



# TURBOBASIC C 16

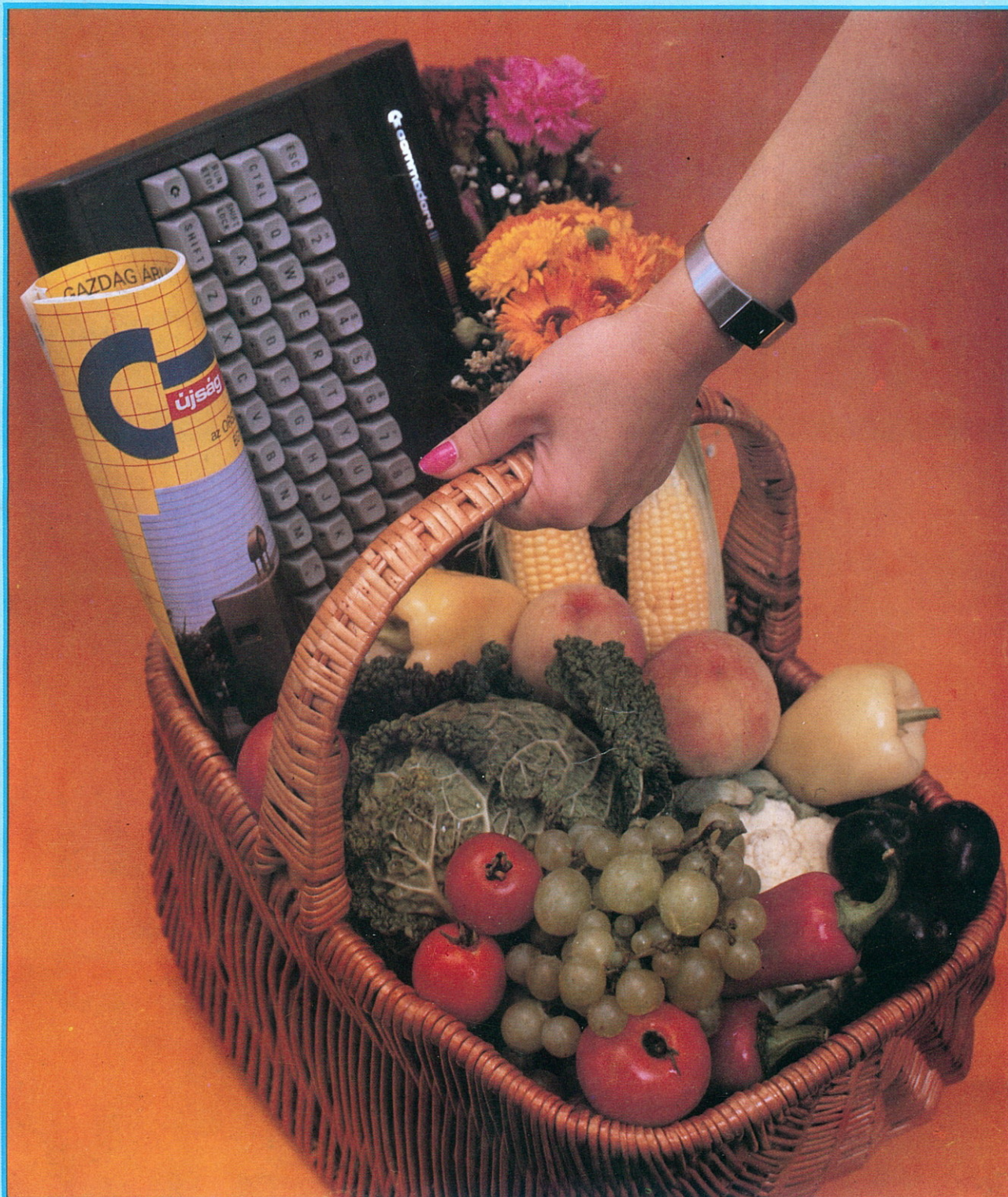
C 128 token lista

Mit tud az AMIGA?

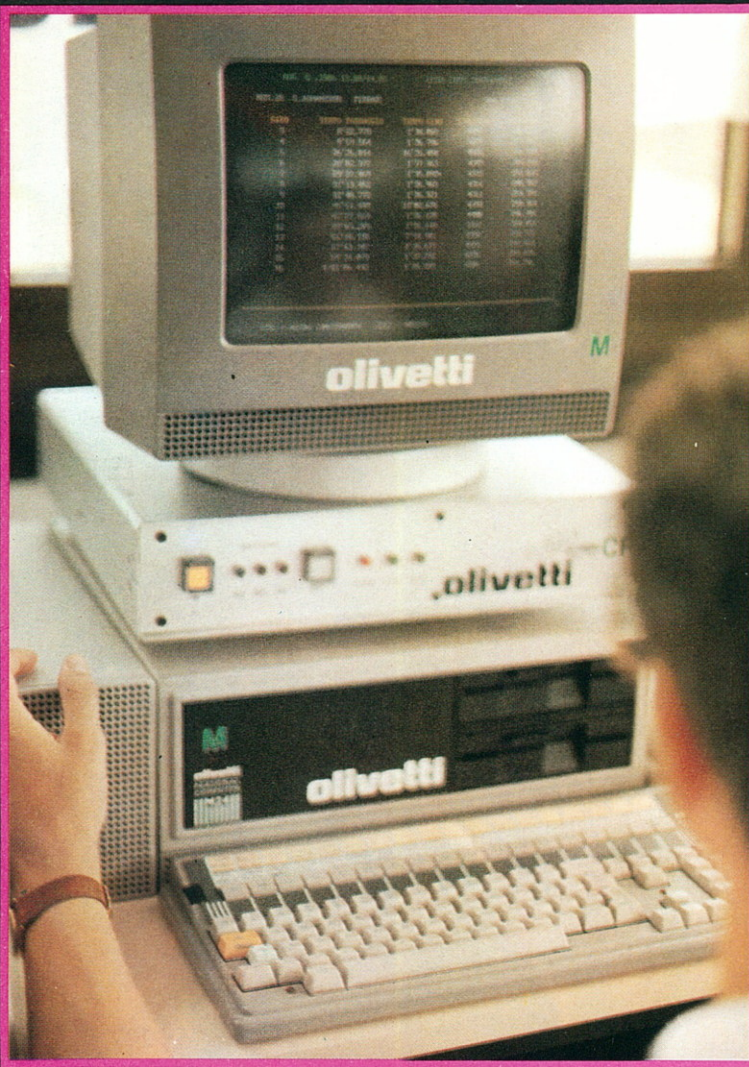
újság

1986/2

AZ  
ORSZÁGOS COMMODORE EGYESÜLET  
TAGJAINAK

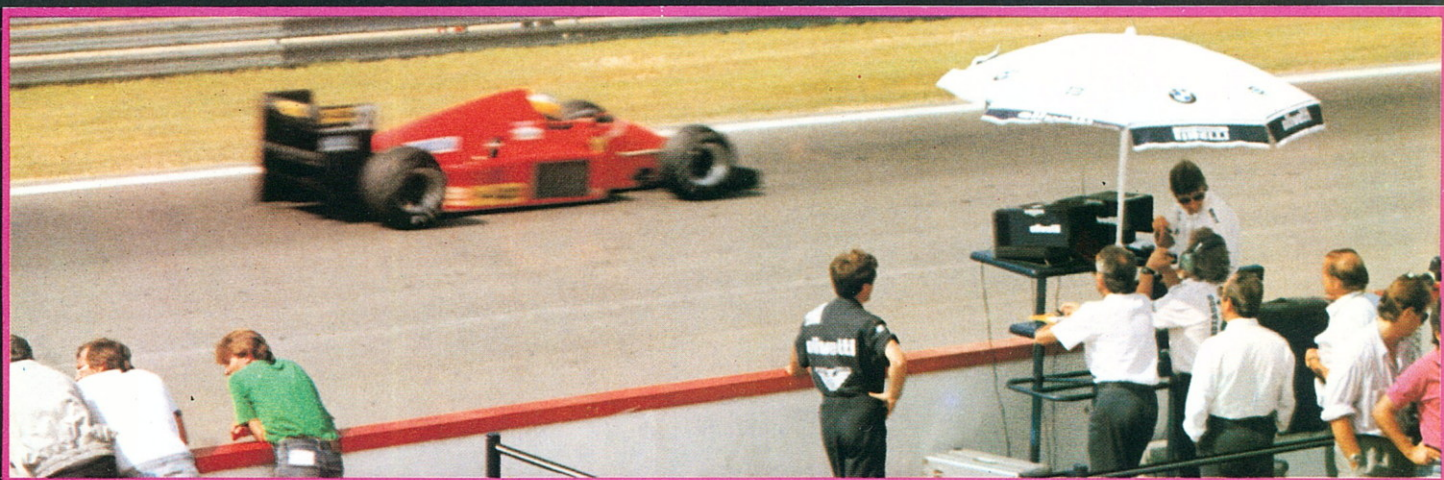






A jól sikerült magyarországi verseny lebonyolítását természetesen több számítógép, több rendszer is segítette. A nézők számára legismertebb az eredményeket nyilvántartó géplánc, amely OLIVETTI M24-es, IBM kompatibilis gépekből épült lokális hálózat. A hardverről több információnk nincsen, a szoftverről van: a COMPURACE elnevezésű programcsomag négy nyelven, olaszul, angolul, franciául és németül beszél; teljes nyilvántartást tartalmaz adatokkal a világbajnoki futamokról a versenyzőkről a gépek konstruktöreiről, és a motorokról a gumikról a versenystállókrol a versenyek anyagi támogatóiról és természetesen az aktuális versenyeredményekről.

Valamennyi adatfajtát többféleképpen lehet megközelíteni, lekérdezni. Például: az eredményeket természetesen lekérdezhethetjük versenyzőként, de megtehetjük ezt autókonstruktőrönként, motortípusonként, gumitípusonként, csapatonként, de még a szponzoronként is. Ami a rendszer leglátványosabb része, az az azonnali adatszolgáltatás verseny közben. Ezt úgy érik el, hogy minden autóban van egy kis jeladó, amelynek jelzését a rajt-cél vonalon való áthaladásakor veszi a rendszer. Ezután már a számítógépé a feladat, hogy az időt mérje, sorrendet állítson fel, stb. Nem csekélység...

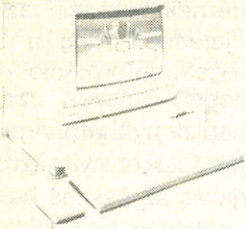




# A TARTALOMBÓL

## AMIGA LAP

4. oldal



Feltűnt a Bizományi Áruház polcán az első fecske, az első AMIGA gép, ami itthon található gazdára. Mit tud, hogyan működik a máris legendás hírcsodagép? Eladási hírek, BASIC leírás, kiegészítő hard- és szoftverek.

## PROGRAMGYORSÍTÓ

14. OLDAL

Mitől olyan lassú a gépünk, amilyen? Vagyishogy mitől olyan gyors? Hogyan működik a BASIC INTERPRETER, és hogy lehet ugyanazt a feladatot gyorsabban is megoldani?

## TURBOBASIC

17. OLDAL

Gyorsabban, egyszerűbben, jobban a C 16-ossal. Különböző funkciók egy hazai fejlesztésű BASIC-ben. Használható a Plusz 4-eshez is.

## COMMODORE 128

22. OLDAL

Sokan már hallottak róla, néhányan már használják, most bárki megismerheti Hardverleírás és programozási ötletek, valamint a 7.0-ás BASIC teljes tokenlistája cikkünkben.

## KÖNYVSZORÍTÓ

28. OLDAL

Öt alapfokú BASIC könyv bírálata rovatunkban: az eredmény nem valami fényes.

## TMK

32. OLDAL

Mitől koszos egy lemezmeghajtó? Hát a kosztól! De hogyan lehet megszabadulni tőle?

### Az Országos Commodore Egyesület módszertani kiadványa

Felelős kiadó: Horváth Judit, az egyesület elnöke  
Felelős szerkesztő: Pogány György  
Szerkesztő: Angyalosi László  
Művészeti szerkesztő: Várhelyi László  
Fotó: Gál Imre  
Szerkesztőségi titkár: Tóth Éva  
Készült a Globál GMK gondozásában  
Levélcím: Commodore újság  
Pozsonyi út 50. fsz. 4.  
1133  
Telefon: 408-603

## Kedves Tagtársaink!

Mint az egyesület elnöke úgy érzem, hogy az egyesület keretén belül sikerülhet egy olyan kollektívát kialakítani, amely a hazai számítógépes kultúra elterjesztését szívégyének tekinti. Ezen a területen a tagok külön-külön is úttörő munkát végeznek. Az egyesület keretén belül viszont lehetőség van arra, hogy szakember, kereskedő, felhasználó és újságíró összefogjon és egyesült erővel dolgozzon - ugyanazért.

Faját munkámból tudom - mint kereskedő és felhasználó - mennyire fontos a számítástechnika alkalmazása. Ezért úgy látom, hogy egyre nagyobb súlyt kell helyezni az oktatásra, a jövő szakembereinek képzésére, de nem szabad megfeledkeznünk arról sem, hogy az országban jelenleg a mezőgazdaság és ipar számos területén is használják a Commodore kis számítógépeket. Egyesületünkben lehetőség van egy olyan fórum megteremtésére, ahol a különböző szakterületen dolgozók együttesen léphetnek fel, közösen képviselhetik a közös ügyet. A Commodore újság elsősorban abban segíthet, hogy információinkat kicseréljük. Természetesen egyesületünk nyitott az új ötletek befogadására és lehetőségeinkhez mérten, minél rövidebb időn belül megpróbáljuk azok hasznosítását elősegíteni.

Horváth Judit

az Országos Commodore Egyesület elnöke

Mint hogy lapunk olyanok kezébe is kerül, akik még nem tagjai az egyesületnek, ismét leírjuk azt, hogy melyik páholyban mit nyújt az egyesület.

## COMMODORE DEÁKPÁHOLY

- negyedévenként információs bulletin friss Commodore hírekkel, új, itthon is kapható programok és könyvek listájával
  - pötyögő szolgálat, amely megkíméli a tagokat a pötyögéstől
  - ingyen apróhirdetési lehetőség a Commodore újságban
  - Különböző vásárlási kedvezmények
- TAGSÁGI DÍJ: évente 116,- forint  
félévre 60,- forint

## COMMODORE PLUSZPÁHOLY

- minden az előbbieken felsorolt szolgáltatás
  - a havonta megjelenő 36 oldalas Commodore újság, amely az angol Commodore Computing International és a nyugatnémet Data Welt lapokkal együttműködésben készül
  - vásárlási és szervizkedvezmények, egyéb pénzkímélő akciók
- TAGSÁGI DÍJ: évente 1264,- forint  
félévre 650,- forint

## COMMODORE SZUPERPÁHOLY

- minden az előzőekben felsorolt szolgáltatás
  - havonta 10+4 példány a Commodore újságból
  - nagyobb mértékű kedvezmények
- TAGSÁGI DÍJ: évente 12.128,- forint  
félévre 6100,- forint

Akik a félév folyamán fizetik be a tagdíjat, visszamenőleg megkapják a Commodore újságot és természetesen azokat a kedvezményeket is igénybe vehetik, amelyek még nem jártak le. Ha valamelyik egyesületi tagunk kíván magasabb páholyba lépni, úgy csak a különbözetet kell befizetni az egyesület számlájára.

### HOGYAN LEHET ÖN IS EGYESÜLETÜNK TAGJA?

Befizeti az Ön által kiválasztott páholy tagdíját a Budapest, XIII. Visegrádi u. 7/b. alatti OTP fiókban vezetett MNB 217-98292, OTP 565-3610 számú számlára. Ezt megteheti személyesen vagy postán. Egy hónapon belül megkapja tagsági igazolványát, és az első Commodore újságot.



# BASIC AMIGA™

**A személyi számítógépeken a programozók napjainkban már biztonságosan dolgoznak a Microsoft-BASIC különböző változataival. Az Amiga nyelve is a Microsoft-BASIC egyik változata így nem okoz gondot az elsajátítása. Az MS-BASIC változatok hasonlóságából eredően a programok közvetlen áthelyezése az egyik számítógépről a másikra is viszonylag egyszerűen megoldható. Az Amiga BASIC-jének néhány jellegzetes, más gépektől eltérő tulajdonsága:**

## ALAPINFORMÁCIÓK

A program sorait nem kell sorszámozni. Az esetleges vezérlésátadási – ugrási – helyek megjelölése ún. címkékkel, olvasható nevekkel történik. Egy eljárás, szubrutint tehát el kell nevezni, és a rutin hívásakor nem sorszámot, hanem a nevet kell megadni. A program struktúrájának szervezésére az IF-THEN-ELSE-ENDIF feltételes vezérlésátadó, a FOR-NEXT, WHILE-WEND ciklusszervező és a SUB-SUBEND szubrutinkezelő utasítások használhatók fel. A változók, címkék és alprogramok azonosító nevei max. 40 karakterből állhatnak. Az Amiga az adattípusok gazdag választékát kínálja a felhasználónak. Az egész számokra két pontossági fokozatot ajánl, a 16 és a 32 biteset, a lebegőpontos számok esetében a 32 és a 64 biteset. A stringek, ill. szöveges részek max. 32768 karakterből állhatnak. A tömbök max. 255 dimenziósak lehetnek és 32768 elemet tartalmazhatnak.

A szubrutin mindig egy ún. fejléccel kezdődik, amely a rutin nevét és paramétereit rögzíti. Az alprogram hívásakor a paraméterek értékadása is lezajlik. A rutin változóinak a sikeres futtatás érdekében nem szabad előfordulnia a főprogramban.

Az Amiga BASIC-jéből hiányzó függvények egy szubrutinkönyvtárból hívhatók be. A futtatás során az operációs rendszer a behívott függvényeket beépíti a BASIC programba. A LIBRARY utasítással újabb függvények definiálhatók. A függvényrutinok gépi és C-nyelven írhatók meg és a könyvtárba tehetők. Az egyes függvények a nevük alapján azonosíthatók. A „Con-

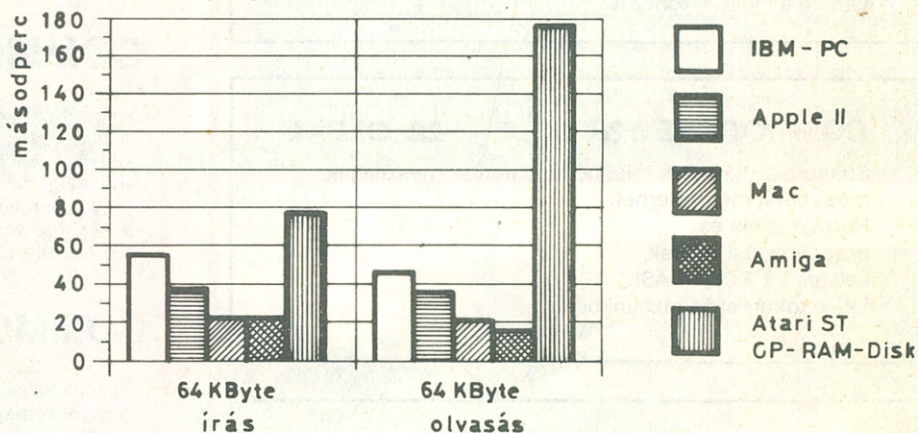
vertFd” segédprogrammal a teljes könyvtár a diszkre menthető.

