

\$FF07 — TED vezérlő 2.

- b7. — 256 karakteres mód, 1-ben aktív
- b6. — PAL (európai)/NTSC (amerikai) TV jel 0/1 állapotban
- b5. — TED STOP, 1-ben aktív
- b4. — Többszínű üzemmód, na vajon mi-ben aktív?
- b3. — 38/40 karakteres képernyő, 0/1 állapotban
- b2-0 — vízszintes finom scroll

Alapértékeik: \$FF06 — \$1B, \$FF07 — \$08. Tehát: a gép képernyője bekapcsolt, 25 soros képernyő, a képernyő-előállítás működik (TED STOP = 0), a kép 3 pixellel lefelé tolvá (függőleges finom scroll bitek), egy sorban 40 karakter látszik, a kép a bal szélső pozícióban van. Mi van még?

\$FF12 — többfunkciójú regiszter

- b7-6 — nem használt
- b5-3 — bittérkép helye 8 kilobájtban
- b2. — karaktergenerátor, bittérkép RAM/ROM, 0/1 állapotban

- b1-0 — 1. hanggenerátor felső két bitt
- \$FF13 — többfunkciós regiszter**
- b7-2 — karaktergenerátor kezdete 2K-ban
- b1. — SINGLE CLOCK, 1-ben aktív
- b0. — RAM/ROM lapozás érzékelő, 0/1 állapotban

\$FF14 — szín- és képernyő-, illetve szín- és fényerőmemória helye 2K-ban

- b7-3 — a fenti adat
- b2-0 — nem használt

Színregiszterek:

- \$FF15 — háttérszín
- \$FF16 — első segédszín
- \$FF17 — második segédszín
- \$FF18 — harmadik segédszín
- \$FF19 — keretszín

- b7. — nem használt
- b6-4 — fényerő
- b3-0 — szín

Alapállapotok: \$FF12:\$C4. A karaktergenerátor tehát ROM-ban van, a bittérkép helye \$0000—\$1F3F (nem pont 8192 bájt, hanem csak 8000), az első hanggenerátor magas bitjei nullán vannak. \$FF13:\$D1, azaz a karaktergenerátor \$D000-tól helyezkedik el (mint azt a \$FF12-ből tudjuk, a ROM-ban), a SINGLE CLOCK ki van kapcsolva, vagyis a gép a kereten gyorsabban fut, és a ROM van belapozva (vagyis a \$FF3E/\$FF3F regiszterek közül az előbbibe írtunk utoljára). \$FF14:\$0F, vagyis a kép- és színmemória \$0800-tól helyezkedik el. A kezdőcíme szabadon áthelyezhetők a regiszterek átírásával. Vigyázni az \$FF12-vel kell, mert ha zenélgetés közben vigyázatlanul írunk bele, a hang az alsó két bit átírásakor eltorzulhat, vagyis íraskor különösen figyelni kell, hogy csak a „grafikus” biteket írjuk át (ez persze fordítva is igaz). Óvatosan kell bánnunk az \$FF13 címmel, mert ha írskor a 0. bitet véletlenül 1-re váltjuk, ezt fogjuk mondani: „Mitől lett ilyen tetű laszú a programom?” — és vagy rájövünk („Én marha!”), vagy nem („Vacak ez a gép, átmegyek AMIGÁra!”). Ha a képet (vagy a karaktergenerátort) a ROM alá (\$8000 fölé) tesszük, vigyázni kell arra, hogy a ROM ki legyen lapozva (karakteres kép esetén), vagy jelezni kell az \$FF13 2. bitjével, hogy RAM-ból dolgozunk (ezt a karakterkészlet esetén a helytől függet-

lenül mindig meg kell tenni, a grafikus képnél csak akkor, ha a ROM alatt helyezkedik el).

A karakteres üzemmódot

A képinformáció két fő forrása a képernyőmemória és a karaktergenerátor. A kép- (és szín-) memória az a terület, ahonnan a TED a megjelenítendő karaktereket és ezek színét veszi. Mérete $2*25*40=2*1000$ bájt. A könnyebb kezelhetőség okán ezeket $2*1024$ bájton tároljuk. A tárban egymás után következnek a karakterek balról jobbra, és felülről lefelé haladva. Tehát a kívánt karakter címe: a képernyőmemória alapcíme+40*sor+ +oszlop. Természetesen a képlet a színmemóriára is érvényes, ha a képernyő-helyett a színmemória alapcímét írjuk be. A színmemória 7. bitje a villogást jelzi, a 6—4. bitek a fényerőt, míg a 3—0. bitek a színt határozzák meg.

Lépjünk be MONITORba, és adjuk ki:

```
T D000 D7FF 2000
> FF12 C0 20
```

Ezzel átmásoltuk a karaktergenerátort RAM-ba, és át is kapcsoltunk rá. Lépjünk ki BASIC-be X paranccsal. A kép ritka csúf látványt nyújtva kavargó, „szemetes”. Ezt az okozza, hogy a BASIC-ba lépéskor a rendszerprogram azonnal ROM-ra kapcsolja át a karakterkészletet, de már nem állítja vissza \$D000-ra. \$2000-tól pedig nem sok ROM található, vagyis a TED „lebegő” adatokat olvas be. Ugyanez történik, ha egy futó BASIC programba hiba csúszik. Ha erre a program ráfut, sok felhasználó fogja nevünket kellemetlen jelzőkkel illetni. Nos, lépjünk vissza MONITOR-ba, és írjuk át (vakon) az \$FF12-es regisztert \$C0-ra! Írjuk be: >FF12 C0 F 0C00 0C27 0. A képernyő felső sora megtelik kukaccal (gy. k. @-cal). Ide csinálhatunk egy ablakot, hogy ez ne törlődjék le, majd elkezdhetünk kísérletezni. Nézzük meg, mit találhatunk a karaktergenerátor első nyolc bájtjában! Az adatok: >2000 3C 66 6E 6E 60 62 3C 00. Ennek így első látásra nem sok köze van bármiféle grafikához. Ha azonban egymás alá írva felírjuk a bájtokat binárisan, a „kukac” karakter képét látjuk meg. A karakterek képe ugyanis nyolc bájtban ábrázoltatik, egy-egy bájtban a karakter képének 1—1 sora. A nyolc bit alapesetben és bővített háttérszín módban nyolc képpontot jelent. Az 1-es bit karaktereszínű, a 0-ás háttérszínű lesz a ké-