## ABLAKOK

Az Amiga rendszerint két képernyőablakkal dolgozik. Az ún. output-ablakon történik a futó program adatainak

be- és kivitele. A felhasználó ezen az ablakon keresztül ad parancsokat a BASIC interpreternek. A másik, az ún. listing-ablakon jelennek meg a futó program szöveges információi. Az Amiga BASIC-jében tetszés szerint több kiegészítő ablak nyitható. A program amikor saját ablakot nyit, az output-ablakot zárja és az összes be- és kivitelt a saját ablakján bonyolítja le. A két elsődleges ablak sem mindig látszik, hibátlan programfutás esetén pl. a listing-ablak zárt.

A program szerkesztése a listing-ablakon kényelmesen végezhető. A teljes listán belül a kurzor botkormánnyal vagy kurzor-billentyűvel tetszőlegesen elmozdítható. A szerkesztést egy EDIT menü is támogatja, melynek leggyakoribb parancsai a CUT (kivágás, ill. törlés), a COPY (másolás) és a PASTE (beszúrás).



## A COMMODORE ELADTA AZ ELSŐ EZER AMIGÁT

A Commodore Business Machines Ltd. (UK) közölte, hogy az első 1000 egységből álló Commodore Amiga szállítmányt, amelyet 1986. május 19.-én indítottak útra, már eladták, és naponta érkeznek megrendelések.

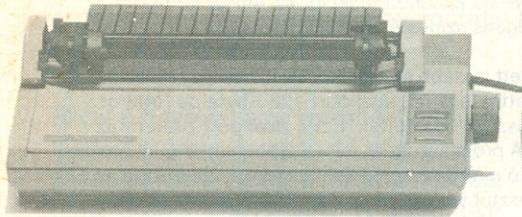
Ez már a Commodore Amiga nagysikerű indulásakor érezhető volt a 7. hivatalos Commodore Show-n, ahol rendkívül nagy érdeklődést mutattak mind a viszonteladók, mind a közvetlen felhasználók.

Az Amiga kétféle konstrukcióban áll rendelkezésre. A Commodore Amiga System I. 512 Kbyte RAM-mal és egy 800 Kbyte kapacitású 3 1/2 collos lemezegységgel rendelkezik, az ára pedig 1475 £ + VAT (adó). A Commodore Amiga System II-höz még egy külső 3 1/2 collos drive is tartozik, az ára 1675 £ + VAT. Mindkét konfiguráció tartalmazza a klaviatúrát, az egeret, egy színes monitort és az Amigát ismertető programokat.



## HIBAKEZELÉS

A futtatás során előforduló hibára az interpreter leállítja a programot, megfelelő hibajelzést küld a monitorra és a listában bekeretezi a hiba keletkezési helyét. A hiba előfordulása nyitja a listing-ablakot, amelyen a szükséges javítás azonnal elvégezhető. A program



ellenőrzése és az esetleges hibák keresése a RUN menü alatt történik. A START indítja a futást. A részletes ellenőrzés a TRACE ON és a TRACE OFF utasításokkal szervezhető meg. A TRACE ON utasítás kiadása után az Amiga narancssárga keretben kijelzi, hogy a program futása éppen mely sorban tart. A futás figyelése a TRACE OFF utasítással szüntethető meg. A program lépésről-lépésre a STEP utasítással ellenőrizhető, ilyenkor minden egyes utasítás végrehajtása után automatikusan leállítja a futást, mintha STOP utasítást kapna.

## GRAFIKA

Az Amiga grafikus utasításkészlete a felhasználónak számos rajzoldási lehetőséget kínál. A LINE utasítással a monitor két pontja között tetszőleges színű egyenes húzható. Az egyenes-húzások megfelelő stratégia szerinti váltogatásával derékszögű négyszögek rajzolhatók, amelyek egy választott

színnel befesthetők. A CIRCLE utasítással körök, körívek és elipszisek rajzolhatók. Az összetett alakzatok, mint pl. sokszögek az AREA utasítással pontról-pontra definiálhatók és az AREA FILL utasítással összeköthetők. A komplexebb alakzatok elemekből állíthatók össze. Az egyes területek beszínezése a PAINT utasítással történik.

A grafikánál alkalmazott színek a PALETTE utasítással válogathatók. A PALETTE utasítás a video-chip RGB (piros, zöld, kék) színregisztereit tölti fel az új értékekkel. Amennyiben négy szín nem elegendő, a SCREEN utasítással több szín generálható a monitoron.

Normál módban az Amiga a felhasználónak 640x320 pontos felbontást és négy színt kínál. Több szín megjelenítése csak a felbontás csökkentésével érhető el. Az alkalmazott trükk egy virtuális képernyő létrehozása a

SCREEN utasítással, amely a SCREEN CLOSE utasítással ismét eltávolítható.

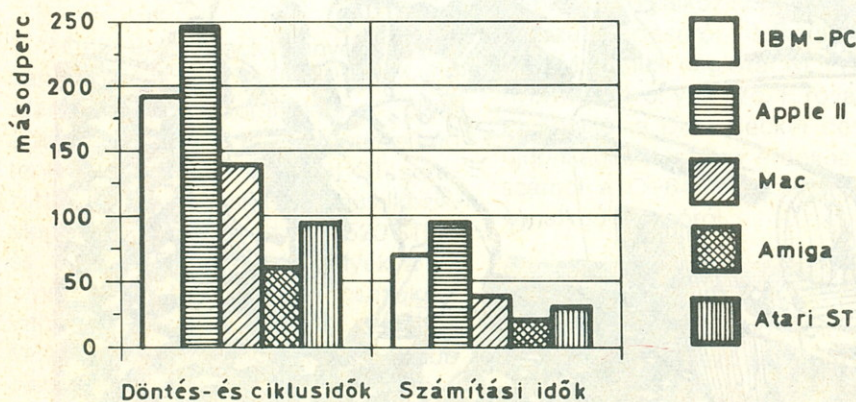
## MOZGÁS

A grafikai lehetőségeket tovább bővíti, hogy a monitoron mozgatható objektumok jeleníthetők meg. Két változat ismeretes: a hardver sprite-jai gyorsan mozgathatók, de különböző korlátozások alá esnek, míg a szoftver sprite-jai, BOB-jai (Blitter Objects) lassúbbak, de lényegesen sokoldalúbb a felhasználhatóságuk. Az Amiga mindkét

típust ugyanazon BASIC utasításokkal kezeli. Az újabb objektumok létrehozását a diszkról behívható „Objedit” nevű program segíti. A sprite-ok az OBJECT SHAPE utasítással definiálhatók. Az objektum pozícióját a képernyőn az OBJECT.X és az OBJECT.Y, mozgási sebességét az OBJECT.VX és az OBJECT.VY rögzíti. Az objektum gyorsuló mozgást is végezhet. Az időben automatikusan változó sebességet az OBJECT.AX és az OBJECT.AY mutatja. Két objektum ütközése a COLLISION utasítással definiálható. Az OBJECT.PRIORITY utasítással meghatározható, hogy egymás mögötti elhaladáskor melyik tárgy takarja a másikat. A háromdimenziós hatás így egyszerűen BASIC-ben megvalósítható. Az Amiga menükészlete a MENU parancs segítségével tovább bővíthető. Grafikai menüpontok nem realizálhatók.

## HANG

Az Amiga hang és zenei képességei BASIC-ben is kihasználhatók. A négy hangcsatorna valamelyikén a SOUND utasítással megadható egy hang frekvenciája, időtartama és erőssége. A jellegzetesebb hullámformák a WAVE utasítással állíthatók be. A beszéd szintetizálása a TRANSLATE és a SAY utasításokkal történik. A TRANSLATE utasítás az emberi beszédet – jelenleg még csak angol nyelven – gépi jelkézzé kódolja át. A SAY utasítás pedig hangokat állít elő amely felis-





amely felismerhetően tükrözi az előzőleg letárolt szöveget.

## GYORSASÁG

Az Amiga adatátviteli sebességét tesztelték egy be- és kiviteli rutinnal. A mért idők a 2. ábrán láthatók, összehasonlítva más ismertebb gépekkel. Az ábrából megállapítható, hogy az Amiga BASIC-je előkelő helyet foglal el gyorsaságával a többi típushoz képest. A programfutási időket is tesztelték. A 2. ábra baloldali rovata a döntés-(IF-THEN) és a ciklusidőket mutatja, míg a jobboldali a számítási időket veti össze mintegy 5000 db. összeadás, kivonás, szorzás és osztás esetében. Ezen teszteredmények alapján is jobbnak bizonyult az Amiga.

Az Amiga 512 KByte-os tárterülettel dolgozik, amelyből bekapcsoláskor 216 KByte szabad kapacitás jelentkezik be. A nagy tárhelytelenség teszi a második diszkmeghajtó használatát.

Computer Persönlich, 1986. V. 14. cikkéből fordította:

Szloboda Gyula

## Oldalkocsival a siker felé

Nincs több szoftver problémája az Amiganak: az „oldalkocsival” a Commodore szupersztárja is teljesen IBM PC kompatibilis. A Nyugatnémetországban kifejlesztett és hamarosan az USA-ba is exportált „Sidecar”-ral, melyet „oldalkocsinak” fordíthatnánk, az Amiga teljesen IBM PC kompatibilis lett.

Bizonyára az első pillanatban hihetetlennek tűnik, hogy a Motorola 68000 processzor IBM kompatibilis lehet, hiszen az IBM PC-kbe az Intel 8088 vagy 8086 mikroprocesszorokat építették be. Technikailag a megoldás rendkívül egyszerű. Az oldalkocsiba – nevezzük így – egy teljes IBM PC központi egységet építettek be 8088 (4.77 MHz)-es processzorral, 256 Kbyte RAM-mal, timerrel, floppy-, interrupt-, és DMA vezérlővel. Egy 360 Kbyte-os 5.25"-os floppy meghajtó is helyet kapott az oldalkocsiban. Minden be- és kimeneti funkciót az Amiga lát el, így a Commodore sztár egy szuperintelligens színes és hangos terminálja lett az IBM PC-nek.

De az Amiga semmi esetre sem lett lassabb az oldalkocsival, mivel a fejlesztők a két gép kapcsolatához kifejlesztettek egy ún. dual-port 128 Kbyte-os RAM-ot. Ezt az Amiga bővítő csatlakozó buszára illesztették. Ezt a dual-port RAM-ot az oldalkocsi processzora és az AMIGA processzora is, mint központi memóriát érik el. A párhuzamos és soros nyomtató interface sem jelent akadályt, mivel ez is az Amiga szabvány csatlakozóin keresztül valósul meg. Az oldalkocsiban van egy külön tok fenntartva az Intel 8087 lebegőpontos aritmetikai processzor számára, így a rendszer hihetetlen számítási sebességgel és pontossággal rendelkezik.

Az IBM PC sikere egyrészt a hatalmas szoftver kínálaton, másrészt a merevlemez illeszthetőségen alapszik. Az oldalkocsiban három bővítő kártya hely található. Ez lehet számítógép hálózati, modem vagy merevlemez illesztő kártya. A merevlemez nem csak az Intel processzor érheti el, hanem az Amiga processzora is, mégpedig sebességvesztés nélkül. Az IBM PC minden video módját támogatja az Amiga szoftver, a monochrom szöveg és a színes grafikus módok is.

Nagy elismerésre méltó az Amiga oldalkocsin keresztüli bővítési lehetősége. A központi memória kapacitása elérheti a két Mbyte-ot is. A 3.5"-os floppyk nagy sikert arattak a felhasználók között és az oldalkocsi ezt a lemezformátumot is tudja kezelni.

E kiegészítő berendezés mintapéldánya a fejlesztőknél tekinthető meg, a kereskedelemben ez év októberében mutatkozik be. A berendezéshez mellékelnek egy MS-DOS lemezt is különböző szervizprogramokkal.

IF (4\*0,5\*PILSNI VILAGOS 13% + 2\*VODKA 42% >= 1.32%) AND RENDOR THEN GOTO GYALOG





# DATA WELT

Das aktuelle Computermagazin

6/86 Juni

A Data Becker cég kiadásában megjelenő DATA WELT folyóirat 1984-ben indult útjára. A lap kezdetben csak negyedévenként jelent meg, majd 1985 elejétől kéthavonta, 1985 novemberétől pedig havonta nyújt betekintést a Data Becker cég kiadványaiba, mutatja be programtermékeit és tudósít a mikro- és PC világ érdekességeiről.

Mivel e folyóirat egy számítástechnikai kiadó és programfejlesztő cég gondozásában jelenik meg, ezért nem csoda, hogy kissé részrehajló. A számítástechnika világából csak azt és úgy mutatja be olvasóinak, amelyet saját szemüvegén nézve arra érdemesnek tart. Így a számítógéptípusok széles skálájából csak a Commodore 64, 128; Atari XL, ST sorozat; Schneider CPC sorozat gépeire közöl hardver és szoftver ötleteket, érdekességeket. Néhány cikkben hírt ad még más gyártók típusairól is, de több figyelmet nem szentel rájuk.

A rovatok a lapindítás után a mai napig átrendeződtek, kibővültek. A rovatok cikkei az újságban nem egy helyre összegyűjtve kerülnek az olvasó kezébe, hanem egyenletesen elszórva a lapban. A tartalomjegyzék azonban jól összefogja az egy rovatba gyűjtött hírtanyagot. A folyóirat állandó rovatain kívül olvasói tanácsadást és természetesen a következő szám anyagának előrejelzését találhatja az olvasó az újságban.

Mit tartalmaznak az állandó rovatok? A válaszadáshoz legjobb lesz, ha találmokra kiválasztunk egy számot, legyen ez az idei 4. szám, amely áprilisban jelent meg.

## NEWS & TRENDS

A "Hírek, újdonságok, irányok" rovatban mindig a legfrissebb információk szerepelnek. A „Commodorus” című állandó cikksorozat a Commodore cég terveit tárja elénk. Az Atari ST sorozattal három tudósítás is foglalkozik. Megismerkedhetünk az Atari 520 STFM és 1040 STF gépekkel, melyek a jól ismert gépek televízióhoz is csatlakoztatható változatai. A SONY cég egy MSX II operációs rendszerű gépet dob piacra, mely egy 256 KByte-os gépet

takar. A gép különlegessége a 128 KByte-os video RAM és a beépített 3.5"-os floppy 720 KByte kapacitással. Egy AMIGA-ról szóló hírből arról értesülünk, hogy az 512 KByte RAM-ot tartalmazó gép ára 5900.- DM. A következő hír a Commodore PC 20 gyógyászati felhasználásáról tudósít, ebből megtudhatjuk azt, hogy így 1500 hallókészüléket használó beteg adatait tartják nyilván.

## THEMA

A „Téma” rovatban mindig egy-egy területről részletesebben írnak, pl. Winchester lemezt fejlesztettek ki a Schneider CPC-k számára. A rovat egyik átfogó cikke bőséges ismertetést ad a keménylemezekről és a magafloppykról. Egy másik tudósítás a C-64 SF1001 típusú floppy meghajtóról szól.

## PC-PRAXIS

A „PC-Gyakorlat” rovat legérdekesebb cikke az új Apricot Fle gépet mutatja be az olvasóknak, és mindig a nagyobb személyi számítógép kategóriába tartozó gépekkel foglalkozik.

## DRUM & DRAM

Az „Erről, arról” rovat általában hardver témákkal foglalkozik. Egy új nyomtató sorozatgyártásáról értesülhetünk, a Star Micronics cég Star NL-10 mátrixnyomtatójáról van szó. E rovat értesíti az olvasót a Data Becker cég könyv-újdonságairól is. Egy érdekes cikk beszámol a C-64-hez csatlakoztatható vonalkód olvasóról.

## SOFTWARE

Ez a rovat a legújabb Atari ST-re írt áramköri lap tervező program teszteléséről számol be. A Schneider CPC

kedvelőknek két cikk is szól: egy diszk szerviz és egy táblázatos számítógépprogramról. A C-64 rajongók sem maradnak információ nélkül: „Igazi szerszámok” címmel érdekes beszámolót olvashatnak a BASIC-ből hívható rutin-gyűjteményről, a gépi kódú programozást segítő új monitorról, reassembler-ről, 6510 szimulátorról és egy makro-assemblerről.

## MAGAZIN

A „Tippek és trükkök” rovat a Data Becker cég Superbase 64 nevű C-64-re írt adatbáziskezelő programjának egyik alkalmazási lehetőségét, az autózási költség-számítást mutatja be. Egy C-64-re írt C programnyelvről szóló sorozat második része is e rovatban kapott helyet. Itt olvashatunk még a 68000 mikroprocesszorról, Turbo-Pascalról, az MS-DOS BASIC-ről, valamint az Atari ST új BASIC-jéről a GEMBAS-ról. Itt kapott helyt a népszerű QuickTips programlista sorozat, ahol Commodore, Schneider és Atari programokat talál a pötyögésre elszánt olvasó.

## TIPS & TRICKS

Végül a „Magazin” rovatban megtaláljuk a Data Welt játéktippjének a „Hitchhiker's Guide to Galaxy” ismertetését, egy brémai computerklub bemutatását és természetesen az elmaradhatatlan beküldendő rejtvényt.

A bemutatott számból is kiderül, hogy annak érdemes olvasnia rendszeresen a lapot, aki folyamatos betekintést akar kapni a Data Becker cég újdonságaiba és emellett a számítástechnikai újdonságok is érdeklik. De az, aki már elkötelezte magát egy adott cég adott géptípusának, az inkább szorosabb kapcsolatot épít ki a cég hivatalos hírforrásaival, és a piac általános alakulásáról egy független, szélesebb skálát bemutató folyóiratot olvas.

**Így én mint a Commodore egyesület tagja szívesen átvinnék néhány cikket vagy érdekes program listát a Commodore újság számára, de kizárólag a Data Weltre nem támaszkodnék. Ezt a folyóiratot legfeljebb rendszeresen olvasnám egy könyvtár csendes olvasótermében.**

Szabados Pál



Most induló rovatunk elnevezése kissé megtévesztő, de éppen arra szeretnénk felhívni a figyelmet, hogy a számítógép meglehetősen bonyolult rendszer.