atdefinialas.....page # 1

```

Line#  Addr Code      Source
00001  0000                ;kar.atdef.
00002  0000                ;
00003  0000                l1      = $d0
00004  0000                h1      = l1+1
00005  0000                l2      = h1+1
00006  0000                h2      = l2+1
00007  0000                intl1   = $0312
00008  0000                inth    = intl1+1
00009  0000                kar     = $4000
00010  0000                int     = $ce42
00011  0000                ;
00012  0000                *      = $2000
00013  0000                ;
00014  2000 78                sei
00015  2001 a2 38                ldx #<mexak
00016  2003 a0 20                ldy #>mexak
00017  2005 8e 12 03                stx intl1
00018  2008 8c 13 03                sty inth
00019  200b a0 00                ldy #0
00020  200d a9 40                lda #>kar ;karakter-keszlet helye
00021  200f a2 08                ldx #8 ;$0800 masolasa
00022  2011 84 d0                sty l1
00023  2013 84 d2                sty l2 ;az uj helyre
00024  2015 85 d3                sta h2
00025  2017 a9 d0                lda #$d0 ;$d000-tol
00026  2019 85 d1                sta h1
00027  201b b1 d0                masol  lda (l1),y
00028  201d 91 d2                sta (l2),y
00029  201f c8                iny
00030  2020 d0 f9                bne masol
00031  2022 e6 d1                inc h1
00032  2024 e6 d3                inc h2
00033  2026 ca                dex
00034  2027 d0 f2                bne masol
00035  2029 ad 12 ff                lda $ff12 ;karkesz ramban
00036  202c 29 fb                and #$fb
00037  202e 8d 12 ff                sta $ff12
00038  2031 a9 40                lda #>kar ;hely beallitasa
00039  2033 8d 13 ff                sta $ff13
00040  2036 58                cli
00041  2037 60                rts
00042  2038                ;
00043  2038                ;
00044  2038 ac 00 40                mexak  ldy kar ;kicsordulo sor
00045  203b a2 00                ldx #0
00046  203d bd 01 40                mas2   lda kar+1,x ;karakter
00047  2040 9d 00 40                sta kar,x ;egy sorral feljebb
00048  2043 e8                inx
00049  2044 e0 07                cpx #7
00050  2046 d0 f5                bne mas2
00051  2048 8c 07 40                sty kar+7
00052  204b ad 12 ff                lda $ff12
00053  204e 29 fb                and #$fb
00054  2050 8d 12 ff                sta $ff12 ;karkesz ram
00055  2053 4c 42 ce                jmp int

```

end of assembly, error count = 00000

h1	00d1	h2	00d3	int	ce42	inth	0313
intl1	0312	kar	4000	l1	00d0	l2	00d2
mas2	203d	masol	201b	mexak	2038		

pen. Írjuk be: >2000 00 3C 46 06 76 76 66 3C. Láthatjuk, hogy a „kukac” fejre áll. Minden sorban megfordítottuk a bitek sorrendjét, és a bájtok sorrendjét is az első nyolc bájtban. Hogyan definiálhatunk át egy másik karaktert? A karakterek képe egymás után folytonosan helyezkedik el a készletben. A TED a karakter megjelenítését úgy végzi, hogy a karakter sor elején beolvassa a képernyőkódokat. Az egyes karakter első bájtja a karaktergenerátorban annyiszor nyolc bájtal van eltolva, amennyi a karakterkód értéke. Tehát: CÍM = karaktergenerátor kezdőcím+8*karakterkód. Ez nem vonatkozik az inverz karakterekre, mivel ezeket a TED állítja elő az alapkarakter képéből. Tehát ha most a képernyőre írunk egy inverz „kukacot”, az is a feje tején áll. Ha azonban az \$FF07 regiszter 7. bitjét 1-re állítjuk (ORA#\$80), az inverz karakterek eltűnnek, és megláthatjuk a 128–255-ig terjedő karakterek képét,

és ezeket is átalakíthatjuk. Ilyenkor azonban hatástalan a $\langle C= \rangle + \langle \text{SHIFT} \rangle$ kombináció mert a TED a karakterkészlet címét csak 2k pontossággal érzékeli.

Karakter átdefiniálással működik az I. listán található program is. Bővített háttérszín mód esetén ugyanezt a felbontást használhatjuk, de minden karakternek külön adhatunk háttérszínt. Ilyenkor a háttérszíninformációt a TED \$FF15, \$FF16, \$FF17, \$FF18 regiszterei szolgáltatják. Ennek az üzemmódnak nagy hátránya, hogy csak 64 féle karaktert használhatunk. A TED ugyanis a karakterkódok felső két bitjét ilyenkor arra használja fel, hogy megállapítsa a karakterhez tartozó háttérszínt. A felső két bit, és a hozzájuk tartozó regiszterek: 00 — \$FF15 01 — \$FF16 10 — \$FF17 11 — \$FF18. Ilyenkor pl. a 25-ös kódú karakter (az „y”) az \$FF15-ös háttérszínnel jelenik meg, a 25+64=89-es kódú szintén „y”, de a háttérszíninformációt az \$FF16 szolgál-

tatja. Az $25+2*64=153$ -as karakter szintén „y”, de az \$FF17 adta háttérszínnel, és végül a $25+3*64=217$ -es karakter az \$FF18-ból nyert alapon jelenik meg. Vigyáznunk kell arra, hogy ne kapcsoljuk be egyszerre a bővített háttérszín és multicolor módot! (A kép gyászszínt ölt.)

A többszínű módban lehetőségünk nyílik arra, hogy egy karakteren belül 4 színt használjunk. Ennek is ára van: hasonlóképpen a többszínű grafikus módozathoz, a vízszintes felbomlás a felére csökken, vagyis egymás mellett 4 darab elhízott pixel van egy karakterben. Az egymás mellett álló bitek párokat alkotnak, és az így kialakuló $4*2$ bit alkotja a négy pixelt. A bitpárosok és a színforrások: 00 — \$FF15 01 — \$FF16 10 — \$FF17 11 — a karakterhez tartozó bájt a szímemóriában. Kalandra fel! Állítsuk az \$FF07 4. bitjét 1-re! Mit látunk? Semmi különös, azon kívül, hogy a kurzor nem villog (pontosabban: semmi sem villog, a szín-

```
finomszkröll.....page # 1
```

```
Line#  Addr  Code      Source
```

```
00001  0000          ;scroll
00002  0000          ;
00003  0000          ;
00004  0000          ;
00005  0000          ;
00006  0000          ;
00007  0000      eltol      = $d0
00008  0000      intl       = $0314
00009  0000      inth       = $0315
00010  0000      irq       = $ff09
00011  0000      raszt     = $ff0b
00012  0000      pont     = $ffle
00013  0000      vez2     = $ff07      ;ted vezerlo 2.
00014  0000      iveg     = $fcc3
00015  0000          ;
00016  0000          ;
00017  0000          ;
00018  0000          x = $2000
00019  2000 78          sei
00020  2001 a2 16      ldx #<mexak
00021  2003 a0 20      ldy #>mexak
00022  2005 8e 14 03   stx intl
00023  2008 8c 15 03   sty inth
00024  200b a9 07      lda #7      ;38 sor, jobb szel
00025  200d 85 d0      sta eltol
00026  200f a9 c2      lda #$c2
00027  2011 8d 0b ff   sta raszt   ;raszter helye
00028  2014 58          cli
00029  2015 60          rts
00030  2016          ;
00031  2016          ;
00032  2016 ad 09 ff   mexak lda irq   ;irq
00033  2019 8d 09 ff   sta irq     ;torlese
00034  201c ad 0b ff   lda raszt
00035  201f c9 c2      cmp #$c2
00036  2021 d0 14      bne alja
00037  2023 a6 d0      ldx eltol
```




```

O 00038 2025 a9 99          lda #$99
O 00039 2027 cd 1e ff      var  cmp pont
O 00040 202a b0 fb          bcs var
O 00041 202c 8e 07 ff      stx vez2
O 00042 202f a9 cb          lda #$cb
O 00043 2031 8d 0b ff      sta raszt
O 00044 2034 4c c3 fc      jmp iveg          ; irq vege
O 00045 2037                ;
O 00046 2037                ;
O 00047 2037                ;
O 00048 2037 c6 d0        alja  dec eltol
O 00049 2039 10 1b        bpl ok
O 00050 203b a9 07        lda #$7
O 00051 203d 85 d0        sta eltol
O 00052 203f ac c0 0f      ldy $0fc0        ; also sor 1. karakter
O 00053 2042 a2 00        ldx #0
O 00054 2044 bd c1 0f      mascik lda $0fc1,x    ;masolo
O 00055 2047 0d c0 0f      sta $0fc0,x      ;ciklus
O 00056 204a e8            inx              ;39 karakter
O 00057 204b e0 27        cpx #$27         ;1-gyel elobbre
O 00058 204d d0 f5        bne mascik
O 00059 204f 8c e7 0f      sty $0fe7
O 00060 2052 a9 07        lda #7
O 00061 2054 85 d0        sta eltol
O 00062 2056 a9 08        ok   lda #$08         ;40 sor, bal szel
O 00063 2058 8d 07 ff      sta vez2
O 00064 205b a9 c2        lda #$c2
O 00065 205d 8d 0b ff      sta raszt
O 00066 2060 20 11 db      jsr $dbl1        ;billentyuzet

end of assembly, error count = 00000

alja 2037 eltol 00d0 inth 0315 intl 0314
irq  ff09 iveg fcc3  mascik 2044 mexak 2016
ok 2056 pont ff1e raszt ff0b var 2027
vez2 ff07