„Csak úgy fusiban” nem illik benne turkálni, forrasztgatni. A FUSI rovatban tehát olyan hardver-átalakítások tervrajzát, szerelési leírását fogjuk közölni, amelyek nagy része otthon, „fusiban” elvégezhető, de csak profi színvonalon.

Ebben a számban néhány jótanácsot szeretnénk adni azoknak, akik már fogtak a kezükben forrasztópákát, de az integrált áramkörös technikával még nem nagyon foglalkoztak. Aki az itt közölt útmutatót betartja, az valószínűleg nem fogja tönkretenni saját számítógépét, de a teljesen kezdők részére semmi biztosat nem állíthatunk. Következő számunktól részletes rajzokkal közöljük a felsorolt hardverkiegészítések elkészítési módját:

- beépíthető C 16-os memóriabővítő 64 KByte
- C 16, C 64 cartridge - hardver leírás
- C 16, C 64 cartridge - szoftver
- rádugható C 16 memóriabővítő 64 KByte
- EPROM égető

Mielőtt szétszerelnénk számítógépünket, ismernünk kell néhány – a számítógép szempontjából létfontosságú – előírást, szabályt. Cikksorozatunk első részében ezekre szeretnénk felhívni a figyelmet.

#### SZERELÉSEK

A Commodore homecomputerek szétszedésekor, belső részükben való mérésekor mindig gondolnunk kell a gép nagybonyolultságú IC-inek (TED, SID, VIC, MPU stb.) CMOS "lelkivilágára". A következő szempontokat tartsuk be:

1. A CMOS IC-k – a napjainkban szinte minden tokba beleintegrált védelem ellenére is – rosszul viselik el a lábakra jutó sztatikus töltést. Szerencsétlen esetben ez az IC kiégését okozhatja. A védekezés módja elméletileg egyszerű, nem engedjük a rajtunk összegyűlt töltést az IC-n keresztül kisülni, vagy ami még jobb, elkerüljük az ilyen feltöltődést. Gyakorlatban az ilyen töltések pl. műszálas ruha viselésekor, műanyagpadlón való sétáláskor, műanyagszéken (!) való mozgolódáskor alakulhatnak ki, célszerű ezeket kerülni. A veszély elsősorban akkor nagy, amikor az IC-t megfogjuk és behelyezni próbáljuk az áramkörbe, vagy ellenkezőleg: épp kiemeljük belőle. Nagyon jó

védelmet jelent, ha ilyen esetekben vizes kézzel nyúlunk a tokhoz, ekkor kezünk vezető lévén nem alakul ki rajta veszélyes töltés. Szintén jó hatásfokú, sajnos elég kevés esetben használható az ezüsthóliába való csomagolás, vezető szivacsba való dugás is. Ezeket inkább a biztonságos tárolásra használjuk.

2. A sztatikus töltésen kívül a zárlatok is veszélyesek az IC-k életére. Ezek többféleképpen lehetnek:

- a/ Földzárlat, egymáshoz képest való zárlat – talán ezek okoznak legkisebb mértékben maradandó elváltozást.
- b/ Tápszárlat – az esetek nagy részében ezt rosszul viselik a CMOS IC-k kimenő lábai. Villámgyors melegekedést, végső esetben azonnali kiégést okozhatnak, általában a főszereplő IC-re nézve.
- c/ Tápnál nagyobb feszültség, esetleg hálózat vagy törpefeszültségű váltóáram felé való zárlat – ezekben az esetekben szinte biztos az IC kiégése, sajnos számíthatunk más, közvetlenül a zárlati pontra nem kapcsolódó tok tönkremenetelére is. Ez utóbbi esetben nagyon könnyen teljesen ki lehet égetni egy gépet, ettől nagyon tartózkodjunk! A zárlatok ellen védekezni alapvetően a következő módon lehet:

1. Jó pákával, lehetőleg nagy gyakorlat-

tal forrasszunk. Erre az áramkörök finom rajzolata miatt egyébként is szükség van.

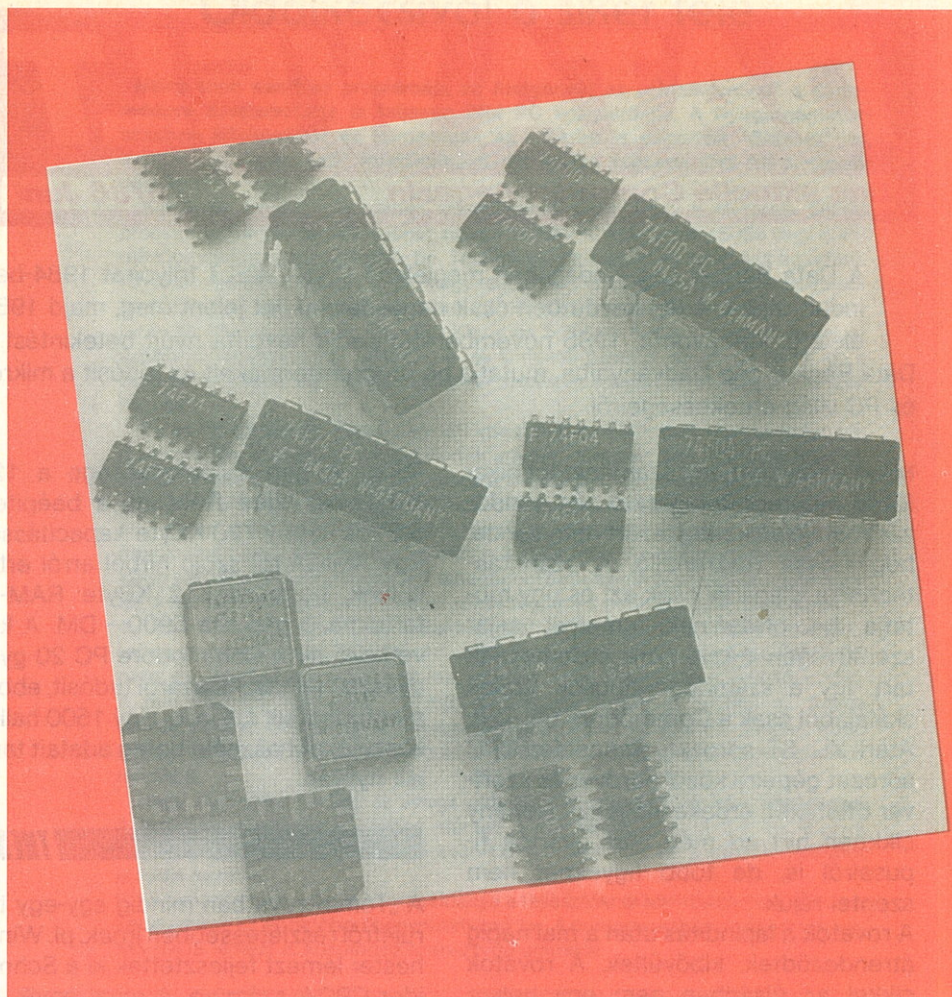
2. Mindig, minden változtatás után vegyük szemügyre az egész áramkört, különösen azt a részletet, amin épp változtattunk (kicseréltünk egy ellenállást, stb.). A tapasztalat azt mutatja, az esetek nagy részében itt követünk el hibát.

3. Ha bekapcsoltuk az áramkört (számítógépet) és nem működik helyesen, pl. nem jelentkezik be a tévén, rögtön kapcsoljuk ki és GOTO 2. így elkerülhető a hosszú ideig zárlatra kötés romboló hatása.

A CMOS IC-ken kívül a számítógépekben található többi integrált áramkör kevésbé érzékeny. Kiemelhetjük még a memória IC-eket; a dinamikus RAM-ok, valamint a ROM-ok a tapasztalat szerint ritkábban romlanak el, a többi TTL IS IC-k is szinte mindig jól elviselik az áramkörben való szerelést, mérést.

#### A TECHNOLÓGIA

A számítógépek szerelésénél jóval nagyobb gondot kell fordítani az egyéb áramkörökkel szemben a szerelés technológiájára. A finomabb,

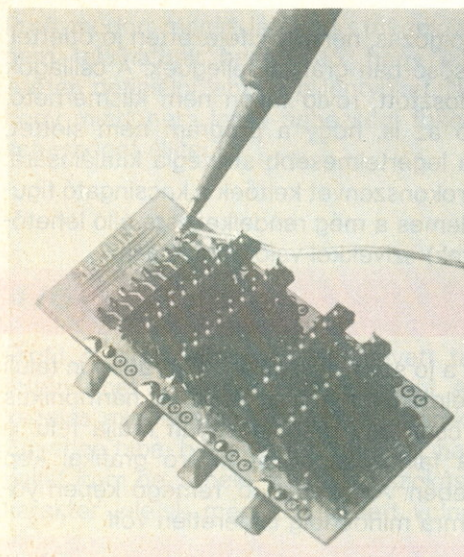




kétoldalas NYÁK lemez miatt a forrasztási pontok, fóliák egész közel vannak egymáshoz, növekszik a zárlatveszély. Másrészt csökken a forrsemek hőterhelhetősége, tehát rövidebb idő alatt kell forrasztanunk ahhoz, hogy elkerüljük a rézfólia leválását. Az alkatrészek közel vannak egymáshoz, egy kis elhajlás már problémát okozhat. Előfordul, hogy otthoni körülmények között nem megszokott technológiát kell igénybe vennünk (Pl. IC kiforrasztás IC és NYÁK sérülés nélkül), ezekre a cikkekben külön útmutatást adunk.

## MÉRÉS

Az áramkörök mérése hasonló, mint egyéb áramkörök vizsgálata, illetve



azokénál könnyebb. A kis áramokkal működő CMOS IC-k megkövetelik a nagy bemenő ellenállású műszerek használatát. A méréseknél is be kell tartanunk az I. pontban leírtakat. A mérendő feszültség szintek döntően 0 és 5 V, frekvenciák DC és 10 MHz közé esnek, mindezekből látható, hogy egy jó minőségű oszcilloszkóp szükséges munkánkhoz. A tapasztalat szerint magyar vagy nyugati oszcilloszkóppal nincs probléma, különösen ha figyelünk arra, hogy a földelést minden mérés előtt rákössük a számítógép földjére.

Szovjet gyártmányú oszcilloszkópok – a hálózattól való nem tökéletes leválasztás miatt – CMOS IC-k kiegészését okozhatják. Kerüljük el használatukat. Fontos a mérőhegy megválasztása is, vékonyat használjunk. Oszcilloszkóppal való méréskor lehet a legkönnyebben összezárni egy IC szomszédos lábait.

Egyszerűbb méréseknél jól használhatunk digitális multimétereket. Ezek rendelkeznek azzal az előnnyel, hogy a szükséges nagy bemenőellenállás eleve biztosított. Segítségükkel pl. az

áramkörökben előforduló zárlatok bekapcsolás előtt megtalálhatók.

## ANYAGOK, SZERSZÁMOK

Hardvereink építéséhez felhasznált anyagok négy csoportra oszthatóak:

- nyomtatott áramköri lemezek,
- Passzív elektromos alkatrészek (ellenállások, kondenzátorok, tekercsek...)
- aktív elektromos alkatrészek (diódák, tranzisztorok, IC-k...)
- szerelékanyagok (foglatok, kapcsolók, dobozok...)

## NYOMTATOTT ÁRAMKÖRI LEMEZEK

Kétoldalas, üvegszálaz, minél jobb minőségűt használjunk. Az áramkörök bonyolultsága folytán a rajzolat az esetek többségében finom, tehát furatgalvánozás szükséges. Mivel ez utóbbit házilag nem lehet elkészíteni, helyette a következők közül választhatunk:

- vezetékiszálakkal pótoljuk a furatfémvezetést,
- valahol elkészítetjük a nyomtatott áramkört.

Mindenképpen tartsuk be a következőt: egy megbízhatatlan, rossz minőségű NYÁK lemezzel nagyon sok gondot tudunk magunknak okozni. Az esetek többségében az azonnal „bemutatózó” hibákon felül számos csak huzamos működés után lép fel, és pl. 100 %-ra feljavítani egy hibás furatgalvánozású lemezt nagyon nehéz feladat.

## PASSZÍV ELEKTROMOS ALKATRÉSZEK

Áramköreinkben kisteljesítményű, kis méretű alkatrészeket használunk. Az ellenállások 0,125 W-osak, az elektrolitkondenzátorok 6–10 V-osak, a nem elektrolitkondenzátorok 20–40 V-osak lehetnek. Beszerzésüknél alapvető szempont legyen a hosszúidejű stabilitás, ha esetleg az ár magasabb is.

## AKTÍV ELEKTROMOS ALKATRÉSZEK

Tranzisztorok, diódák terén teljes biztonsággal használhatunk keleti, nyugati alkatrészeket egyaránt. Ha nem kapunk megfelelő típust, mindig „felülről” közelítsük meg a kérdést, (nagyobb áram terhelhetőség, teljesítmény stb.) Integrált áramköröket lehetőség szerint nyugati típusút szerezzünk, a kérdéses helyekre nem árt tokot rakni. Az IC-k vásárlásánál – mivel számos

gyártóval találkozhatunk – mindig tájékozódjunk a specifikációk felől. (Pl. nagyon sokféle 64 Kx1 bit-es dinamikus RAM létezik, soknak a lábkiosztása is eltér.)

## SZERELÉKANYAGOK

Itt is számos gyártó értékesíti termékét. Sajnos nem biztos, hogy az a kapcsoló, vagy foglat, amire tervezzük áramköreinket, egy hét múlva is kapható lesz. Érdemes már a tervezést megelőzően beszerezni a „kényesebb” anyagokat, így nem érhet minket kínos meglepetés. A beszerzésnél a már említett szempont legyen a fő: a drágább, de hosszabb élettartamú termék jobban megéri olcsóbb társánál. Egyes kiegészítőknél a kereskedelemben nem előforduló alkatrészeket is használnunk kell, például műanyag házakat, amibe az áramköreinket helyezük.

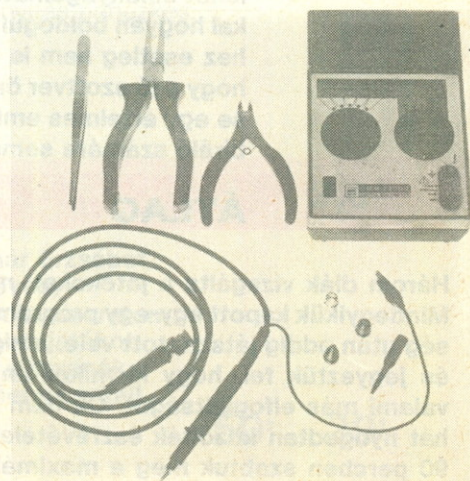
## SZERSZÁMOK

A teljesség igénye nélkül bemutatunk néhány szerszámot, melyeket munkánk során használhatunk. Érdemes külön kiemelni a forrasztópákát: csak törpefeszültségű, a hálózattól galvanikusan elválasztott típust használjunk, hegye vékony legyen. Nagyon jó a 24 V-os hőfokszabályozós Weller típus, ennek hegye speciális anyaggal és cinnel befuttatott.

Különleges szerszámokat is használunk. Az ún. Stripax blankoló fogó a vezetőkek csupaszítását könnyíti meg, szippantó segítségével a furatokból a cint tudjuk kiszívni. Ugyancsak furattisztításra használhatjuk az orvosi fecskendő tűjét, ezt nem futja be a cin. A fólia vágására használhatjuk a sniccer kést.

Általánosan elmondhatjuk, hogy az esetek döntő részében áramköreink építéséhez azok szerszámai elegendőek, akik már foglalkoztak elektromos készülékek készítésével, javításával.

## Zambelly Péter





# SZERPENTJN



**Béres László**  
33 éves,  
a Magyar Rádió  
szerkesztőriportere

## ALAPÖTLET

4

Ebben elsősorban az alapötlet újdonságát kívánjuk értékelni, ütőképességét, meglepő voltát, eredetiségét.

Egy már sokak által megírt stratégiai játék variációja. A rosszul választott, elcsépelet cím inkább dezinformál, mint orientál.

## MEGVALÓSÍTÁS

4

Ez az akadály éppen arról szól, amit a neve is jelez: milyen az alapötlet részletekbe menő megvalósítása, a részötletek színvonala, azaz az eleméletben megfogalmazott alapötlet kidolgozása.

Korrekt kidolgozás, néhány kifejezetten jó ötlettel, amelyek elsősorban grafikai jellegűek. A csillagok helye jól elosztott, rövid távon nem kiismerhető, és kedvező az is, hogy a program nem siettet, időt hagy a legértelmesebb stratégia kitalálására. Kedvesek, rokonszenvet keltőek a kacsingató figurák is. Szellemes a még rendelkezésre álló lehetőségek (életek) szívekkel való ábrázolása.

## GRAFIKA

5

Ezt nem kell külön magyarázni, a játékprogramok egyik lényegesebb elemének minősítését értelemszerűen nem hagyhattuk el.

Elsősorban a jó színkihasználás miatt átlagon felüli esztétikai élményt nyújt a játék. A harmonikus színhatást csak egy megoldásában múlja felül a program: a falhozcsapódást kísérő grafikai kép újszerűségében. Az elmosódó, remegő képernyőrajz számomra mindeddig ismeretlen volt.

## HANG

5

Ez sem maradhat ki.

A képernyőn kibontakozó történésekkel szinkronban álló effektusok közül külön figyelmet érdemel a falnak ütközés hanghatása, és a szívdobogás alapmotívuma.

## KEZELHETŐSÉG

5

Sem a játék készítői, sem futtatói számára nem lehet elhanyagolható szempont, hogy egy játékkal hogyan boldogul el az egyszerű, számítógéphez esetleg nem is értő játékos. Alapelvünk az, hogy a jó szoftver önmagát magyarázza, kezelése egy értelmes ember, vagy akár egy programbíró számára semmiféle gondot nem jelent.

A játék egyszerűségét jól követi a kezelhetőség. A gép és a program működése nem akadályozza az élvezetes játéknak.

## ÁTLAG

4,6

Három diák vizsgálta a játékokat, nagyon egyszerű módszerrel. Mindegyikük kapott egy-egy programot, és minimális induló segítség után addig játszhatott vele, amíg csak jólesett. Azt figyeltük, és jegyeztük fel, hogy ki mikor únja meg, hagyja abba, keres valami más elfoglaltságot. Ők nem ismerték szándékunkat, így hát nyugodtan közölték észrevételeiket, fölálltak, ha unták már. 90 percen szabtuk meg a maximális játékidőt



**Balázs Péter**  
(10 éves)  
Hát... láttam már  
jobb számítógépes  
játékokat is!





**Király Zoltán**  
23 éves,  
az ELTE  
harmadéves  
matematikaszakos  
hallgatója,  
gyakorlott programozó



**Puskás Zsolt**  
16 éves,  
a Piarista  
Gimnázium  
másodikos diákja.

## ÁTLAG

**3**

A játék alapötlete nagyon régi, sok hasonlót láthattunk már az eddigi gépeken. A régi változatokhoz képest új ötletek nem használtak a játéknak, nem tették élvezetesebbé. Pozitív az, hogy egy választás végigjátszása után új, nehezebb pályát kapunk.

**2**

Az ötlet a régi kígyós-játék ötlete. A program ennek egy viszonylag gyenge megvalósítása.

**3,0**

**4**

Profi módon megírt játék, de túl sok apró ötletet nem találhatunk benne. Jó, hogy billentyűjáték esetén definiálni lehet a billentyűket. Nem világos, hogy miért nem lehet nehézségi fokozatot, vagy sebességet állítani.

**2**

Nem jöttem rá, hogy lehet elérni azt, hogy újra az első játékos játszhasson. A program egyik legnagyobb fogyatékosága, hogy be lehet hülyíteni. Tehát, ha bejárom az összes mezőt, és nincs több hely, ahova csillagot lehetne tenni, a program örök ciklusba kerül, amiből nincs kilépés.

**3,3**

**4/5**

Profi! A piros villogó csillag helyett talán valami feltűnőbbet is lehetett volna csinálni. A képernyőkezelés mindenestül nagyon jó, a címkép is szép. Egyetlen hiba: billentyűdefiniálásnál, ha valamelyik nyilat írom be, a neki megfelelő szokásos grafikus karakter jelenik meg. Ezt jó lett volna elkerülni!

**2/3**

Sivár! Jól néznek ki a lüktető szívek, de miért nem lehetett hasonló színvonalon megoldani a játékeret is.

**4,0**

**3/4**

Lényegében szokványos, helyenként nem túl kellemes.

**3**

Van hangja, nem rossz.

**3,8**

**3/4**

A játékot billentyűről játszani igen nehéz, mivel a reakcióidő hosszú, és bizonyos határok között kiszámíthatatlan.

**3**

A program nem sok eligazítást ad. Nem jelzi például a játékosok számát, nincsenek fokozatok, ráadásul billentyűzetről az irányítás nehézkes.

**3,8**

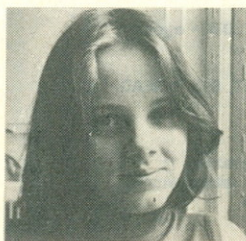
**3,7**



**Tury Kinga**  
(11 éves)  
Hű!!!  
Erre nagyon oda kell figyelni!  
De érdemes...

**26'**

**2,5**



**Molnár Erzsébet**  
(13 éves)  
Jó pofa, szeretem  
az ilyen játékot,  
még játszanék,  
de mennem kell.

**50' 29'**

**3,6**



**Megszámíthatatlan telefon és megszámlálható mennyiségű levél reklamálja az előző számban beígért Plusz 4-es gépeket.** Mint minden ilyen esetben, a szerkesztőség is ludas a dologban, bár igyekszünk a legpontosabb információkat beszerezni az érdekelt vállalatok vezetőitől. Ennek ellenére a cikkben félreérthető megfogalmazás és pontatlan adat is szerepelt. E havi számunkban megkíséreljük a pontosítást, és reménykedünk benne, hogy szeptemberben már kevesebben fognak telefonálni.

**– Hol vannak a július végére várt Plusz 4-es gépek?**

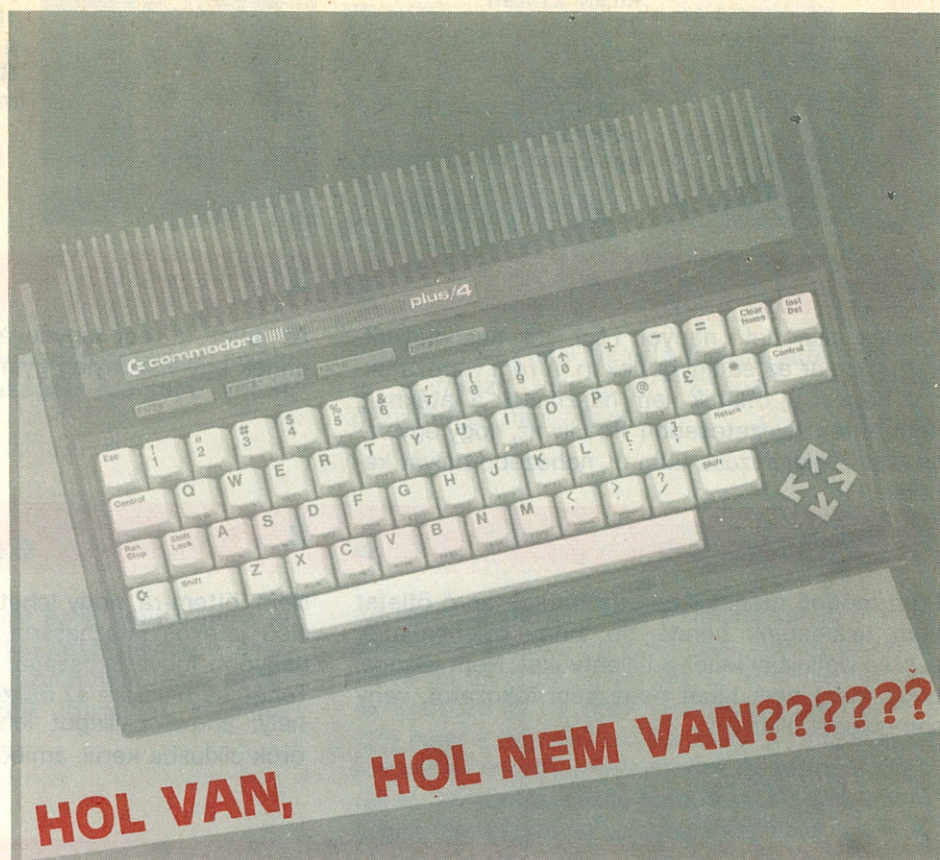
**Rényi Gábor a Novotrade Rt. ügyvezető igazgatója:**

– Sajnos a nemzetközi üzletkötésekben – bár természetesen vannak határidők – sok minden nem látható pontosan előre. Részben az itthoni engedélyezési eljárás, részben a szállítás megszervezése miatt reményeink indokolatlanok voltak, a gépekből 2200 már beérkezett az országba (augusztus 11.). A többivel kapcsolatban én már nem merek semmit mondani, de higgyék el: ugyanúgy várjuk a gépeket, mint Önök.

**Kardos András az ÁPISZ igazgatója:**

– Gondok voltak a szállítással, ezért nem érkeztek meg a gépek. Ha most óvatos lennék azt mondanám, hogy nem tudom mikor jönnek, de azért elárulom: augusztus 12-én elindult a kamion az esetleges szállítmányért. Ennél biztosabb információnk nincs.

**– Lapunkban az jelent meg, hogy a Novotrade 2C boltban magánszemélyek is kaphatnak Plusz 4-est. Az érdeklődésre a bolt dolgozói azt a felvilágosítást adták, hogy erről szó**



**sem lehet. Mi az igazság?**

(Rényi Gábor)

– Mindenki kaphat gépet, aki bizonyítani tudja, hogy oktatási célokra használja. Jogi szempontból tehát „magánszemélyek” – például GMK-k, szülői munkaközösségek – is vásárolhatnak, azonban a bizonyítás nekik is szükséges. Nyilvánvaló, hogy az előző havi

megfogalmazás félrevezető volt, hogy ismét jogi szakkifejezést használjak „természetes személyek” a szükséges igazolást nem tudják biztosítani, így gépet sem kaphatnak. Sajnálom, ha bárki úgy érzi, hogy ezzel félrevezettük, nem állt szándékunkban. Reménykedjünk abban, hogy a következő szállítmány már szabadon vásárolható lesz.

## COMMODORE PLUS 4 RAM LISTÁJA I. RÉSZ

Nulláslap és a rendszerváltozók területe

CIM		LEÍRÁS			
HEXADECIMÁLIS	DECIMÁLIS				
00	0	A processzorport adatirány-regisztere	22-23	34-35	általános mutató 1 #
01	1	a processzorport i/o regisztere	24-25	36-37	általános mutató 2 #
02	2	futás alatti token-keresés a veremben	26-2A	38-42	aritmetikai és stringműveletekre használt memóriaterület
03-04	3-4	sor újraszámozási vektor	2B-2C	43-44	mutató a BASIC program kezdetére
05-06	5-6	sor újraszámozási vektor	2D-2E	45-46	mutató a változóterület kezdetére
07	7	a keresett karakter	2F-30	47-48	mutató a tömbkezdetre
08	8	a stringet záró időzítője kapcsoló	31-32	49-50	mutató a tömbök végére
09	9	az oszlop, a TAB utasításnál	33-34	51-52	mutató a fűzők végére
0A	10	LOAD=0, VERIFY=1, interpreterkapcsoló	35-36	53-54	fűzők segédmutatója
0B	11	mutató az input puffekben / a dimenziók száma	37-38	55-56	mutató a BASIC RAM végére
0C	12	a dimenzió alapértelmezésjelző	39-3A	57-58	az aktuális BASIC sorszám
0D	13	adattípus, \$00 = numerikus, \$FF = szöveges	3B-3C	59-60	CHARGET-mutató
0E	14	adattípus, \$00 = valós, \$80 = egész	3D-3E	61-62	a következő utasítás mutatója CONT parancsnál
0F	15	LIST utasítás kapcsoló	3F-40	63-64	DATA utasítás aktuális sorszáma
10	16	FN kapcsoló	41-42	65-66	mutató a következő DATA elemre
11	17	\$00 = INPUT, \$40 = GET, \$98 = READ	43-44	67-68	az INPUT rutin forrásmutató
12	18	kapcsoló	45-46	69-70	a változó neve
13	19	az ATN előjele	47-48	71-72	a változó címe
14-15	20-21	aktív i/o egység	49-4A	73-74	a változó indexe vagy a következő utasítás mutatója (pl. FOR-NEXT-ben)
16	22	egészértékű cím, pl. a sorszám	4B-4C	75-76	a programmutató eltolásvektora
17-18	23-24	mutató a fűzőveremben (string stack)	4D	77	összehasonlító művelet maszkja
19-21	25-33	mutató az utolsó fűzőre	4E-4F	78-79	mutató az FN-re
		fűzőverem	50-52	80-82	fűzőr-leíró



## CÍM

## LEÍRÁS

## HEXADECIMÁLIS DECIMÁLIS

HEXADECIMÁLIS	DECIMÁLIS	LEÍRÁS	C2	194	inverz karakter nyomtató csatlakozás, 0 = kikapcsolt
53	83	HELP billentyű csatlakozás	C2	194	0 = kikapcsolt
54	84	ugrás a függvényekre	C3	195	logikai sormutató
55	85	ismeretlen (SIZE)	C4-C5	196-197	a kurzor X, Y pozíciója
56	86	ismeretlen (OLDOV)	C6	198	a lenyomott billentyű koordinátái
57	87	aritmetikai regiszter	C7	199	billentyűzet beviteli csatlakozás
58-59	88-89	dimenzionált blokk transzfermutatója, MSB byte-ja	C8-C9	200-201	az aktuális képernyősor mutatója a kurzornak
5A-5B	90-91	blokk-transzfer mutató, MSB byte-ja	CA	202	az aktuális oszlop mutató az adott sorban a kurzornak
5C-5D	92-93	dimenzionált blokk transzfermutatója, LSB byte-ja	CB	203	editor macskakörömmód csatlakozás, Ø = kikapcsolt
5E-5F	94-95	blokk-transzfer mutató, LSB byte-ja	CC	204	az editor munkaterülete
60	96	a beolvasott szám előjele	CD	205	az aktuális kurzor fizikai sorszáma
61	97	lebegőpontos akku #1; exponens	CE	206	az utolsó karakter ASCII értéke
62-65	98-101	lebegőpontos akku #1; mantissza	CF	207	inzertr mód csatlakozás, Ø = kikapcsolt
66	102	a lebegőpontos akku #1 előjele	DO-E8	208-232	nullás oldal szabad területe szoftverek számára
67	103	számláló a polinom kiértékeléséhez	E9	233	képernyősor
68	104	a lebegőpontos akku #1 túlcordulás byte-ja	EA-EB	234-235	az aktuális képernyőszín mutató
69	105	lebegőpontos akku #2; exponens	EC-ED	236-237	a billentyűzet dekódálási vektora
6A-6D	106-109	lebegőpontos akku #2; mantissza	EE	238	a billentyűzet dekóder
6E	110	a lebegőpontos akku #2 előjele	EF	239	a billentyűzet pufferben lévő karakterek száma
6F	111	az #1 és #2 lebegőpontos akkuk előjelét összehasonlító byte	FO	240	szünet csatlakozás
70	112	a lebegőpontos akku #1 kerekítő byte-ja	F1-F2	241-242	monitor zéróalaps munkaterülete
71-72	113-114	a polinom kiértékelése, kazettapuffermutató	F3	243	monitor input puffer mutató
73-74	115-116	automatikus sorszámozás; \$ØØ + növekmény. Ha Ø=nincs AUTONUMBER	F4	244	monitor input adat hossza
75	117	a HIRÉS grafika 10K byte-os foglaltság jelzője	F5	245	ismeretlen (CHKSUM)
76	118	különböző munkaértékek	F6	246	ismeretlen (LENGTH)
77	119	pl. a string hossza a KEY utasításban	F7	247	ismeretlen (PASS)
78	120	közvetett beviteli regisztere	F8	248	ismeretlen (TYPE)
79-7B	121-123	ds\$ - mutató	F9	249	ismeretlen (USEKDY)
7C-7D	124-125	verem teteje futáskor (BASIC verem!!!)	FA	250	a STOP billentyű teszt
7E-7F	126-127	hangerő regiszter	FB	251	aktuális ROM konfiguráció
80	128	hangcsatorna száma	FC	252	karakter a RS232-esre-bekapcsol
81	129	parancsmód csatlakozás	FD	253	karakter az RS232-re-kikapcsol
82	130	tizedesponnt mutató	FE	254	képernyő editor
83	131	aktuális grafikus üzemmód	FF	255	szám konvertáláskor
84	132	aktuális színválasztás	100-10F	256-271	sring print puffer
85	133	multicolour 1	110	272	A regiszter
86	134	előter színe	111	273	X mentő
87	135	max. oszlopszám	112	274	J terület
88	136	max. sorszám	113-122	275-290	RAM szín-tábla
89	137	színezés balra csatlakozó PAINT-hez	124-1FF	291-511	processzor-verem (system stack)
8A	138	színezés jobbra csatlakozó PAINT-hez	200-258	512-600	editor input puffer; aktuális
8B	139	a színezés megáll, ha a háttérszín nem ugyanaz	259-25A	601-602	BASIC sor vagy monitor sor
8C-8D	140-141	grafikai területmutató	25B-25C	603-604	BASIC puffer
8E	142	grafikai rutinok regisztere	25D	605	BASIC puffer
8F	143	grafikai rutinok regisztere	25E-26D	606-621	DOS ciklusszámláló
90	144	a kernel I/O státuszszó: ST értéke	26E	622	file-név #1 puffer
91	145	a STOP billentyű csatlakozója és RVS is	26F	623	file-név #1 hossza
92	146	billentyűzet ellenőrző regisztere	270-271	624-625	meghajtó #1 készülékszáma
93	147	\$ØØ=LOAD, \$Ø1=VERIFY csatlakozó	272	626	file-név #1 mutató
94	148	soros(IEC) kivétel, csatlakozó	273	627	file-név #2 hossza
95	149	az IEC busz output puffere	274-275	628-629	meghajtó #2 készülékszáma
96	150	az Y regiszter mentési területe	276	630	file-név #2 mutató
97	151	nyitott file-szám vagy az end-of-file tábla indexe	277	631	DOS logikai címe
98	152	az INPUT eszköz alapértelmezés, Ø=billentyűzet	278	632	DOS fizikai címe
99	153	a kivételi eszköz, alapértelmezés: 3= TV vagy monitor	279-27A	633-634	DOS másodlagos cím
9A	154	\$ØØ = közvetlen üzemmód, \$ØØ = program mód csatlakozó	27B	635	diszk ID
9B	155	szalag 1. menet, hibajelzés	27C	636	diszk ID csatlakozó
9C	156	szalag 2. menet, hibajelzés	27D-2AC	637-684	DOS output sztring puffer
9D	157	beviteli vég mutató, LSB byte-ja	2AD-2AE	685-686	DOS output szövegfüzér puffer
9E	158	beviteli vég mutató, MSB byte-ja	2AF-AB0	687-688	grafikai változó, X pozíció
9F-A0	159-160	adatregiszter 1.	2B1-2B2	689-690	grafikai változó, Y pozíció
A1-A2	161-162	adatregiszter 2.	2B3-2B4	691-692	grafikai változó, X eltolás
A3-A5	163-165	software óra	2B5-2B6	693-694	grafikai változó, Y eltolás
A6	166	soros olvasás bitszámlálója és beviteli vég csatlakozó	2B7-2B8	695-696	X abszolút értéke
A7	167	a szalagos soros beviteli/kihozatali munkaterülete	2B9-2BA	697-698	Y abszolút értéke
A8	168	a soros busz ciklusszámlálója	2BB-2BC	699-700	X előjele
A9	169	szinvektor	2BD-2C0	701-704	Y előjele
AA	170	soros írás bitszámlálója	2C1-2C2	705-706	grafikai szorzó tényező
AB	171	a file-név hossza	2C3	707	hibaértek
AC	172	a logikai file-szám	2C4	708	kis marker
AD	173	a másodlagos cím	2C5	709	nagy marker
AE	174	az egység szám	2C6-2C7	710-711	a szög előjele
AF-B0	175-176	a file-név mutató	2C8-2C9	712-713	szög szinusz
B1	177	szalag hibaszámoló	2CA-2CB	714-715	szög coszinusz
B2	178	a képernyős I/O kezdőcíme, LSB byte-ja			szögtávolság
B3	179	a képernyős I/O kezdőcíme, MSB byte-ja			
B4-B5	180-181	beviteli címek			
B6-B7	182-183	szalag bázis mutató			
B8-B9	184-185	különböző mutatók			
BA-BB	186-187	szalagírás adat mutató			
BC-BD	188-189	közvetlen fűzér mutató			
BE-BF	190-191	rutin-könyvtár mutató			
CO-C1	192-193	képernyő scroll-regiszter			





# GYORSABBAN GYORSABBAN GYORSABBAN

**A Mikroszámítógép magazin 1986. áprilisi számában  
figyelemre méltó írás jelent meg  
Programgyorsító fogások címmel, melyben a BASIC programok  
futási ideje és megjelenési formája közötti összefüggésekről  
olvashatunk. A cikkben közölt feladat – a képernyő  
első tizenöt sorának feltöltése világoskék 'A' betűvel  
– leglassúbb és leggyorsabb megoldása között  
több, mint tízszeres a futás idő-különbség, és mint majd  
látni fogjuk, van egy még gyorsabb megoldás, melynek  
segítségével ez a különbség ötvenötszörösre nőhet!  
Miből ered ez a hatalmas sebességnövekedés? Erre a kérdésre  
szeretnénk választ adni az alábbiakban.**

Először lássuk a feladat leglassúbb megoldását:

```
10 TI$="000000"
20 PRINT CHR$(147);
30 Z=0
40 POKE 1024+Z,1:POKE 55296+Z,14
50 Z=Z+1
60 IF Z=600 THEN 1000
70 GOTO 40
1000 PRINT TAB(240);TAB(240);TAB(240):
PRINT TI/60;"MASODPERC"
```

Átfutási idő: 16,57 másodperc.