```

memória 7. bitje hatástalan). Ebben a módban lehetőségünk van még arra is, hogy a képernyőn keverjük a multi és a „sima” karaktereket. Hogy egy karakter multi-e vagy sem, azt a hozzá tartozó színmemóriabeli bajt 7. bitje tartalmazhatná, ha már egyszer úgy sem villog ilyenkor semmi... de nem! Nem szabad elfelejtenünk, hogy mi egy COMMODORE gép tulajdonosai vagyunk, a COMMODORE cég pedig sok mindentről híres (vagy inkább hírhedt?), de a logika sosem volt erős oldala. Egy karakter ugyanis akkor jelenik meg multi módban a képernyőn, ha a 3. bitet állítjuk 1-re, vagyis csak 7 szín 8 árnyalatát, plusz a feketét használhatjuk (ez természetesen nem vonatkozik a regiszterekre, ott használhatjuk az összes színt).

Nem szóltam még a finomszkroll-bitekről. Ezek a krakteres képernyő mozgatók használatosak. Erre példát a 2. listán láthatunk.

A bittérképes grafika

BASIC-ből már mindenki használta őket: GRAPHIC1, GRAPHIC3, ez gondolom már ismerős. Nézzük a gép felőli

oldalt! A bittérképes módot úgy képzelhetjük el, mint egy olyan krakteres képet, ahol 1000 féle karaktert használhatunk, de mindegyiknek rögzített a helye. A bal felső sarokban a 0. karakter áll, jobbra mellette az 1., aztán a 2. stb. Lefele pedig negyvenesével lépkedünk. Speciális a színmemória is: két részből áll: a fényerős és a tényleges színmemóriából. Ha belépünk bittérképes módba, a bittérkép helye \$2000-ra kerül, a fényerőmemória \$1800-ra, míg a színmemória (az előzőből következőleg) \$1C00-ra. A bittérkép kijelölése viszonylag bonyolult, ezért álljon itt segítségül egy kis táblázat:

+ 3—5. bitek +	bittérkép helye +
+ 000	+ 0000 — 1FFF +
+ 001	+ 2000 — 3FFF +
+ 010	+ 4000 — 5FFF +
+ 011	+ 6000 — 7FFF +
+ 100	+ 8000 — 9FFF +
+ 101	+ A000 — BFFF +
+ 110	+ C000 — DFFF +
+ 111	+ E000 — FFFF +
+ ----- +	+ ----- +

Minden karakternek két saját színe lehet. Ha nagyfelbontású (320*200) mó-

dot használunk, a háttérszín is és a karakterszín is változhat karakterről karakterre (a továbbiakban krakter alatt „karakterpozíció” álló 8 bajt”-ot értek). Többszínű módban (160*200) két-két szín különbözhet karakterenként (a 01 és 10 bitpárok által kódolt színek), míg a 00 bitpár háttérszínű (\$FF15) lesz, míg az 11 bitpár a BASIC-ből 3. színüként ismert pontot jelent, mely színét az \$FF16-ból nyeri. A 01 bitpár a színt a színmemória 4—7., fényerejét a fényerőmemória 0—2. biteiből kapja, míg az 10 páros a 0—3., illetve a 4—6. bitekből.

Több színt is használhatunk a rasztermegszakítás használatával úgy, hogy képernyőlefutásonként váltogatjuk a színmemóriát, de vigyázzunk, így képet elrontani könnyebb, mint jól csinálni („kevert” színek nagyon tudnak remegni!). A rasztermegszakítás arra is használhatjuk, hogy soronként változtassunk a színeken, így többszínű módban 8 sor*4 szín = 32 színt használhatunk, de ez a gépet nagyon lassítja.

Nézzünk most egy példaprogramot, amely bemutatja, hogyan lehet gépi kód-
ban grafikát programozni, de amely a

pont02.....page # 1

Line#	Addr	Code	Source
00001	0000		;turbo-dot
00002	0000		;
00003	0000		;
00004	0000		;
00005	0000		xh = \$d0
00006	0000		xm = \$d1
00007	0000		ym = \$d2
00008	0000		yh = \$d3
00009	0000		yl = \$d4
00010	0000		xk = \$d5
00011	0000		al = \$d6
00012	0000		ah = \$d7
00013	0000		hb = \$d8
00014	0000		;
00015	0000		* = \$1200
00016	1200		;
00017	1200	08	php ;rajzolas/torles mentese
00018	1201	c0 c8	cpy #200 ;kepen kivul?
00019	1203	b0 4d	bcs kint+1 ;igen
00020	1205	85 d0	sta xh ;x magas mentese
00021	1207	e0 40	cpx #\$40
00022	1209	e9 01	bcs #\$01 ;x>\$140 (320)?
00023	120b	b0 45	bcs kint+1 ;igen
00024	120d	8a	txa ;xm=x mod 8
00025	120e	29 07	and #7 ;oszlop
00026	1210	85 d1	sta xm ;karakterben
00027	1212	98	tya ;ym=y mod 8
00028	1213	29 07	and #7 ;sor
00029	1215	85 d2	sta ym ;karakterben
00030	1217	a9 00	lda #0
00031	1219	85 d4	sta yl ;yl torles
00032	121b	98	tya ;yk=y div 8
00033	121c	4a	lsr a
00034	121d	4a	lsr a
00035	121e	4a	lsr a
00036	121f	85 d3	sta yh ;yh=yk*\$100
00037	1221	4a	lsr a ;yk*\$40
00038	1222	66 d4	ror yl ;kepzese
00039	1224	4a	lsr a
00040	1225	66 d4	ror yl
00041	1227	18	clc ;+\$100*yk
00042	1228	65 d3	adc yh
00043	122a	85 d3	sta yh ;y*\$140 yl,yh-ban
00044	122c	8a	txa ;8*xk=
00045	122d	29 f8	and #\$f8 ;=8*x div 8
00046	122f	85 d5	sta xk ;cimhez
00047	1231	18	clc
00048	1232	a5 d4	lda yl
00049	1234	65 d5	adc xk
00050	1236	85 d6	sta al ;cim also
00051	1238	a9 20	lda #\$20 ;grafkep \$2000-tol
00052	123a	65 d3	adc yh
00053	123c	65 d0	adc xh
00054	123e	85 d7	sta ah ;cim felso
00055	1240	a6 d1	ldx xm
00056	1242	a4 d2	ldy ym
00057	1244	b1 d6	lda (al),y
00058	1246	1d 54 12	ora tabla,x ;pont be
00059	1249	28	plp
00060	124a	b0 03	bcs kiir ;rajzol
00061	124c	5d 54 12	eor tabla,x ;pont kioltasa
00062	124f	91 d6	kiir sta (al),y


```

00063 1251 24 28      kint  bit $28      ;(plp)
00064 1253 60                rts
                80 40 20 10 08 04 02 01
00065 1254      tabla .byt $80,$40,$20,$10,8,4,2,1

end of assembly, error count = 00000

ah      00d7      al      00d6      hb      00d8      kiir     124f
kint    1251      tabla   1254      xh      00d0      xk      00d5
xm      00d1      yh      00d3      yl      00d4      ym      00d2