## ELŐSZÖR IS

Helyezzük a POKE utasításokban használt képernyőmemória kezdőcímet (1024), valamint a színmemória kezdőcímet (55296) egy-egy változóba:

```
10 TI$="000000"
20 PRINT CHR$(147);
30 Z=0:KEP=1024:SZIN=55296
40 POKE KEP+Z,1:POKE SZIN+Z,14
50 Z=Z+1
60 IF Z=600 THEN 1000
70 GOTO 40
1000 PRINT TAB(240);TAB(240);TAB(240):
PRINT TI/60;"MASODPERC"
```

A futási idő így 11,37 másodpercre csökken, ami az eredeti 68,6 százaléka.

**Miért gyorsít a programon, ha a konstansokat egy változóba helyezzük? Ennek megértéséhez egy kicsit meg kell ismerkednünk a BASIC interpreter működésével.**

Mikor egy sor szerkesztését a RETURN billentyűvel lezárjuk, a sor a képernyőmemóriából átkerül az ugynevezett input-pufferbe. Itt történik meg a sor tokenizálása. Az interpreter az összes BASIC parancs és függvény nevét egy egyetlen byte-ból álló szimbólummal, az úgynevezett TOKEN-nel helyettesíti. Ez két okból is hasznos: egyrészt a program rövidebb lesz, mivel a teljes, 3-6 betűs parancsot egyetlen karakterrel helyettesíti, másrészt a futást gyorsítja, ugyanis a token segítségével közvetlenül az adott parancsra, illetve függvényre ugorhatunk. A tokenizálás után, ha nincs sorszám, a sort az interpreter rögtön végre is hajtja. Ha van sorszám, akkor a sort a programba illeszti.

**Minden beillesztett BASIC sor felépítése a következő: az első két byte-on egy mutató található, amely a következő sor hasonló mutatójára mutat, ezzel vannak a programsorok egymáshoz fűzve. Ezen mutatóra az ugróutasításoknál van nagy szükség. A mutató után a sorszám következik, majd maga a tokenizált sor. A sor végét egy 0 értékű byte jelzi, majd a következő sor mutatója jön. Ha ez a mutató nulla, akkor vége a programnak.**

A program végrehajtása úgy történik, hogy az interpreter a program elejétől (vagy megadott sorától) kezdve EGYENKÉNT, EGYMÁS UTÁN előveszi a program karaktereit, és megpróbálja értelmezni. Ez a sorban történő értelmezés biztosítja a program parancsainak sorban történő végrehajtását, mivel egy új parancsra csak akkor kerül sor, ha már az előző összes karakterét értelmezte.

**Vizsgáljuk meg, mi a különbség a két POKE utasítás végrehajtása között!** Tegyük fel, hogy az interpreter eljutott a POKE utasításig. Az utasítás tokenje alapján egy táblázatból kikeresi az utasítást végrehajtó gépi kódú szubrutin kezdőcímet, majd elugrik rá (hasonlóan egy ON GOSUB utasításhoz). A POKE utasításnak két paraméterre van szüksége: a címre, és arra az értékre, amelyet a cím által megadott memóriarekeszbe helyez. A cím megadása különbözik a két utasításnál: az elsőnél 1024+Z, a másodiknál KEP+Z. Talán elhihető, hogy a +Z végrehajtása mindkét esetben egyforma, a különbség az 1024 és a KEP kiértékelése között van. **Az 1024 egy tízes számrendszerben felírt szám. Ezt át kell alakítani egy kettes számrendszerbeli, úgynevezett lebegőpontos számmá.** A szám karaktereit egymás után vesszük elő, először az 1-est, majd a 0-át stb. Az átalakítás menete a következő: az első karakter az 1-es, azaz a szám=1. Ezután egy 0 következik, most a szám=1x10+0, stb., így a teljes, 1024-es szám kiértékelése a //1x10+0/x10+2/x10+4 képlettel írható le, ami 3 szorzást és három összeadást jelent, ez kb. 6 msec időt vesz igénybe. Mivel két POKE is van, valamint hatszáz-

szor újra és újra ki kell értékelni, ez kb. 7 másodpercet jelent. **A KEP (azaz a KE, mivel csak az első két karaktert veszi figyelembe az interpreter) változó megkeresése a következőképpen zajlik.**

Mivel az interpreter nem tudja, hogy a KE változót a programban mikor vettük fel (vagy felvettük-e egyáltalán), nem tehet mást, mint hogy a változóterület elejétől kezdve elkezd keresni a KE változót. Ha az adott változó nem a KE, akkor átlépi. Egy ilyen átlépés kb. 70 mikrosec. Ha megtalálta, akkor tulajdonképpen elkészült, mert a kettes számrendszerbeli alakjában van tárolva, így átalakításra nincs szükség. Tehát idővesztéssel csak a változó keresése okoz. Ha megnézzük a programot, első változó a Z, második a KEP, harmadik a SZIN, így a két POKE időszükséglete 126 msec. Időt jelent még magának a változó nevének a kiértékelése, de erről később majd még több szó esik. A két POKE példája azonban bárhol igaz, ahol számot, vagy változót írunk a programba. Ha az adott kifejezésre csak egyszer kerül a vezérlés, akkor a 6 msec idővesztés szinte semmi. De érezhetően megnő a különbség, ha a kifejezést több százszor ki kell értékelni.

**Látható, hogy a változókeresés időszükséglete csak kb. 100 változó esetén éri utol a kifejezésbe írt szám kiértékelésének idejét, azaz ahol csak lehet, a ciklusban szereplő konstansokat még a ciklus előtt tegyük változóba, és a ciklusban a változót használjuk a konstans helyett.**

## TOVÁBB

Ha karakter kódját, a színkódot, valamint a végértéket is változóba tesszük, akkor további javulás érhető el:

```
10 TI$="000000"
20 PRINT CHR$(147);
30 Z=0:KEP=1024:SZIN=55296:BETU=1:KOD=14:VEGE=600
40 POKE KEP+Z,BETU:POKE SZIN+Z,KOD
50 Z=Z+1
60 IF Z=VEGE THEN 1000
70 GOTO 40
1000 PRINT TAB(240);TAB(240);TAB(240):
PRINT TI/60;"MASODPERC"
```

A futási idő így 9,4 másodperc, ami az eredeti 56,7 százaléka.

**A javulás elég nagy, de két okból nem annyira jelentős; mint az első esetben.** Egyrészt itt nem 4 illetve 5 jegyű, hanem csak 1 és 2 jegyű számokat helyettesítünk változóval, és mint láttuk, a kiértékelés ideje (az összeadások és a szorzások számának növekedése miatt) a számjegyek számától függ, míg a változókeresés ideje csak a változók számától függ, a változó értékétől nem. Másrészt a változók neve eléggé hosszú, nem 1 vagy 2 karakteres, a felesleges karakterek átlépése pedig időbe kerül, főleg, hogy a ciklusban minden felesleges karaktert hatszázszor kell átlépni.

## MÉG TOVÁBB

Nosza, vezessünk be egykarakteres változóneveket! Programunk most így fest:

mutató 2 byte	sorszám 2 byte	a tokenizált sor	0	mutató 2 byte
------------------	-------------------	------------------	---	------------------



```

10 TI$="000000"
20 PRINT CHR$(147);
30 Z=0:K=1024:S=55296:B=1:D=14:V=600
40 POKE K+Z,B:POKE S+Z,D
50 Z=Z+1
60 IF Z=V THEN 1000
70 GOTO 40
1000 PRINT TAB(240);TAB(240);TAB(240):
PRINT TI/60;"MASODPERC"

```

A végrehajtási idő 8,92 másodperc, 53,8 százalék. Ez is javulást jelent. Különben is, a két karakternél hosszabb változónevek könnyen összekeveredhetnek. Az ilyen hibát pedig igen nehéz megtalálni. Legjobb, ha kevés, lehetőleg egybetűs változót használunk.

## NAMÁRMOST

A következő változtatás: cseréljük ki a POKE-okat PRINT-re,

Valamint szüntessük meg a TAB utasításokat az 1000-es sorban:

```

10 TI$="000000"
20 PRINT CHR$(147);
30 V=600
40 PRINT CHR$(65);
50 Z=Z+1
60 IF Z=V THEN 1000
70 GOTO 40
1000 PRINT TI/60;"MASODPERC"

```

Így a futási idő 6,35 másodperc, 38,3 százalék.

Az eredeti programban a PRINT-et két POKE utasítással valósítottuk meg. A két POKE-nak összesen 4 paramétert kell kiértékelni. A PRINT utasításnak csak egy paramétere van, a CHR\$(65), maga a PRINT rutin pedig eléggé gyors. A javulás tehát abból ered, hogy a két POKE-ot egy utasítással helyettesítettük.

## EZ MIND SEMMI!

Ha az „A” betű kódját, a CHR\$(65)-öt változóba helyezzük, tovább gyorsul a program:

```

10 TI$="000000"
20 PRINT CHR$(147);
30 V=600:A$=CHR$(65)
40 PRINT A$;
50 Z=Z+1
60 IF Z=V THEN 1000
70 GOTO 40
1000 PRINT TI/60;"MASODPERC"

```

Az eredmény 4,82 másodperc, 29,1 százalék.

Ennek okát tulajdonképpen már láttuk. A CHR\$(65) esetén a 65-öt ki kell értékelni, át kell alakítani kettes számrendszerbeli számmá, majd ebből képzni a gép a megfelelő kódú karaktert. A változóba helyezett string esetén pedig csak a változókeresés igényel időt, ami, mint már láttuk, jóval kevesebb.

## MÉG EGY KICSIT

Azonban a változókeresés idejét is meg lehet takarítani, ha a stringet közvetlenül a PRINT utasításba írjuk:

```

10 TI$="000000"
20 PRINT CHR$(147);
30 V=600
40 PRINT "A";
50 Z=Z+1
60 IF Z=V THEN 1000
70 GOTO 40
1000 PRINT TI/60;"MASODPERC"

```

Így már csak 4,7 másodperc a futási idő, ami az eredeti 28,4 százaléka.

Ha a 60-as sorban az IF utasításban lévő THEN-t GOTO-ra cseréljük, ismét csökken a futási idő. Ennek egyszerű oka van. Az interpreter előbb azt vizsgálja, hogy GOTO token van-e az IF feltétele után, és ha nem, csak akkor nézi meg, hogy THEN-e. Ez egy nagyon kicsi idővesztés, de ne feledjük: ez is benn van a ciklusban, és a hatszázszor kicsi már sok is lehet.

## MÉG NINCS VÉGE

Ha megfordítjuk a feltételt a 60-as sorban, tovább javul az időeredményünk.

```

10 TI$="000000"
20 PRINT CHR$(147);
30 V=600
40 PRINT "A";
50 Z=Z+1
60 IF Z<V THEN 40
1000 PRINT TI/60;"MASODPERC"

```

A futási idő: 4,63 másodperc, 29,9 százalék.

Vegyük észre, hogy az előző megoldásban egy sorral később került sor a visszaugrásra, mint most. Márpedig minden sor elején fel kell jegyezni a sorszámot, hogy hiba esetén ki tudja írni. Minden utasítás értelmezése előtt tárolni kell az utasítás kezdőcímét, hogy ha a programot megállítjuk, majd CONT-tal újraindítjuk, az interpreter tudja, hogy honnan kell folytatni. Az adminisztrálás pedig időbe telik, ráadásul ezt is hatszázszor kell megcsinálni.

## NO, PERSZE

A következő változtatás: a ciklust FOR-ral szervezzük.

```

10 TI$="000000"
20 PRINT CHR$(147);
30 FOR Z=1 TO 600
40 PRINT "A";
50 NEXT Z
1000 PRINT TI/60;"MASODPERC"

```

Ennek a változatnak a futási ideje: 1,68 másodperc, 10,1 százalék. A FOR-NEXT utasításpár a Z=0, Z=Z+1, valamint az IF ...: utasításokat helyettesíti. A FOR-t a ciklusszervezésre találták ki, és mindent elkövettek, hogy a lehető leggyorsabb legyen. Például a FOR-NEXT ciklus a ciklusváltozó helyét eltárolja, így a cikluson belül nincs szükség a ciklusváltozó keresésére, amire pedig a másik esetben 3-szor is időt pazarlunk. Nem is beszélve arról, hogy a FOR-NEXT ciklus a növekményt is eltárolja, így azt sem kell keresni, átalakítani, mint az előző esetben.

Bárki kipróbálhatja, mennyivel gyorsabban fut egy üres FOR-NEXT ciklus, mint a Z=0, Z=Z+1, IF Z=... ciklus. Ebből is látszik: ha az adott feladatra van BASIC utasítás vagy függvény, akkor még véletlenül se próbáljuk meg azt több más utasítással helyettesíteni, mert a futási idő rettentően megnő.

Ha a NEXT-ből elhagyjuk a Z ciklusváltozót, tovább gyorsul a programunk.

Az új időeredmény: 1,53 másodperc, 9,2 százalék.

Ennek oka is ismert már. A Z-t feleslegesen kell kiértékelni, és minden, ami felesleges, az csak az időt húzza.

## MÉG MINDIG

Felesleges adminisztrációra kényszerítjük a gépet azzal is, ha az utasításokat külön sorba írjuk (már láttuk, fel kell jegyezni a sorszámot). Írjuk hát egy sorba az egész programot! Ezenkívül feleslegesek a szóközlők az utasítások és a paraméterek között, ezeket az interpreter csak átlépi, de az átlépés is idő. (bár kétségtelen, hogy a programlista szebb, áttekinthetőbb velük.) És ne feledjük: minden kis idő a 600-szorosára nő, ha a ciklusban van. Programunk most így fest:

```

10 TI$="000000":PRINTCHR$(147);:FORZ=1TO
600:PRINT"A";:NEXT:PRINTTI/60;"MASODPERC"

```

Időeredményünk 1,47 másodperc, 8,87 százalék.

## NO VÉGRE

Az utolsó sebességnövelő lépés gondolata már önként adódik: A ciklusszámot kellene valahogyan csökkenteni, hogy ne kelljen 600 PRINT-et és 600 NEXT-et végrehajtani. Erre itt megvan a lehetőség. A PRINT-be ne egy, hanem mondjuk hatvan darab "A" betűt írjunk. Így a ciklushossz 10-re csökken.

```

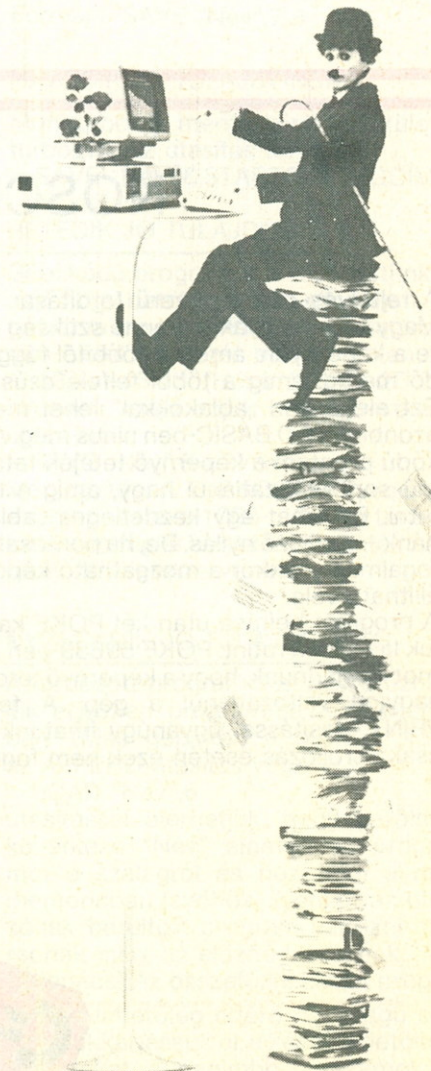
10 TI$="000000":PRINTCHR$(147);:FORZ=1TO
10
20 PRINT"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA";:NEXT
30 PRINTTI/60;"MASODPERC"

```

Programunk ezen változata:

Futási idő: 0,3 másodperc, ami az eredetinek 1,8 százaléka. Ezen az utolsó változtatáson is jól látható a lényeg: próbáljuk meg a ciklusokat a lehető legrövidebbre készíteni, a ciklusban ne legyen semmilyen felesleges utasítás, de még egy felesleges karakter sem. A hosszú konstansokat (pláne, ha előjelet, tizedespontot, vagy más időhúzó karaktert is tartalmaznak) helyezzük változóba.

Mint látjuk, fáradásunk meghozta gyümölcsét. Az eredeti, csigalassú programot sikerült alaposan felgyorsítani, és ehhez csak egy kis figyelem kellett.