```

példálódzáson kívül is hasznos lehet! Ez a program egy gyors pontrajzoló HI-RES (High RESolution — nagyfelbontású) grafikus képre. Ugyan a BASIC ROM is tartalmaz ilyen rutint, az azonban kissé lassan fut. Ez a rutin nem a létező leggyorsabb (hallottam egy olyanról, amely körülbelül 60 óraciklus alatt rajzol ki egy pontot), viszont védett túlsordulás ellen (vagyis nem hibázik, ha 319-nél nagyobb vízszintes, vagy 199-nél nagyobb függőleges koordinátát adunk meg) és rajzolásra/törlésre egyaránt használható. Működése:

A rutin meghívása előtt a carry bitben kell megadni, hogy rajzolni (SEC) vagy törölni (CLC) akarjuk-e a pontot. Ezt a program elmenti (PHP), majd ellenőrzi, hogy nincs-e túlsordulás. Ha van, elugrik a rutin végére, ahol egy PLP utasítás van elrejtve egy BIT \$28-ban. Ilyen-

kor végrehajtja a PLP-t (hiszen a BIT \$28 második bájtyára ugrik, ami \$28, azaz a PLP kódja), majd RTS-sel visszatér. Ha egyszerűen a PLP-re ugranék, a veremből rossz visszatérési értéket kapna a gép, hiszen szubrutinon belül írtunk bele. E kis kitérő után folytassuk azt az ágat, hogy nem volt hiba a paraméterekben. A pontot ugyebár egy bit ábrázolja a bájton belül. A bájty címüt a GRAFIKUS-KÉP CÍME + (Y div 8) * 320 + Y mod 8 + + 8 * (X div 8) képlet adja meg, ahol div (divízió — osztás) maradék nélküli egész osztást jelent, vagyis 7 div 4 = 1, mod (modulus — maradék) pedig az osztás maradéka, vagyis 7 mod 4 = 3. A bit sorszáma a 7 — X mod 8 művelet eredménye, ugyanis a 7. bit a bal szélső, nem a nulladik. Az osztás műveletet LSR utasítással lehet megoldani, a mod 8 pedig AND#7 gépi kódban. A 8*(X div 8)

pedig X AND#\$F8, hiszen az alsó három bitet vágjuk le.

A div művelettel a pont karakterpozícióját számítjuk ki. Ezen képletek alapján számolja ki a program a címeket, majd a bitnek megfelelő maszkot kiolvassa egy táblázatból. A maszkkal elvagyolja az eredeti bájtot, majd visszaolvassa a feltételregisztert, és eldönti, kell-e törölni a pontot. Ha nem, kiírja, ha igen, eEOR-ozza a maszkkal. Mivel előzőleg a bitet 1-re írtuk, az EOR azt 0-ba viszi, vagyis készen vagyunk, kiírhatjuk a bájtot a képernyőre. Ezekután a PLP-t átlépjük a BIT utasítással, és visszatérünk a hívó programba. Aki nem hiszi, számoljon utána!

Köszönöm e havi figyelmüket, a következő cikk a hangról fog szólni, és bemutatásra kerül egy digitalizátor program is.

```

O  FACHO = $62      O
O  RESULT = $0100  O
O  WRITXT = $AB1E  O
O  INTFLP = $BC49  O
O  FACASC = $BDDF  O
O  ;              O
O  ;              O
O  ...           O
O  LDA #$81      O
O  LDX #$7D      O
O  JSR HDCONV    O
O  LDA #<RESULT  O
O  LDY #>RESULT  O
O  JSR WRITXT    O
O  ...           O
O  ;              O
O  ; ==          O
O  HDCONV = *    O
O  STA FACHO     O
O  STX FACHO+1   O
O  LDX #$90      O
O  SEC           O
O  JSR INTFLP    O
O  JSR FACASC    O
O  RTS           O

```

Hex-Dec konvertálás C64 assemblyben

Assembly programozás közben hasznos szokás az interpreter részeinek szubrutinként történő felhasználása. Lerövidíthető így a tizenhatosból tízes számrendszerbe történő átszámítás munkája is, ahogy az a HDCONV szubrutinban látható. Bemeneti paraméterként az A/X regiszterekbe kell tölteni a szám alsó/felső byte-ját. Eredményként egy 1–5 karakter hosszú számot kapunk a verem elején, egy \$00 byte-tal lezárva. A példában megadott \$7D81 átszámításakor a \$0100 címtől kezdve a következők lesznek a memóriában: \$33 32 31 32 39 00. A szám első jegyé mindig a \$0100-en lesz.

A példa egyben egy kis trükköt is bemutat. Könnyen adódhat olyan helyzet, amikor csak egy byte átszámítását akarjuk, de mindig 3 jegyű számmá, vagy éppen nem akarunk bajlódni az eredmény jegyeinek számolgatásával. Ilyenkor a bemeneti adat felső byte-ját \$7D (125) értékkel kell feltölteni. Ennek hatására az eredmény ötjegyű, és az alsó byte átszámítása a \$0102–04 byte-okon lesz megtalálható.

Még többet ésszel!

VIII. rész

Alighogy leadtam az előző rész kéziratát, megjelent a 92/9. szám, benne egy igen érdekes cikkel a ciklusokról a 28–29. oldalon. Mulatságos volt felfedezni ezt az egybeesést, amely egyáltalán nem káros, és talán nem is unalmas. Néhány hónappal későbből tehát ajánlom ezt a cikket elsősorban azok számára, akiknek valami nehézséget okozott a VII. rész megértése. Bár a cikk (ismeretlen) szerzője vagy fordítója esetenként más szak kifejezéseket használ (például a hurkok dobozolását én ciklusok egymásba ágyazásának nevezem), az eltérés nem nehezíti a megértést.

Folytassuk a múltkori gondolatsort a FOR ciklusról! Gyakran látom, hogy a különféle programok szerzői takarékoságból elhagyják a NEXT után a ciklusváltozó megnevezését. Ezt sok BASIC verzió megengedi, egy kevés helyet és futási időt lehet vele nyerni. A gyakoriságát mégis inkább az emberi kényelmség okozza, azt hiszem. Persze ezt azért hozom szóba, mert nem helyeslem a módszert.

Most is van egy esztétikai és egy technikai ellenérvem. Ha egy programban ráakadok egy „névtelen”, változó nélküli NEXT-re, esetleg alapos elemzésre van szükségem, hogy kiderítsem, melyik ciklus is fejeződik be éppen most. Ez nem kedvez az olvashatóságnak, ennek következtében a fejlesztés során maga a programozó is hibázhat. Peresze nem valami szép az olyan program, ahol egy ciklus nagyon sok sorból áll, de egy tízoros rutin is megírható olvashatatlanul. Talán jó lenne, ha minden utasítás külön sorba kerülne, ezáltal valamivel világosabb lehetne a program képe, de ugyanakkor szörnyű terjedelmes is. Marad tehát a szokásos tördelés, amely azonban sokszor azt eredményezi, hogy egy NEXT elbújik valahol az utasítások között. Azok tehát, akik macacsul elhagyják a NEXT változóját, legalább próbálják meg a NEXT-et mindig a sor elejére írni, hogy látni lehessen. De ez még mindig inkább csak az esztétikai oldal.

A BASIC szabályai szerint egy névtelen NEXT a legutoljára megnyitott ciklusra vonatkozik. A gép nem képes kitalálni, hogy melyik ciklust óhajtom lezárni, ha nem közlöm vele. Ebből aztán durva hibák születhetnek, ahogy az az 1. és 2. példákban, kissé laboratóriumi tisztasággal megfigyelhetők. Megfelelő körülmények között a 71. és 8. sorokban a NEXT az először megnyitott ciklust zárja le, hibásan. A ciklusok biztonságos elhagyásának technikájához tartozik tehát a ciklusok egyértelműsége is.

Lehet, hogy azt gondolják néhányan: senki sem olyan bágyú, hogy a ciklus megírásakor elfeledkezze arról, hogy az előbb kiugrott belőle, és ebből baj lehet. És aki egy héttel korábban megírt ciklusába szór be egy elágazást? Igaz, egy figyelmes és gyakorlott programozó nem követ el ilyen hibákat, de a tanácsokat nem is nekik szántam.

Egy eset van, amikor elfogadhatónak tartom a névtelen NEXT-eket: az egysoros ciklusokban. Am itt sem szép, és felesleges a formabontás egy ilyen ritka eset kedvéért.

A témakör lezárása előtt egy pillanatot időzzünk az egymásba ágyazott ciklusok kérdésénél. A veremről elmondottak fényében most már bizonyára nyilvánvalónak látszik, hogy a rosszul, átfedéssel egymásba illesztett ciklusok (3. pl.) nem egyszerűen logikai, elvi hibát tartalmaznak, hanem a gép a végrehajtás során hibát is talál. Tehát nemcsak az a baj, hogy a J-ciklus mindig

csak elkezdődik, hanem az I-ciklus befejeződése után a gép megszünteti a belső, J-vel megnyitott ciklust is, aminek hatására a NEXT J már nem hajtható végre.

Éppen ideje már, hogy szóba hozzam az előző rész feladványait. Nézzük meg együtt a 4. példát, amelyről azt állítottam, hogy jó! Persze azért ez túlzás volt, de a mostanában megtárgyalt hibatípus legalábbis nincs meg benne. Az elmondottak alapján azt várnánk, hogy 8–9. sorok ciklusa hatására a 3. sorban levő NEXT-hez már nem lesz FOR, és megvan a hiba. Ha kipróbáljuk, a várttal ellentétben a program hibajelzés nélkül lefut.

Roppant kíváncsi lennék, hogy hány olvasónak jutott eszébe, hogy a GOSUB ugyanazt a vermet terheli, mint a FOR. Akkor ezek hogy nem zavarják egymást? Sőt, ha betartjuk az eddigi elvet, akkor a 8. sor FOR-ja hatására nemcsak az előző FOR adatainak, hanem a később végrehajtott GOSUB visszatérési adatainak is le kellene dobódnia a verem tetejéről. Ennek következtében a 9. sorban levő RETURN már nem lenne végrehajtható, és vagy hibáüzenetet kapnánk, vagy az egy szinttel korábbi hívás helyére ugranánk vissza. Pedig a gyakorlat nem ezt mutatja.

Aki ért valamit az assemblyhez, az egy monitorral megkeresheti a megoldást. A C64 interpreterében a \$49/4A címen átadott változónévvel megnyitott ciklus adatait keresi a veremben. Ez a szubrutin csak addig keres, amíg a FOR tokenjétől eltérő jelzést nem talál. Márpedig a GOSUB hatására a verembe íródó adatok épp ilyen eltérő jelzést produkálnak.

Közérthetőbben: ha egy ciklust egy szubrutinban nyitok meg, akkor a gép a vermet csak a legutolsó GOSUB-ig nézi át, azonos változóra nyitott ciklust keresve. Ezért nem veszi észre a 4. példában, hogy egy I-ciklus már nyitva van, és megnyit egy másikat is! Ezek után természetes, hogy amikor befejezi a szubrutinbeli ciklust, a visszatérés után zavartalanul folytatja a „mester”-ciklust is.

Vagyis majdnem zavartalanul. Habár úgy tűnhet, hogy a ciklusban a gép lokálisan kezeli a változókat, még sincs így. Ez csak egy mikrogép, itt a változóknak csak egyetlen érvényességi köre létezik: a globális. Ez azt jelenti, hogy bármelyik változó értékét szubrutinból, ciklusból, akárhonnán ki lehet olvasni vagy át lehet írni. Ha tehát a 8. sorban a FOR-ral az I értékét beállítom, akkor azt az I-t írtam felül, amelynek egy már korábban meghatározott értékére a 3. sorban, a NEXT-nél számítottok. A két ciklus ugyanazt a változót használja, ami megengedett, de nem valami értelmes dolog. Így a főciklus nem fog háromszor lefutni, hanem ez esetben csak egyszer. Éppenséggel pontosan ezt használtuk ki a múltkor a ciklusból való kilépéskor.

Érdekes kísérlet, ha az 1. sorban a végértéket 3-nál nagyobbra állítjuk be. A szubrutinból I 3-mal tér vissza, minden esetben, ami kevés a külső ciklus befejezéséhez, kész tehát egy végtelenített ciklus.

Van még két olyan típushelyzet, ahol megfigyelhetjük a szubrutinhívás lokalizáló hatását a FOR ciklusra. Az 5. példában a NEXT nem talál I-ciklust, mert a GOSUB-nál abbahagyja a keresést. A 6. példában a szubrutinban a verembe tett FOR a visszatéréskor ledobódik a veremről. A RETURN-re a gép megkeresi a legutolsó szubrutinhívás bejegyzését, kiolvassa az ott talált visszatérési paramétereket, majd kivesszi őket a veremből, mert már nincs rájuk szükség. Ezzel persze megszünteti a FOR később beírt adatait is. Nagyon hasonlít tehát ez az eljárás a FOR ciklus befejezésekor látottakra.

Összeállt tehát egy fontos szabály a ciklusokat és szubrutini-


```
(1.)
20 FOR I=4 TO 7
22 IF D>5 THEN 35
24 NEXT
35 ::
70 FOR J=1 TO 3
71 PRINT S(J): NEXT
```

```
(2.)
1 FOR I=1 TO 5
2 FOR J=1 TO 3
6 IF C THEN NEXT
8 NEXT
```

```
(3.)
1 FOR I=1 TO 3
2 FOR J=2 TO 4
3 PRINT I, J
4 NEXT I
5 NEXT J
```

```
(4.)
1 FOR I=1 TO 3
2 GOSUB 8
3 NEXT I: END
8 FOR I=1 TO 2
9 NEXT I: RETURN
```

```
(5.)
1 FOR I=1 TO 3
2 GOSUB 8
3 END
8 NEXT I
9 RETURN
```

```
(6.)
1 GOSUB '8
2 NEXT I
3 END
8 FOR I=1 TO 3
9 RETURN
```

```
(7.)
20 IF A THEN IF B THEN PRINT "*"
```

nokat csak szigorúan egymásba ágyazva, ha úgy tetszik dobozolva szabad megszerkeszteni. Ettől eltérve vagy logikai, vagy technikai (futási) hibát okozunk. Ennyit a FOR ciklusról.