## PERIFÉRIÁK

Gyakran jelent gondot egyszerűbb BASIC programoknál, hogy periféria hibánál a program leáll, hibaüzenetet ír ki. A következő két egyszerű rutint mindig a perifériaműveletek előtt érdemes meghívni. Ezek levizsgálják a perifériák állapotát, és ha azokkal valami nincs rendben nem állnak le hibaüzenettel, hanem megfelelő sorra ugrathatók. Ha minden rendben van, akkor már jöhet is a periféria kezelés.



```

1 REM C= UJSAG SORSZAM 016
10 REM DATA WELT SOMMER '84 110. OLDAL
11 REM *****
12 REM * SZUBRUTIN A PERIFERIAK *
13 REM * VIZSGALATARA *
20 REM *****
24 REM * NYOMTATO *
25 REM *****
30 OPEN 1,4:POKE 768,61
40 PRINT#1:CLOSE 1
50 POKE 768,139
60 IF ST <>-128 THEN 90
80 PRINT"NYOMTATO NINCS BEKAPCSOLVA"
90 RETURN
100 REM *****
102 REM * FLOPPY *
104 REM *****
110 POKE 768,61
120 OPEN 1,8,15,"I":CLOSE 1
130 POKE 768,139
140 IF ST <>-128 THEN 170
160 PRINT" FLOPPY NINCS BEKAPCSOLVA"
170 RETURN

```

## NOSCROLL

A rejtélyes cím egyszerű fordítása: NE LAPOZZI! Magyarázata: gyakran lenne szükség olyan területre a képernyőn, amely a többitől függetlenül állandó marad, amíg a többi felfelé csúszik, scroll-oz. Ezt elegánsan „ablakokkal” lehet megoldani, ami azonban a 2.0 BASIC-ben nincs meg. Az alábbi gépi kódú program a képernyő tetején tetszőleges számú sort változatlanul hagy, amíg a többi engedni futni. Ez tehát egy kezdetleges „ablak”, modhatnánk – szellőző nyílás. De, ha pontosabban akarunk fogalmazni, akkor a mozgatható képernyő mérete állítható vele.

A program beírása után két POKE-kal kapcsolhatjuk ki és be a rutint. POKE 59639-ben kell azt a számot megadnunk, hogy a képernyő tetején hány sort hagyjon változatlanul a gép. A felső sorokba PRINT utasítással ugyanúgy írhatunk továbbra is, csak scrollozás esetén ezek nem fognak eltűnni.



```

10 REM *****
11 REM * C= UJSAG SORSZAM 018 *
15 REM * DATA WELT 85/12 73.OLDAL*
31000 REM *****
31010 REM *** COMMODORE 64 NOSCROLL ***
31020 REM *** $9F00 - $9FD8 ***
31030 REM *** BEKAPCSOLAS: POKE1,53 ***
31035 REM *** KIKAPCSOLAS: POKE1,55 ***
31040 REM *** SOROK SZAMA: ***
31055 REM *** POKE59639, SZAM***
31070 REM *** IRTA:STEFAN SIEMONEIT ***
31080 REM *****
31085 :
31100 FOR I=0 TO 215:READ X:POKE40704+I,
X:S=S+X:NEXT
31105 IF S<24214 THEN PRINT"MHIBAS ADAT
BEIRAS":END
31110 SYS 40704:END
32000 DATA169,147,32,210,255,169,0,141,3
2,208,141,33,208,169,158,32,210,255,169
32001 DATA0,133,251,169,160,133,252,162,
49,160,0,177,251,145,251,200,208,249
32002 DATA202,240,5,230,252,76,30,159,16
9,0,133,251,169,224,133,252,162,49,160
32003 DATA0,177,251,145,251,200,208,249,
202,240,5,230,252,76,57,159,169,53,133
32004 DATA1,169,4,141,247,232,169,0,170,
160,170,185,210,158,32,210,255,200,208
32005 DATA247,96,162,225,189,1,9,232,32,
210,255,16,247,169,32,32,210,255,32,210
32006 DATA255,32,210,255,32,210,255,76,1
09,8,234,147,32,32,32,42,42,42
32007 DATA32,67,79,77,77,79,68,79,62,69,
32,54,52,32,78,79,83,67,82,79,76,76,32
32008 DATA42,42,42,42,13,13,158,32,54,52
,75,32,82,65,77,32,83,89,83,84,69,77
32009 DATA32,32,51,56,57,49,49,32,66,65,
83,73,67,32,66,89,84,69,83,32,70,82,69
32010 DATA69,19,13,13,13,32,32,32,32,32,
32,32,32,32

```



# TURBOBASIC

**A BASIC-Programozás és gyors magnókezelés teljesen új útjait nyitja meg ez a program a Commodore 16, 116, és PLUS 4-es gépeken. Jól szervezett változókezelésével és a turbóval kombináltan már ko-**

**molyabb adatkezelés is megvalósítható, továbbá a programtesztelés és javítás során is rendkívüli segítséget nyújt. Ez a szerkesztőség véleménye. A többi dicséret már magától a szerzőtől.**

## ELSŐ JÓ TULAJDONSÁG

A COMMODORE BASIC eredetileg maximálisan két képernyősor bevitelét teszi lehetővé. Most azonban lehetőség van hat képernyősor egy BASIC-sorként való bevitelére! Ez különösen helytakarékoskodás szempontjából nagy jelentőségű, mert minden külön sor plusz 4 byte-tal többet venne el a memóriából. Az IF...THEN...ELSE utasítást pedig kiválóan lehet így használni, mert az ide felhasználható utasítások könnyebben elférnek a hat sorban.

## MÁSODIK JÓ TULAJDONSÁG

Egy BASIC-sor javítása csak egyetlen byte-tal is azt eredményezi, hogy az esetlegesen használt változók, stringek és tömbök értékei azonnal törlődnek. A program azonban biztosítja a programjavítást a változók, tömbök és stringek értékeinek megtartása mellett is! Természetesen a programok tesztelése így könnyebbé válik, mert a programfutás megszakítása és a hibajavítás után a program GOTO-val a változók eredeti értékei mellett tovább folytatható.

## HARMADIK JÓ TULAJDONSÁG

A Commodore gépeknél sehol sem találkozunk az egyes változók törlési lehetőségeivel. A CLR utasítás ugyanis mindent töröl. Ez különösen tömbök esetében hátrányos, mivel ha rájövünk, hogy egy feltöltött tömb dimenzionálása kicsinek bizonyul vagy fordítva, egyszerűen lehetetlen az újradimenzionálás megoldása az adatok elvesztése, vagy rendkívül körülményes megtartása nélkül. Ha ugyanis dimenzionálunk egy nagyobb, és áttöltjük az előbbi tömb adatait, a régit kitörölni nem lehet, ott marad és foglalja a helyet. Kénytelenek vagyunk az adatokat file-okban a szalagra vagy floppyra tenni, ami a TURBOBASIC használatában nem szükséges. A CLR utasítás után veszőkkel elválasztva felsorolhatjuk azokat a változókat, stringeket és tömböket, amiket törölni akarunk, a többi pedig marad érintetlen, a töröltek helye viszont felszabadul.

A helyes formátum: CLR (változó 1, változó 2, változó 3...)

Pl.: CLR A, A\$, A (0,0), A%(0,0)

így egy tömb újradimenzionálása a következő lépésekben történhet meg: Pl. A\$(20,20) helyett A\$(100,20)

1. Dimenzionálunk egy másikat (DIM B\$(20,20)
2. Áttöltjük az adatait (FOR ciklussal)
3. Töröljük az első tömböt (CLR A\$(0,0)
4. Dimenzionálunk egy helyeset (DIM A\$(100,20))
5. Visszatöltjük az adatokat (FOR ciklussal)
6. És töröljük a segédtömböt (CLR B\$(0,0))

Ha CLR utasítás után nem áll változó, akkor mindent töröl. A változókat törlik egyébként a következő utasítások is: NEW, RUN, DELETE. Ha sorokat akarunk törölni úgy, hogy megmaradjanak a változók, akkor üssük be a sorszámát üresen. Ha sok sort akarunk törölni, és ragaszkodunk a DELETE-hez is, akkor viszont mentsük ki a változókat szalagra, töröljük DELETE-tel, majd hozzuk vissza a változókat a turbóprogrammal. Sokkal gyorsabb, mintha az eredeti file-okban vinnénk ki. A helyes módját pedig alább leírjuk.

## NEGYEDIK JÓ TULAJDONSÁG

A programokat nagy gyorsasággal menthetjük szalagra és tölthetjük be szalagról.

A helyes formátum:

SAVE "Név", 7,n

LOAD "Név", 7

ahol "n" a különböző turbómódokat jelenti. Ha elhagyjuk, akkor közönséges turbósan viszi ki szalagra a programot. TURBO üzemmódok

SAVE "Név",7,1 a programot változóival együtt menti ki.

Betöltésekor nehogy RUN-nal indítsunk, mert kitörli a kimentett változókat! Indításhoz kitűnő parancs a "GOTO sorszám" is.

Hasznos tanács: Ha kimentés előtt tömöríteni akarjuk a stringeket, (ugyanis az INTERPRETER elég szellősen tárolja őket) akkor használjuk a FREE utasítást. Pl.:?FREE(0). A FREE-rutin ugyanis a számítás előtt tömöríti a stringeket. SAVE "Név",7,2 duplasebességű mentés olvasás.

SAVE "Név",7,4 sorszám betöltésekor rögtön az egyik BASIC-sorra történik a

vezérlésátadás, mintha GOTO-val indítottuk volna el.

SAVE "Név",7,2 + 1, azaz SAVE "Név",7,3 ötvözhetjük a paramétereket a kívánságnak megfelelően, például egy programot a változóival duplasebes üzemmódban menthetünk ki.

## ÖTÖDIK JÓ TULAJDONSÁG

A változók, stringek és tömbök önmagukban is kimenthetőek! Ez különösen nagy adatállomány esetén rendkívül hasznos. Jobb, és gyorsabb adatkezelés valósítható meg, mint az eredeti OPEN-CLOSE-os file-technikával.

Formája: SAVE "Név",7,8

## HATODIK JÓ TULAJDONSÁG

Monitorból is menthetünk ki területet turbózva. Az utasítás formája:

SAVE "Név",6,STARTCIM,VÉGCIM

## HETEDIK JÓ TULAJDONSÁG

Gépi kódú programot is menthetünk ki úgy, hogy betöltése után elinduljon, és turbózva legyen. Ehhez először a program elejét és végét be kell állítanunk a 43,44-45,46 (Hexa 2B,2C-2D,2E) címeken, majd adjuk be a következő utasítást.

SAVE "Név",6,4,SYSCIM

Megjegyzés: A TURBOBASIC program a memória hexa 1000-1900-ig terjedő területén foglal helyet. A kimentés az e fölötti területekre vonatkozik.

## NYOLCADIK JÓ TULAJDONSÁG

A turbózott programokat összefűzhetjük egyetlen programmá a memóriában. (Más gépeken a MERGE vagy az APPEND utasítás végzi el.)

LOAD "Név",6

utasítással elérhetjük, hogy betölti a következő "Név" elnevezésű programot a szalagról és hozzáfűzi a már memóriában levőhöz. Ilyenkor az utolsónak betöltött program változói maradnak meg, az előzőé elvesznek.

Jótanács! Az összefűzni kívánt programokat lehetőleg eltérő nagyságú sorszámokkal lássuk el, a később betölteni kívántakat a nagyobb sorszámokkal. Kiváló eszköz ehhez a RENUMBER utasítás.



# TURBOBASIC

használati  
utasítás  
a 17. oldalon  
a program  
működésének  
magyarázata  
a 20. oldalon

1100	A6	31	LDR	#31
1102	A6	08	STX	#08
1104	A6	32	LDR	#32
1106	06	01	STX	#01
1108	20	3E	SE JSR	#8E3E
110E	04	08	89 JSR	#8953
1110	20	53	09 STY	#08
1113	98	4A	8A JSR	#8A3D
1115	A0	01	LDR	#01
1117	A0	01	04 JSR	#0401
111A	A0	23	STA	#23
111C	A0	23	LDR	#23
111E	A0	23	STA	#23
1120	A0	23	LDR	#23
1122	85	25	STA	#25
1124	88		DEY	
1125	20	D1	04 JSR	#04D1
1128	18		CLC	
1129	E5	5F	SBC	#5F
112B	49	FF	EOR	##FF
112D	18		CLC	
112E	65	08	ADC	#31
1130	65	31	STA	#31
1132	A0	24	LDR	#24
1134	A0	24	ADC	##FF
1136	A0	32	STA	#32
1138	85	32	SBC	#60
113A	E5	60	TAX	
113C	AA		SEC	
113D	38		SEC	
113E	A5	5F	LDR	#5F
1140	E5	31	SBC	#31
1142	A0		TAX	
1143	88	03	BDS	#1148
1145	08		INX	
1146	08	25	DEC	#25
1148	18	22	CLC	
114B	20	22	ADC	#22
114D	20	22	BCC	#1150
114F	18	23	DEC	#23
1150	20	00	04 JSR	#0400
1153	91	24	STA	#(24),Y
1155	00		INY	
1156	00	F8	BNE	#1150
1158	00	23	INC	#23
115A	00	25	INC	#25
115C	0A		DEX	
115E	00	F1	BNE	#1150
115F	20	0B	11 JSR	#110B
1162	20	18	88 JSR	#8818
1165	A0	00	LDR	#00
1167	20	A5	04 JSR	#04A5
116A	F0	58	BEQ	#11C4
116C	18		CLC	
116D	A5	31	LDR	#31
116F	A4	32	LDR	#32
1171	85	5A	STA	#5A
1173	84	5B	STY	#5B
1175	65	0B	ADC	#0B
1177	90	01	BCC	#117A
1179	C8		INY	
117A	18		CLC	
117B	69	04	ADC	##04
117D	90	01	BCC	#1180
117F	C8		INY	
1180	85	58	STA	#58
1182	94	59	STY	#59
1184	20	C0	88 JSR	#88C0
1187	A0	00	LDR	#00
1189	A0	01	LDR	#01
118B	91	5F	STA	(\$5F),Y
118D	C8		INY	
118E	91	5F	STA	(\$5F),Y
1190	C8		INY	
1191	A5	14	LDR	#14
1193	91	5F	STA	(\$5F),Y
1195	A5	15	LDR	#15

1197	C8		INY	
1198	91	5F	STA	(\$5F),Y
119A	C8		INY	
119B	98		TYA	
119C	18		CLC	
119D	85	5F	ADC	#5F
119F	65	02	STA	#5F
11A1	90	02	BCC	#11A5
11A3	84	60	INC	
11A5	84		NOP	
11A6	84		NOP	
11A7	84		NOP	
11A8	84		NOP	
11A9	84		NOP	
11AA	84		NOP	
11AB	84		NOP	
11AC	A4	0B	LDR	#0B
11AE	84		DEY	
11AF	A5	A5	04 JSR	#04A5
11B2	91	5F	STA	(\$5F),Y
11B4	C0	09	CPY	#09
11B6	00	F6	BNE	#11AE
11B8	20	C8	11 JSR	#8818
11BE	20	60	18 JSR	#1860
11C1	4C	E4	87 JMP	#87E4
11C4	4C	E4	11 JMP	#11EA
11C7	00		BRK	
11C8	20	F1	8A JSR	#8AF1
11C8	20	E7	FF JSR	#FFE7
11CE	A0	00	LDR	#00
11D0	88	79	STY	#79
11D2	88		DEY	
11D3	8C	F3	04 STY	#04F3
11D6	8C	F0	04 STY	#04F0
11D9	8C	F1	04 STY	#04F1
11DC	8C	EF	04 STY	#04EF
11DF	89	80	LDR	#80
11E1	A0	07	LDR	#07
11E3	85	7C	STA	#7C
11E5	84	7D	STY	#7D
11E7	4C	0A	8A JMP	#8ACA
11EA	20	60	8F JSR	#8660
11F4	38	0F	87 SEC	#870F
11F8				
11F1	A5	31	LDR	#31
11F3	E5	00	SBC	#00
11F5	85	00	STA	#00
11F7	A5	32	LDR	#32
11F9	E5	01	SBC	#01
11FB	85	01	STA	#01
11FD	18		CLC	
11FE	A5	2D	LDR	#2D
1200	65	00	ADC	#00
1202	65	2D	STA	#2D
1204	A5	2E	LDR	#2E
1206	65	01	ADC	#01
1208	85	2E	STA	#2E
120A	18		CLC	
120B	A5	2F	LDR	#2F
120D	65	00	ADC	#00
120F	85	2F	STA	#2F
1211	A5	30	LDR	#30
1213	65	01	ADC	#01
1215	85	30	STA	#30
1217	60		RTS	
1218	A2	FF	LDR	##FF
121A	86	3A	STX	#3A
121C	A2	00	LDR	#00
121E	20	31	A7 JSR	#1731
121F	09	00	00 JMP	#0000
1223	F0	08	BEQ	#1230
1225	90	00	10 STA	#1000,X
1228	E8		INX	
1229	00	FA	CPX	##FA
122B	90	F1	BCC	#121E
122D	4C	0C	CC JMP	#CC4C
1230	A0	00	LDR	#00
1232	90	00	10 STA	#1000,X
1235	A2	00	LDR	#00
1237	A0	00	LDR	#00
1239	20	3A	90 JSR	#903A
123C	86	3B	STX	#3B
123E	84	3C	STY	#3C
1240	20	73	04 JSR	#0473
1243	AA		TRX	
1244	F0	A7	BEQ	#11E0
1246	90	03	BCC	#1248
1248	4C	00	16 JMP	#1600
124B	4C	00	11 JMP	#1100
124E	00		BRK	
124F	00		BRK	
1250	80	0B	LDR	#0B
1252	A0	12	LDR	#12
1254	80	2E	03 STA	#032E
1257	80	2F	03 STY	#032F
125A	80	2E	03 STA	#032E
125C	A0	13	LDR	#13
125E	80	30	03 STA	#0330
1261	8C	31	03 STY	#0331
1264	A9	18	LDR	#18
1266	A0	12	LDR	#12
1268	8D	03	03 STA	#030D
1269	80	03	03 STY	#0300
126E	A9	E0	LDR	#E0
1270	A0	16	LDR	#16
1272	80	08	03 STA	#0308
1275	8C	09	03 STY	#0309
1278	A9	01	LDR	#01
127A	A0	19	LDR	#19
127C	85	2B	STA	#2B
127E	84	2C	STY	#2C
1280	A9	00	LDR	#00
1282	80	00	19 STA	#1900
1283	80		RTS	
1285	00		BRK	
1287	00		BRK	
1288	00		BRK	
1289	00		BRK	
128A	00		BRK	
128B	00		BRK	
128D	85	93	STA	#93
128F	84	90	STY	#90
1291	A5	RE	LDR	#RE
1293	C9	07	CPY	#07
1295	F0	07	BEQ	#129E
1297	C9	07	CPY	#07
1299	F0	03	BEQ	#129E
129B	4C	50	F0 JMP	#F050
129E	A0	19	FF LDR	#FF19
12A1	85	08	STA	#08
12A3	20	18	E3 JSR	#E318
12A6	B0	28	BDS	#12D0
12A8	A5	93	LDR	#93
12AA	48		PHA	
12AB	00		LDR	#00
12AD	85	93	STA	#93
12AF	A0		LDR	#00
12B1	85	00	STA	#00
12B3	20	74	15 JSR	#1574
12B6	20	64	15 JSR	#1564
12B9	A9	52	LDR	#52
12BB	20	60	LDR	#60
12BE	B0	F3	BDS	#12B3
12C0	A5	9A	LDR	#9A
12C2	10	0F	BPL	#12D3
12C4	20	C0	03 JSR	#E8C0
12C7	20	C0	03 JSR	#E8C0
12CA	20	E0	03 JSR	#E9E0
12CC	90	04	BCC	#12D3
12CF	68		PLA	
12D0	4C	AA	15 JMP	#15AA
12D3	A0	FF	LDR	##FF
12D5	C8		INY	
12D6	C4	AB	CPY	#AB
12D8	F0	0A	BEQ	#12E4
12DA	20	C0	03 JSR	#15C0
12DB	D9	37	03 CMP	#0337,Y
12DD	F0	F3	BNE	#12D5
12DE	00	CF	BNE	#12B3
12E2	00	CF	BNE	#12B3
12E4	70	68	PLA	
12E5	70	68	BEQ	#12E9
12E7	A9	80	LDR	#80
12E9	85	93	STA	#93
12EB	A6	2B	LDR	#2B
12ED	A4	2C	LDR	#2C
12EF	A5	RE	LDR	#RE
12F1	C9	06	CPY	#06
12F3	D0	0B	BNE	#1300
12F5	38		SEC	
12F6	A5	2D	LDR	#2D
12F8	A4	2E	LDR	#2E
12FA	E9	02	SEC	##02
12FC	AA		TRX	
12FE	B0	01	BDS	#1300
12FF	88		DEY	
1300	A0	32	03 LDR	#0332
1303	29	48	AND	##48
1305	F0	0E	BEQ	#1315
1307	AE	33	03 LDX	#0333
130A	AC	34	03 LDR	#0334
130D	29	40	AND	##40
130E	D0	04	BNE	#1315
1310	A5	2D	LDR	#2D
1313	86	01	STX	#01
1317	84	C2	STY	#02
1319	38		SEC	
131A	AD	35	03 LDR	#0335
131D	E0	33	03 BCC	#0333
1320	95	03	STA	#03
1322	AD	36	03 LDR	#0336
1325	E0	34	03 SBC	#0334
1329	95	04	STA	#04
132B	18		CLC	
132D	65	D3	ADC	#D3
132E	85	D3	STA	#D3
1330	98		TYA	
1331	65	D4	ADC	#D4
1333	85	D4	STA	#D4
1335	A2	C0	LDR	##C0
1337	AD	32	03 LDX	#0332
133A	10	02	03 BPL	#133E
133C	A2	4C	LDR	##4C
133E	86	D0	LDR	#D0
1340	29	64	15 JSR	#1564
1343	A9	A6	LDR	##A6
1345	20	60	14 JSR	#1460
1348	65	00	BDS	#1350
134A	20	63	15 JSR	#1563
134D	4C	05	16 JMP	#1605
1350	20	63	15 JSR	#1563
1353	38		SEC	
1354	A9	10	LDR	##10
1356	65	90	STA	#90
1358	A5	93	LDR	#93
135A	F0	03	BEQ	#135F
135C	A9	1C	LDR	##1C
135E	60		RTS	
1360	A0	1D	LDR	##1D
1361	A0		RTS	
1362	A5	RE	LDR	#RE
1364	C9	07		