A 7. példához hasonlólt láttam a korábban említett cikkben, és ez eszembe juttatott valamit. Kérdésem: pontosan ugyanezt a kiválasztást hogyan lehet másképpen megcsinálni, és mi a különbség a megoldások között?

A másik feladathoz kérlek, kedves Olvasó, vedd szemügyre a 92/5. szám 8. oldalán levő kis programot. Elnézést kérek

Mesterházi Sándortól, de munkájában sajnos hibát találtam. A legközelebbi részben ezt a programot fogom elemezni, remélhetőleg közérdekű tanulságokat leszűrve. Lehetséges, hogy ez el fog vezetni a lemezkezelés vagy a Kernal rutinok megtárgyalásáig, majd meglátjuk. Ígéretemhez híven továbbra is csak a BASIC programozásban felhasználható eszközökről beszél-

Hódi Gyula

Eladó egy C 64, 1541/2 Action replay V6, magnó, 170 lemez, kazetták, sok szakkönyv. Irányár: 27 000 Ft. Írjál! Piller Dávid, 2040 Budapest, Semmelweis u. 19.

Here is the disk news to 64. Magyar nyelvű lemezújság. Küldj egy lemezt 30 Ft-os bélyeggel, ingyen rávesszük. Faragó Imre - Jerry, 1046 Budapest, Külső Szilágyi u. 20.

Eladó Amiga 500-hoz 0,5 Mb órás bővítő 3000 Ft, +4-hez: 8360-as 2000 Ft, 8501-es 1500 Ft, 6526-os 1000 Ft, 251641-es 1000 Ft. Kiss György, 6000 Kecskemét, Bajza 13. IV/12.

C-64-esek! Akartok 1296 db kazettát ingyen tele programokkal? Nem kell más tenned, csak küld el címedet és egy 15 Ft-os bélyeget! Akkor én elküldöm, mit kell tenned. Címem: Bota Zsolt, 3397 Maklár, Vasút u. 16.

Amigára a legújabb programok! Válaszborítékért tájékoztatót küldünk! SzT, Bp. Pf.: 88 1683.

Profi-Ass 64 és Profi-Mon 64 programokért eprom törlés vagy C 64 program égetés. Tel.: 140-38-20.

C-64 + 1541 typ. floppy speeddossal (10-szeres gyorsaság) + sok játék és felhasználói programok eladó. Ára 21 000 Ft. Kérném IBM-re a Power Basic programot. Tel.: 1202-155. 18 óra után.

HELLO C-64 hardware-szerelmeseink!!! Ha neked kéne 4 db 8 KB-os eprom, akkor olcsó áron megkaphatod tőlem, 1 db 100 Ft, de ha mind a négyet megveszed, akkor csak 350 Ft a 4 db. Van még egy ötödik is, de annak egy lába hiányzik, ennek az ára: 50 Ft. Cím: Nagy Roland, 8000 Székesfehérvár, Benke F. u. 13. fszt. 1.

Az EPROM BT A Commodore 64-re + az ismert D&T cartridge családja mellett kifejlesztett egy teljesen új, két részből álló modulrendszer.

Részei:

1./ BASIC MODUL

Jelölése: BM
Helye: a C-64 ROM bővítő portja
Tartalma: - NYÁK csatlakozó, a cserélhető modulok számára
- RESET gomb
Fogadhat: 1 vagy 2 — csak cégünk által programozott — IC-t

A cserélhető modul helyes csatlakoztatása mechanikusan biztosított.

2./ CSERÉLHETŐ MODUL

Helye: a BASIC MODUL NYÁK csatlakozója
Megnevezése: játékprogramok esetén
GAME MODUL
1 IC-vel 85—100 Kbyte terjedelem
4 változat, 4—5 program/változat
2 IC-vel 170—200 Kbyte terjedelem
6 változat 8—10 program/változat
Szoftveres menüváltás és modulkikapcsolás.

Minden termékre 6 hónap garancia + használati útmutató.

Árak: BASIC MODUL 1100 Ft
GAME MODUL 1000 Ft 1 IC-vel
2000 Ft 2 IC-vel

Az árak az ÁFA-t tartalmazzák!

Megrendelhető: EPROM BT.

1046 Budapest, Török I. u. 25.

Tel.: 1690-779 Postai utánvétel!

AZ 5-RE VEGZODO SORSZAMU SOROKAT NEM KELL BEIRNI!

```

10 REM *****
20 REM * OROKNAPTAR *
30 REM * KESZITETTE: *
40 REM * LOOS JOZSEF *
50 REM * COMMODORE PLUS/4 GEPRE. *
60 REM *****
70 VOL 7
80 PRINT "*****" TAB(12); "*****"
85 REM [CLR] 4X[LE] [C=/A] 12X[SHF/*] [C=/S]
90 PRINT TAB(12); " | OROKNAPTAR | "
95 REM [SHF/B] [SHF/B]
100 PRINT TAB(12); " | "
105 REM [C=/Z] 12X[SHF/*] [C=/X]
110 PRINT "*****" TAB(14); "(SOOLSOFT)":SOUND 1,980,2
115 REM 7X[LE]
120 FOR MM=1 TO 1300:NEXT MM
130 REM -----
140 PRINT "C":C=1
145 REM [CLR]
150 PRINT "MELYIK EVET KERESI (EVSZAM+RETURN) ":INPUT " ";P:SOUND 1,890,6
155 REM [JOB] 2X[JOB]
160 REM
170 REM **** OROKNAPTAR *****
180 REM
190 PRINT "EV: HONAP: "
195 REM [CLR] 2X[LE] 2X[JOB] 4X[SPC] 2X[JOB]
200 PRINT " ";P
205 REM [HOME] 2X[LE] 5X[JOB]
210 IF C=2 THEN PRINT "*****";HO:GOTO 230
215 REM [HOME] 2X[LE] 17X[JOB] 4X[SPC] 4X[BAL]
220 INPUT "*****";HO:HH=HO
225 REM [HOME] 2X[LE] 17X[JOB] 4X[SPC] 4X[BAL]
230 IF HO>12 OR HO<1 THEN GOTO 220
240 PRINT "*****"
245 REM [HOME] 4X[LE] 40X[-]
250 PRINT "*****"
255 REM [HOME] 4X[LE]
260 PRINT "HETFOKEDD SZERCSUT PENT SZOMIVAS."
265 REM 3X[JOB] [JOB] ... [JOB] [CTR/<] [CTR/>]
270 YY=31:GOSUB 320:NA=1:EV=P:GOSUB 470:PRINT "*****"
275 REM [HOME] 7X[LE]
280 FOR G=2 TO WW:PRINT " ";:NEXT:PRINT "01":;G=0
285 REM [JOB]
290 IF IN<6 THEN FOR G=0 TO 5-IN:PRINT "0"+CHR$(50+G);:NEXT
295 REM 3X[JOB]
300 PRINT:PRINT:LL=0:G=G+1:GOTO 380
310 REM -----
320 IF HO=4 OR HO=6 OR HO=9 OR HO=11 THEN YY=30
330 IF HO=2 THEN YY=28
340 IF P/4=INT(P/4) AND HO=2 THEN YY=29
350 IF P/100=INT(P/100) AND P/400<>INT(P/400) AND HO=2 THEN YY=28
360 RETURN
370 REM -----
380 FOR JJ=1 TO 5:PRINT " ";
385 REM [JOB]
390 FOR KK=1 TO 7:G=G+1:M$=CHR$(48+LL)+CHR$(48+G)
400 IF VAL(M$)>YY THEN 440

```

ÖRÖK- NAPTÁR


```

410 PRINT "HONAP:";M$;:IF G=9 THEN G=-1:LL=LL+1
415 REM 3X[JOBBS]
420 NEXT KK:PRINT:PRINT
430 NEXT JJ
440 PRINT:PRINT"-----"
445 REM 40X[-]
450 SOUND 1,890,4:SOUND 2,340,3
460 GOTO 600
470 REM ***** SZAMITASOK *****
480 IF HO<3 THEN EV=EV-1:HO=HO+12
490 SS=INT(365.25*EV)-INT(EV/100)+INT(EV/400)+31*(HO-1)-INT(.4*HO+2.3)+NA
500 EV=SS+1721060:IN=EV-INT(EV/7)*7
510 IF IN=0 THEN WW=5:GOTO 580
520 IF IN=1 THEN WW=10:GOTO 580
530 IF IN=2 THEN WW=15:GOTO 580
540 IF IN=3 THEN WW=20:GOTO 580
550 IF IN=4 THEN WW=25:GOTO 580
560 IF IN=5 THEN WW=30:GOTO 580
570 IF IN=6 THEN WW=35:GOTO 580
580 IF IN=6 THEN WW=35
590 RETURN
600 REM -----
610 PRINT "LAPOZAS ELORE: M"
615 REM 2X[JOBBS]
620 PRINT "LAPOZAS HATRA: Z"
625 REM 2X[JOBBS]
630 PRINT "UJ KERESESE : A"
635 REM 2X[JOBBS]
640 PRINT "VEGE : V"
645 REM 2X[JOBBS]
650 GETKEY A$:SOUND 1,980,2
660 IF A$="M" THEN HH=HH+1:GOTO 720
670 IF A$="Z" THEN HH=HH-1:GOTO 720
680 IF A$="A" THEN GOTO 140
690 IF A$="V" THEN GOTO 750
700 IF A$<>"Z" OR A$<>"M" THEN GOTO 650
710 IF A$<>"A" OR A$<>"V" THEN GOTO 650
720 IF HH=13 THEN HH=HH-12:P=P+1:GOTO 740
730 IF HH=0 THEN HH=HH+12:P=P-1
740 C=2:HO=HH:GOTO 190
750 PRINT "VEGE":SOUND 1,890,2:END
755 REM [CLR] 2X[LE] 2X[JOBBS]

```

EV: 1992 HONAP:? 12

HETF KEDD SZER CSUT PENT SZOM VAS.

	01	02	03	04	05	06
	07	08	09	10	11	12
	14	15	16	17	18	19
	21	22	23	24	25	26
	28	29	30	31		

LAPOZAS ELORE: M
LAPOZAS HATRA: Z
UJ KERESESE : A
VEGE : V

1. A program kezelése:

Az év és hónap megadása után
M — lapozás előre
Z — lapozás hátra
A — új év, hónap keresése
V — vége

2. A program célja: naptári napok megismerése.

3. A program műfaja: felhasználói program.

4. A program leírása:

Az év és hónap megadása után a gép a képernyőre kiírja a keresett hónap napjait, adatait. Ebben a naptárban, előre és hátra is lehet lapozni. Nagyobb lapozás, ugrás esetén új adat is kereshető. (Másik év, hónap.)

5. Programozási feladatok megoldása:

— Határvédelmek: 12-nél nagyobb hónapszámot nem fogad el a program.

6. Megjegyzés:

— Az évet és hónapot számmal kell megadni a gépnek! (1992. 02.)

Helló fiúk, lányok, C-64-esek, Amigások, +4-esek és még sokan mások! Nem kéne csinálni egy számítógépes találkozót? Ha van valami ötleted, akkor légy szíves írd meg. A cím: T.K.S., 8000 Székesfehérvár, Benke Ferenc u. 13. fszt. 1.

C-64-re kazettára, elemezre német és más oktató programok szótárak, könyvelői, felhasználói programok. Játékok már 5 Ft-tól. Játékleírás

40 Ft/gépelte oldal. Egyedi program készítését vállalom. Borítékért tájékoztatom. 5000 Szolnok, Győrffy I. u. 12. I/7.

Hi 64-esek! Hatalmas akció!!! Csak most és csak nálunk!!! Vegye meg most karácsonyi ajándékát!!! 64 Kbyte-os, és 32 Kbyte-os bővítőkártyák egyedi programösszeállítással. De ha akarja, akkor megváltoztatjuk a kártyán lévő prog-

ramokat (max. 240 blokk). Zenélős főmenü, szép kivitel, bővebb tájékoztató: T. K. S., 8000 Székesfehérvár, Benke F. u. 13. fszt. 1.

Eladó Apple Macintosh, belső és külső 3.5" floppyval, Imagewriter grafikus printerrel, sok programmal demókkal, egerrel és Inside Macintosh I. II. III. könyvvél és fejlesztő programcsomaggal. Irányár: 40 000 Ft. Telefon: 1863-914, este.

Amigaa programok eladók 10 Ft/db. 5.25" lemezre is. Listához, tájékoztatóhoz szükséges 1 lemez és 20 Ft-os bélyeg. Lemez nélkül 80 Ft. Bogár József Zsolt, 1183 Budapest, Széchenyi u. 28.

Yo Amigások!!! Nem akartok 1250 db lemezt 92-es szuperprogramokkal? Ha igen, akkor írj és elküldöm a tájékoztatót!!! A címem: Cuddy, 8000 Székesfehérvár, Benke F. u. 13. fszt. 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