# program működése

A program startcíme \$1250. Ez egy rövid rutin, amely beállítja a LOAD, SAVE, Interpreter start, Token szerinti utasítás végrehajtást végző vektorokat, továbbá eltolja a BASIC-eleje mutatót \$1901-re a TURBOBASIC-program végére. Így megoldható, hogy bővíthető is vagy Plusz 4-sel is használható.

Az új címek: LOAD = \$128B  
SAVE = \$1362  
Melegstart = \$1218  
Végrehajtó = \$16E0

Miért volt szükség a vektorok átírására? A LOAD, SAVE-vektorok átírása a turbo, és a változók mentése-töltése miatt kellett. A melegstart átírására első sorban azért volt szükség, mert az inputpuffer megnagyobbítását a \$200-as eredeti címen nem lehetett megoldani helyhiány miatt. Ezért az inputpuffert \$1000-re tettem, viszont ennek a mérete és címzése miatt szinte a teljes melegstart rutint át kellett írni. A másik, hogy az inputpufferből a memóriába átrámolt bármelyik bevitt BASIC-sor eredetileg törli az összes változót. S mivel ennek a folyamata itt zajlik le, így ezért is módosítani kell az interpreter melegstart rutinját.

A végrehajtó rutin vektorát a módosított CLR-parancs miatt kellett átírni, hogy az új rutinokkal szabályozni lehessen a megfelelő változó törlési műveleteket.

**A program főbb lépésekben a következőket tartalmazza:**

**\$1100-\$11CH** – ez a beadott BASIC-sort helyezi el a memóriában, ezért kiszámolja a megfelelő sorszámot, hosszt, ebből a szükséges startcímet, eltolja a mögötte levő programot és a VÁLTOZÓKAT megfelelő byte-tal, beilleszti az új sort, összláncolja a többivel és a stringváltozó pointerit az eltolásnak megfelelően újraállítja.

\$1109: JSR \$8E3E – ez a sorszám kiszámító rutint hívja

\$110B: JSR \$8953 – ez a tokenizáló rutin, SB-re kell a bevitt sor új hossza.

Tokenizálással ugyanis rövidebb lesz! \$1110: JSR \$8A3D – ez megkeresi a helyét a memóriában és a flagokat állítja be ennek megfelelően. C=0, ha nincs ilyen sor, akkor törölni nem kell. \$1115-\$1150-ig – az azonos sor törlése történik, ehhez a megfelelő pointeret is beállítjuk.

\$115F: JSR \$11CB – rutin a szükséges BASIC állapotot állítja be normálra. Például a veremtermutatót elejére állítja.

\$1162: JSR \$8818 – ez a programfűzéshez szükséges sorleje pointeret beállítja a változáshoz.

\$1165 – itt elhozza a pufferből az első karaktert. Ha az nem nulla, akkor a hosszal itt változtatjuk meg a változóterület mutatókat, és kiszámítjuk a szükséges címeket.

\$1184: JSR \$88C0 – Pedig már ennek alapján előretolja a memóriában a program végét sőt a változóterület tartalmát is, hogy "be tudjuk szűrni" az új sort.

**Végül berakjuk a sort a helyére. A változók így megmaradtak és az interpreter által szükségesnek tartott CLR-utasítást pedig kihagyjuk.**

\$11F0-tól a BASIC-vége, és a változóterület pointeret az új címekhez állítjuk.

\$1218-on kezdődik az az interpreter ciklus, amelynek tartalmát a hosszabb inputpufferrel módosítottuk. Itt elintéződik az is, hogy a beadott sort BASIC-sornak, vagy direkt parancsnak vegye-e. (Ha az első elhozott karakter szám, BASIC-sornak veszi. Ld. \$1240-ben a CHRGET rutin hozza el a karaktert.)

\$128B-\$12B1-ig a turbós LOAD paramétereinek a beolvasása és beállítása történik. Ha az egység szám nem hét, akkor az \$F050 eredeti LOAD rutinra ugratunk.

\$12B3-\$12E2-ig a szalagon levő program keresése és a fejbeolvasás történik. Ennek alapján fogjuk a töltésmódot a későbbiek folyamán a lehetőségek szerint szétválasztani. \$332-re kerül az a byte, amely a kimentett program jellemzőit tartalmazza.

\$12E4-\$133A-ig a módok beállítása itt történik meg.

\$133C-\$134D-ig pedig beolvassa a programot, majd a \$1605-ön levő nagy változórendező rutinra ugrik.

\$1353 – a hibás betöltés rutinjának startcíme.

\$1362 – A SAVE paraméterek vizsgálata történik. A SAVE eredeti vektorára (\$F1A4) csak nem hetes egység szám esetén adódik a vezérlés.

\$136F-\$141C-ig a paraméterek sze-

rinti SAVE-léshez szükséges változópointeret betesszük a fejlécbe, a file-jellemzővel és a file-névvel együtt.

\$145D – a SAVE vége, a rendszer normalizálása történik.

\$1460 – a LOAD rutin tényleges turbós LOAD-ja, a megfelelő hosszt betölti. \$D5-re kerül a fej vagy programrész között különbséget jelző típus adat.

\$14B5 – egy byte-ot olvas be a szalagról.

\$14BE – egy bitet olvas be a szalagról.

\$14EB – a SAVE turbós része.

\$15F1 – byte írása a szalagra.

\$152D – bit írása a szalagra.

\$15C9 – a string értékeknek a program végéhez fűzése.

\$1605 – itt történik a LOAD lezárásához szükséges szelektálás, hogy beállítsa-e a változómutatókat vagy nincsenek azok betöltve, AUTORUN gépi kódból vagy BASIC GOTO sorszám legyen-e, stb. Itt tesszük a string értékeket is vissza a memória végébe, ha voltak. (\$1649-\$166B-rutin)

**A BASIC "GOTO sorszám" gépi kódú indítása a \$16BB címtől kezdődik. A sorszámot a fejlécben a \$356-\$357 tárolja, majd a \$8D53 és a \$8BDC rutinok indítják el.**

\$16E0 – a CLR utasítás módosított végrehajtása történik. Különbséget kell tenni egy numerikus, egy string és egy tömbváltozó törlésénél, továbbá a meküresedett helyre össze kell húzni a maradék változókat. Vigyázni kellett arra is, hogy A\$-t ne töröljük ki az A% helyett és fordítva. Ezért minden változót a maga helyén kell megkeresni is. A változók közötti vesszővizsgálatot a \$1769-es címtől kezdődően végezzük.

\$1832-ben van a stringpointer és hosszok beállítását végző rutin, amely nagyon fontos, mivel a stringváltozó és értéke egymáshoz oda van láncolva és ha ez megbomlik, akkor az egész rendszer megbomolhat.

Megjegyzés: A rendszer nincs felkészülve grafika használatához, mivel benyúlik a bittérképes területbe. Ha azonban átírjuk (bővített 16-oson vagy Plusz 4-en) \$4000-től, akkor a grafika használatának semmi akadálya.

TURBOBASIC

TURBOBASIC



## SPRITE BÁRMIKOR

Az alábbi gépi kódú program először is definiál egy sprite-ot, az egyszerűség kedvéért a jól ismert Commodore léghajót, majd azt megmozgatja. Ez eddig igazán nem nagy dolog. Ami izgalmasabb, hogy a mozgást az IRQ rutinokkal végezteti, ami azt jelenti, hogy a számítógép egyéb működésétől szinte függetlenül bármikor a képernyőn lehet a sprite. Programírás közben ugyanúgy, mint bármilyen más BASIC program futtatása közben a léghajó rója köreit a képernyőn. Így a rutin nagyon jól használható más programok színesítő betéteként. Persze van néhány olyan gépi funkció, amelyek szintén az IRQ rutinokat használják (pl. lemezkezelés), amikor a léghajónk egy kicsit dőcög, esetleg megáll, majd újra indul.

A program 20-as sorától definiálható a sprite, 6. sorától pedig a kiegészítő adatokat lehet változtatni. A sprite útvonalának átírásához már a gépi kódú részt kell megfejteni.



Amint látható e havi számunkban is többségében külföldi programok szerepelnek. Ennek oka az, hogy olvasóink még nem indultak be igazán, alig kaptunk itthoni programötleteket, leírásokat. Hogy a jövőben többen kapjanak kedvet arra, hogy közkinccsé tegyék ötleteiket közöljük a lap feltételeit.

1. Minden leközlött programért, programötletért, programrészért a lap honoráriumot fizet. Ezt nem kell nagyon komolyan venni, tehát a honorárium inkább jelképes, mint óriási. Nem tudunk annyit fizetni egy játékprogramért, mintha azt forgalmazná valamelyik cég, de annyit tudunk, hogy a készítője legalább ne szégyenkezzen. Az egyébként eladhatatlan programötletekért pedig lényegesen többet fizetünk, annál, mintha senki sem venné meg.

Tájékoztató ár: kb. 50 soros BASIC program 300–500 Ft.  
2. A lapban közölt programok természetesen megrendelhetők a Pötyögő Szolgálattól, amiért külön honorárium nem jár.

3. A közléstől függetlenül a készítő oda adja el a programját, ahova akarja, csak kérjük, hogy ezt árulja el nekünk is, hogy ne érhessenek meglepetések (pl. több lapban jelenik meg egyszerre ugyanaz!).

4. Kérjük, hogy a programokat lehetőleg kazettán vagy

```

1 REM C= UJSAG SORSZAM 017
2 REM DATA WELT 85/4 90.OLD.
3 REM TIPP: FRANK RIEMENSCHNEIDER, HANNO
VER
4 REM SPRITE-MOZGATAS IRQ-VEKTORRAL
5
6 F=0:REM SZINBEALLITAS(0-15)
7 P=1:REM PRIORITAS (0=ELOTERBEN, 1 HATT
ERBEN)
9 X=1:REM X-IRANYU NAGYITAS (0=NINCS)
10 Y=1:REM Y-IRANYU NAGYITAS (0=NINCS)
17 REM
18 REM SPRITE ADATOK BEOLVASASA
19 REM
20 FOR I=0 TO 62:READ A:POKE 832+I,A:NEX
T
21 DATA 000,127,000,001,255,192,003,255,
224,003,231,224,007,217,240,007,223,240
23 DATA 007,217,240,003,231,224,003,255,
224,003,255,224,002,255,160,001,127,064
26 DATA 001,062,064,000,156,128,000,156,
128,000,073,000,000,073,000,000,062,000
28 DATA 000,062,000,000,062,000,000,028,
000
29 REM
30 REM IRQ-ADATOK BEOLVASASA
31 REM
99 FOR I=49152 TO 49437:READ A:S=S+A:POK
EI,A:NEXT
100 IF S<>37895 THEN PRINT"HIBAS ADATBEI
RAS!":END
101 SYS 49152,F,P,X,Y
102 DATA 169,013,141,254,007,169,064,141
,021,208,032,253,174,032,158,183,138,041
103 DATA 015,141,045,208,032,253,174,032
,158,183,138,201,000,208,006,141,027,208
104 DATA 076,044,192,169,064,141,027,208
,169,000,141,028,208,032,253,174,032,158
105 DATA 183,138,201,000,208,010,141,029
,208,169,064,133,252,076,079,192,169,064
106 DATA 141,029,208,169,040,133,252,032
,253,174,032,158,183,138,201,000,208,010
107 DATA 141,023,208,169,229,133,251,076
,109,192,169,064,141,023,208,169,208,133
108 DATA 251,169,024,141,012,208,169,000
,141,016,208,165,251,141,013,208,169,001
109 DATA 133,253,120,169,145,141,020,003
,169,192,141,021,003,169,000,133,254,088
110 DATA 096,165,254,201,000,240,011,201
,001,240,048,201,002,240,047,076,209,192
111 DATA 238,012,208,173,012,208,197,252
,208,014,173,016,208,201,064,208,021,169
112 DATA 001,133,254,076,049,234,201,255
,208,010,169,000,141,012,208,169,064,141
113 DATA 016,208,076,049,234,076,212,192
,076,229,192,076,012,193,206,013,208,173
114 DATA 013,208,201,050,208,234,169,002
,133,254,076,200,192,206,012,208,173,012
115 DATA 208,201,024,208,010,173,016,208
,201,000,240,015,076,200,192,201,000,208
116 DATA 203,169,000,141,016,208,076,200
,192,169,003,133,254,076,200,192,238,013
117 DATA 208,173,013,208,197,251,208,178
,169,000,133,254,076,200,192,000,000,000

```

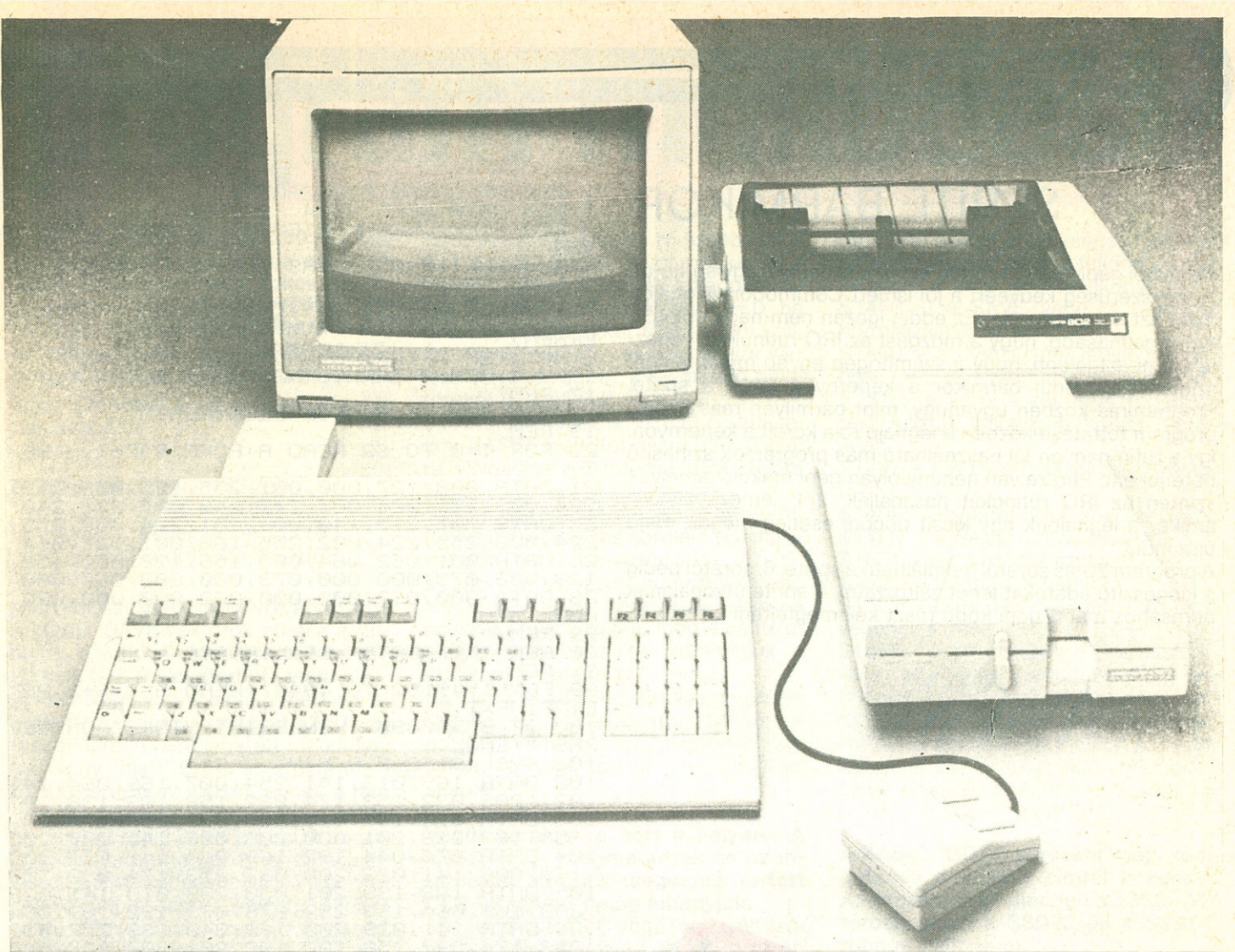
lemezen küldjük, ezek visszaküldését vállaljuk. Minden program mellé kérünk legalább néhány soros leírást, használati utasítást. Akinek módja van rá, kérjük printelt listát is mellékeljen, az sohasem árt.