PÖTYÖGŐ SZOLGÁLAT

261	+4	91/ 2	80-as képernyő	60.-
262	C64	91/ 2	Fizikateszt	40.-
263	C64	91/ 2	40 Sprite	100.-
264	C64	91/ 2	Raster - Master	60.-
265	C64	91/ 3	Rullett	80.-
266	+4	91/ 3	Tape-disc copy	90.-
267	C64	91/ 3	BASIC rearranger	40.-
268	C64	91/ 3	BASIC decompressor	40.-
269	C64	91/ 3	BASIC merger	40.-
270	C64	91/ 3	SEIKOSHA	60.-
271	C64	91/ 3	Cartridge szimuláció	100.-
272	+4	91/ 3	Sally kulcsszókereső	40.-
273	+4	91/ 3	Merge C+4	60.-
274	C128	91/ 4	1.5 MHz-es C64	80.-
275	C64	91/ 4	Morze - H	60.-
276	C64	91/ 4	GPrint	40.-
277	C64	91/ 4	ASCII - CHR\$	40.-
278	C64	91/ 5	Rendező	60.-
279	C64	91/ 5	Keretbeíró	50.-
280	C64	+4	NLO Print	50.-
281	C64	91/ 5	FLD	40.-
282	C64	91/ 5	Raster Split	40.-
283	+4	91/ 6	Programnyilvántartó	60.-
284	C64	91/ 6	Regisztrer	50.-
285	C64	91/ 6	Digitalizált zene	40.-
286	C64	91/ 6	Ekezetes MPS 802	60.-
287	C64	91/ 7	*Ekezetes iratkeszítő	80.-
288	+4	91/ 7	Büvös téglalap	80.-
289	C64	91/ 7	Dominó - H	80.-
290	C64	91/ 7	Raktáros	60.-
291	+4	91/ 7	Tologató	50.-
292	C64	91/ 7	Aren 2000	40.-
293	C64	91/ 7	Bad Lamps	60.-
294	C64	91/ 7	Operation Patriot	40.-
295	+4	91/ 7	Mozaik - B	70.-
296	+4	91/ 7	Logi Kód	50.-
297	C64	92/ 1	Databevivő	40.-
298	+4	91/ 7	Bomber	40.-
299	+4	91/ 7	Dominó - K	40.-
300	+4	91/ 7	The Wall	30.-
301	C64	91/ 7	Mozaik - K	40.-
302	+4	91/ 7	Dominó - R	50.-
303	+4	91/ 7	*Borgok Kincse	150.-
304	+4	91/ 7	Asteroids	50.-
305	C64	91/ 9	Ekezetes irat forrás	30.-
306	C64	91/ 9	Yugo2	40.-
307	C64	91/ 9	Adventure - Sz	100.-
308	C64	91/ 9	Morze Oktató B.	100.-
309	+4	91/ 9	Memory Game Plus	80.-
310	C64	91/ 9	*Capitaly	100.-
311	C64	91/ 9	*Gengszterek	50.-
312	C64	91/ 9	*Küldetés 2000	80.-
313	C64	91/ 9	Itt a \$ hol a \$	40.-
314	C64	91/ 9	Hewing	40.-
315	+4	91/10	Mikrobi	40.-
316	+4	91/10	Nest of Fleas	150.-
317	+4	91/10	Adventure - F	80.-
318	+4	91/10	A világ szeme	120.-
319	C64	91/10	Gondolközé	80.-
320	C64	91/10	Memori - B	100.-
321	+4	91/10	Matekastély	80.-
322	C64	91/11	Lottó	60.-
323	C64	91/11	Totó	60.-
324	C64	91/11	Német gyakorló	50.-
325	+4	91/11	Beszélné szótár	100.-
326	+4	91/11	File examiner	50.-
327	C64	91/11	Sprite szerkesztő	40.-
328	C64	91/11	Egyszínű Sprite	40.-
329	C64	91/12	Rullett javítás	40.-
330	C64	91/12	Tözsde	100.-
331	C64	91/12	IQ teszt	120.-
332	C64	92/ 1	Kalandlap	60.-
333	C64	92/ 1	Lévélyfő	60.-
334	C64	92/ 1	Képernyőmaszkoló	100.-
335	C64	92/ 1	Sprite Kezelő	80.-
336	C64	92/ 1	SI - OPART	30.-
337	C64	92/ 2	Patience Quartet	100.-
338	C64	92/ 2	Monopoly	100.-
339	C64	92/ 2	Akasztófa	60.-
340	C64	92/ 2	Karaktertervező	50.-
341	C64	92/ 3	String Array Manager	80.-
342	C64	92/ 3	Sprite Basic	40.-
343	C64	92/ 3	Screen Help	40.-
344	C64	92/ 3	Gyors validate	50.-
345	C64	92/ 3	BASIC SYNTAXER I.	150.-
346	C64	92/ 4	Tologató C64-re	80.-
347	C64	92/ 4	ABC rendező	80.-
348	C64	92/ 5	*Matek + Római számok	80.-
349	+4	92/ 6	Tesztkészítő + értékelő	80.-
350	C64	92/ 3	BASIC SYNTAXER II.	150.-
351	+4	92/ 7	Három játék	90.-
352	+4	92/ 7	Szókés	60.-
353	C64	92/ 7	Toto	120.-
354	C64	92/ 7	DIGIT analízis program	90.-
355	+4	92/ 7	SCREEN DUMP	120.-
356	C64	92/ 9	UNISEQ filekezelő	100.-
357	C64	92/ 9	Graphic Copy	50.-
358	C64	92/10	Printfox Resizer	100.-
359	C+4	92/11	Tologató II.	120.-

A *-gal jelölt programok csak lemezen rendelhetők !

Kérem, hogy a **Comodore Újság** legközelebbi számában jelenjen meg a következő szövegű apróhirdetés:

(minden kockába egy betűt írjon)

Tagsági szám:

Feladandó az újság címére:

Comodore Újság

Budapest, 1388. Pf. 86.

Alulírott megrendelem a következő programokat a PÖTYÖGŐSZOLGÁLAT-tól:

PROGRAM SORSZÁMA	ÁRA
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
Összesen:	db

A programokat a SZOLGÁLAT által biztosított lemezre (99 Ft/db)

kazettára (40 Ft/db)

az általam küldött adathordozóra kérem.

(Kérjük a megfelelő szöveg aláhúzását!)

Postaköltség (35 Ft)

A fizetés módja: személyesen — csekk — utánvéttel

(Kérjük a megfelelő szöveg aláhúzását!)

ÖSSZESEN:

A megrendelő neve:

Címe:

ACOMP Kft.
JANUÁRI
60 Ft-os
vásárlási utalványa

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

Érvényes: 1993. január 31-ig.

Egy személy részére egyszeri vásárláshoz egy utalvány használható fel!

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

1083 Budapest, Szigony u. 9.	Tel.: 134-3153
3525 Miskolc, Fazekas u. 1-3.	Tel.: 46/321-488
5600 Békéscsaba, Bartók B. u. 37.	Tel.: 66/327-195
6724 Szeged, Csongrádi sugárút 76.	Tel.: 62/313-377
8000 Székesfehérvár, Rádió u. 15/A.	Tel.: 22/319-765
9700 Szombathely, Szalonak u. 31.	Tel.: 94/14-519

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

NOVOTRADE
SZERVÍZ Kft.

MAKROVILÁG
utazási iroda

Beváltható utazási 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 FÉK C-16, C-116, +4 potméteres sebességválogatás 0%-tól 100%-ig fokozatmentesen	5000 Ft
TTL IC-teszter (Cartridge-lemezen a program)	2000 Ft
+4, C-16, C-116 UNI-ROM modul különféle kiépítésekben:	4300 Ft
— 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
2x64 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 beíratásával 2716-tól 27512-ig az eprommal együtt egységiesen	700 Ft
Árainkat az alkatrészárak változásai befolyásolhatják.	
A fenti bővítések megrendelhetők levélben az O.C.E. címen, valamint személyesen a havonta rendezendő klubdélelőttön, ahol rendszeres bemutatót is tartunk.	

Postázás esetén 100 Ft postaköltség kerül felszámításra.

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

JANUÁRI
60 Ft-os
vásárlási utalvány

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

Érvényes: 1993. január 31-ig.
HOBBI ELEKTRONIKA
JANUÁRI
vásárlási utalványa

Értéke:

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

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

Egy személy részére egyszeri vásárláshoz egy utalvány használható fel!

Elköltöztünk!

Értesítjük tisztelt ügyfeleinket,
hogy a

NAVOTRADE
SZERVIZ KFT.

Budapest V., Magyar utca 12—14. szám
alatti szervize elköltözött.

Új címünk:

1083 Budapest VIII.,
Szigony utca 9.
Telefon/Fax: 134-31-53



KARÁCSONYI AKCIÓ!

Amiga 500	37 490 Ft
Amiga 500+	44 900 Ft
Amiga 600	44 900 Ft
Amiga 1200	69 900 Ft
Commodore 64 II.	11 990 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!

Üzletünket ezüst- és aranyvasárnap is
nyitva tartjuk 9—14 óráig.