5. Bár tudjuk, hogy nagyon előkelő, mégsem szeretjük az angol nyelvű feliratokat a programokban. Mint látható, még a külföldiekben is lefordítjuk ezeket. Kérjük, hogy ettől a felesleges munkától kíméljenek minket.

6. Általában a listákban bizonyos változásokat eszközölünk elsősorban az érthetőség, áttekinthetőség kedvéért. Ezt minden szerző tekintse természetesnek.

**Úgy gondoljuk, hogy ezek a feltételek elfogadhatóak, és reméljük, hogy szeptemberben már mozdulni sem tudunk majd a ránk zúduló programtömegettől.**





# COMMODORE

# 128

PERSONAL COMPUTER

*Egy éve jelent meg az európai piacon a C 128-as, ma már hazánkban is lehet belőle nélfány száz darab. Sajnos hivatalos behozatalra nem került, így a terjedése meglehetősen korlátozott és részben ennek következménye, hogy a magyar nyelvű irodalom illetve a szoftverellátás lényegében a nullával egyenlő. Lapunkban gyakran fogunk a 128-assal foglalkozni, részben hiánypótlóként, részben pedig, mert okos gépnek tűnik. Sajnos a világpiac a mi megelőlegezett bizalmunkat még nem igazán erősítette meg. Bár az eladási listák előkelő helyét foglalja el a gép, mégis sokan úgy érzik, hogy ez a konstrukció "se hús, se hal". Akik játszani akarnak, azoknak a 128-as drága, akik viszont másra kívánják használni, azok komolytalannak érzik. Pedig ez a gép éppen arra való, amire nálunk sokszor a nagyobb, IBM méretű gépeket használjk. Többször találkoztunk már azzal, hogy komoly IBM kompatibilis gépeken óriási adatsorokat gépelnek be, sőt primitív játékokat játszanak. Mert természetes, hogy aki a gép mellett dolgozik, az néha játszana is. Hát hiszen éppen erre való a C 128-as.*

A Commodore 128-as gép három üzemmódban használható:

128-as üzem – ez a gép bekapcsolás alapállapota

64-es üzem – ebben az állapotban teljesen kompatibilis a Commodore 64-essel, minden C 64-re készült program korlátozás nélkül futtatható rajta.

CP/M üzem – lehetőséget nyújt a CP/M

3.0 és a CP/M 2.2 rendszerben irt programok futtatására. Ebben az állapotban a gép Z80-as mikroprocesszorral dolgozik.

A 128-AS MÓDSZER

128-as üzemmódban 128 Kbyte írható-olvasható (RAM) memória használható, amely 512 Kbyte-ra is kibővíthető. Le-

hetőség van a programok gyors (kétszeres sebességű) futtatására, valamint a 40 karakter/sor felbontású képernyőn kívül egy további 80 karakteres képernyő használatára. A Commodore 128-as a BASIC 7.0-ás interpretert használja, amely a C 64 67 BASIC utasításával szemben 140 utasítással rendelkezik. A gép tartalmaz egy moni-



tor programot, amellyel a gépi kódú programok írása egyszerűbbé válik.

#### BILLENTYŰZET

A Commodore 128-as készítői különös figyelmet fordítottak arra, hogy a gép használatát kényelmessé, egyszerűvé tegyék. Erre szolgál a nemzetközi szabvány szerinti lapos klaviatúra, a különválasztott tizes billentyűzet, a speciális és az előre programozott funkciógombok, a megduplázott kurzor-mozgató billentyűk, valamint a gép sok szolgáltatást nyújtó képernyő editora is.

#### AZ ÚJ FLOPPY

A Commodore 128-as géphez készült az új VC 1571 típusú hajlékonylemezes meghajtó, amely a géphez hasonlóan három üzemmódban tud működni. Használható 1541-es meghajtóként (C 64-es üzemmódban), normál állapotában azonban alkalmas a mágneslemez mindkét oldalának egyidejű írására-olvasására, így a lemezek tárolókapacitása 340 Kbyte-ra növekszik. Az új meghajtó CP/M üzemmódban más számítógéptípusok által használt, különböző lemezformátumok szerint is tud dolgozni. (Például Commodore PC 10-PC 20; IBM PC; Osborne; Kaypro stb.) 128-as és CP/M üzemben a 1571-es meghajtó használatával a beolvasási sebesség a lemeztől több mint négyszeresére növekszik.

#### A 64-ES MÓDSZER

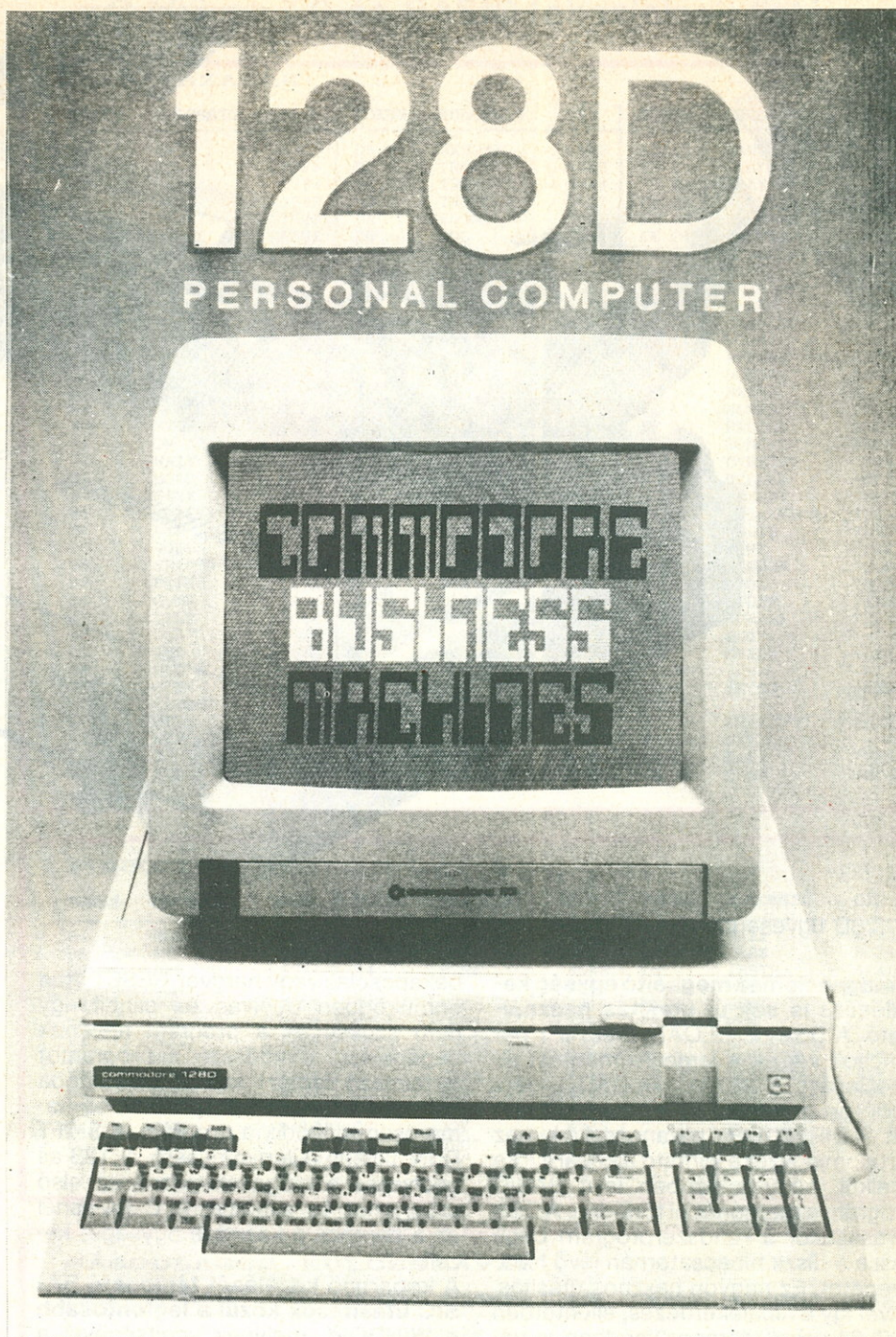
64-es üzemmódban is van lehetőség a 128-as üzem többletszolgáltatásai közül néhány használatára. Megnövelhető a processzor működési sebessége a kétszeresére (2 MHz-es órajel), bekapcsolható az elkülönített tizes billentyűzet és az új kurzor-blokk, vezérelhető a 80 karakteres képernyő, a mágneslemez egység használható kétoldalas lemezkezeléssel és megnövelt beolvasási sebességgel. Ezek a bővítések természetesen a C 64-es rendszerprogramjának módosításával, de kizárólag softver-úton megvalósíthatók.

#### PROGRAMOZÁS

**A Commodore 128-as programozásához nagy segítséget jelent a C 64 ismerete, mivel a gép felépítése, rendszerprogramja nagyon hasonlít a 64-eséhez, követi a „Commodore stílust”, egyesíti a C 64 és a C 16 jó tulajdonságait.**

A 128-as egyik erőssége a Commodore gépekhez használt eddigi legjobb BASIC nyelv. Ez már tudja kezelni a grafikus képernyőt, a hanggenerátort, a sprite-okat és a programírást megkönnyítő parancsokkal is rendelkezik.

Az AUTO paranccsal bekapcsolható az automatikus sorszámzás, a RENUMBER a programok újrászámzását végzi. A DELETE paranccsal tetszőleges sorok törölhetők. Lehetséges a program futása során a nyomkövetés – TRACE, illetve hiba esetén a hibás utasítás kiírása a képernyőre – HELP.



A C 128D minden lényeges dologban megegyezik a 128-assal, csak a kialakítása más.

A 1571-es floppy nagyobb dobozba került,

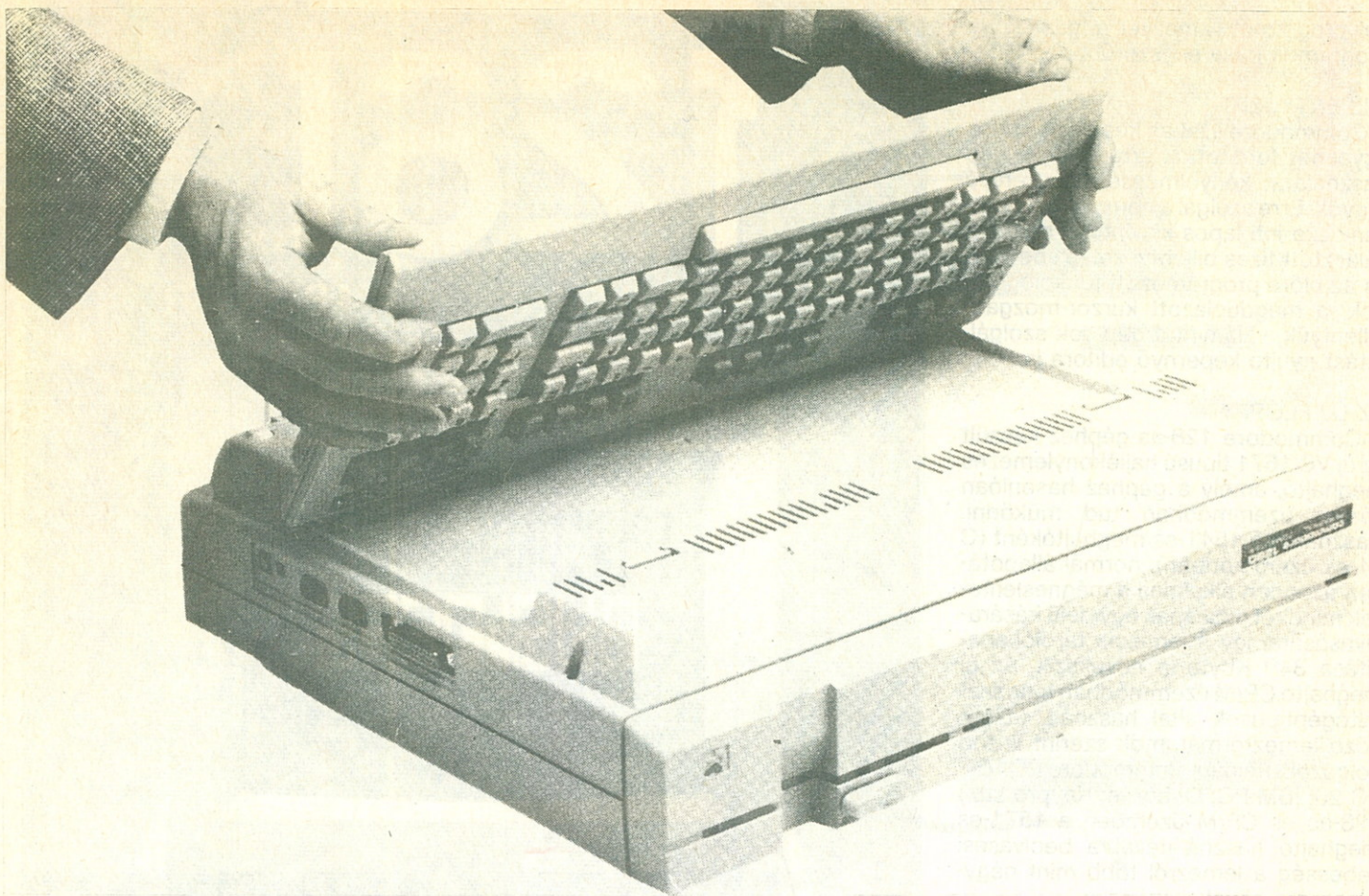
és ebben van a számítógép is. A billentyűzet laposabb kivitelű, és egy szellemes megoldással a floppy aljára csatlakoztatható

**Haladó programozók számára nyújt sok lehetőséget a strukturált programozást segítő ELSE utasítás, a BEGIN – BEND utasításcsoport és a ciklusszervezéseket könnyítő DO/LOOP/WHILE UNTIL/EXIT utasításcsoport. Nagyon jól használható a gép hibaikezelése. A TRAP utasítással megadható egy programsor, ahol bármilyen hiba esetén a program futása folytatódik. A hiba észlelése után az EL változó annak a sornak a számát tartalmazza, ahol a hiba található, az**

ER pedig a hiba kódját. Az ERR\$(ER) tömbben található a hibakódoknak megfelelő hibaikezelések. A RESUME utasítással lehet a hibakiértékelés után a program futását folytatni. A C 128 hibakezelési utasításainak felhasználásával sok programozási trükk valósítható meg. Lehetővé válik például a programból a RUN/STOP billentyű lekérdezése, lenyomása esetén tetszőleges programrész végrehajtása (lásd: a példaprogramokat).

**A Commodore 128-as grafikus utasításai hasonlóak a SUPERGRAFIK és a SIMON'S BASIC utasításaihoz.** Külön említésre érdemes azonban az SPRDEF utasítás, melynek hatására a képernyőn egy sprite-editor jelenik meg. Így könnyen elvégezhető a sprite-ok definiálása, javítása. Az elkészített sprite-ok BASIC változóknak tárolhatók.





A 128D ügyesen elrejthető billentyűzete

**A mágneslemez meghajtó egység kezelésére is sok új utasítás használható.** A DSAVE, DLOAD, DVERIFY parancsok végzik a lemezre mentést és beolvasást, a BSAVE, BLOAD a gépi kódú programrészletek írását, olvasását. A DIRECTORY parancssal a lemez tartalomjegyzéke kiírható a képernyőre anélkül, hogy a gépben lévő BASIC program a memóriából törlődne. A DS\$ olvasásakor a rendszerprogram beolvassa a diszk hibacsatornán lévő hibaüzenetet. Ez nagyon hasznos utasítás, mivel így a hibakérdezés, ellentétben a C 64-essel, parancs módban is véreghajtható.

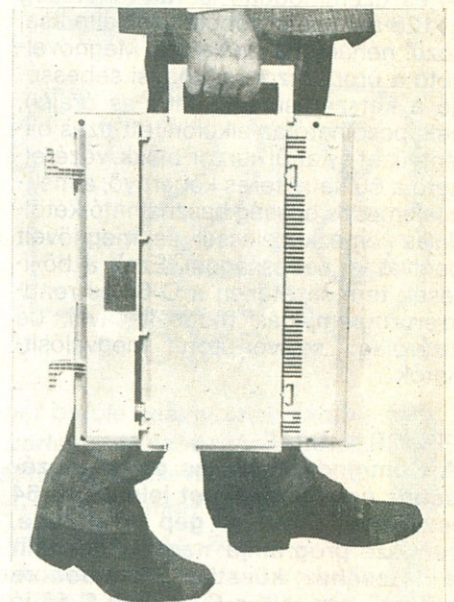
**Több lehetőség kínálkozik a programok beolvasására és automatikus elindítására.** A RUN „programnév” parancsra a megadott programot beolvassa a gép a mágneslemezzel és elindítja. Minden mágneslemezen kijelölhető egy program, amelyet a C 128-as

bekapcsolás vagy hardver-RESET után önműködően beolvas és elindít. Így nem szükséges a program nevének begépelése, mindössze a programot tartalmazó lemezt kell a meghajtóba helyezni, a gépet bekapcsolni, és hamarosan működik a program. A SHIFT RUN billentyű lenyomására a 128-as beolvassa a mágneslemezzel az első programot és elindítja. A C 64-esnél ez a parancs a kazettás egységet kezeli.

**A képernyő kezelését támogató BASIC utasítások közül a legfontosabb a WINDOW,** melynek segítségével a képernyő ablakokat lehet létrehozni.

**A programok begépelését segítik a képernyő editor utasítások.** Lehetséges automatikus insert mód használata, a kurzor sor elejére és végére állítása, tabulátorpozíciók megadása, új sorok beszúrása, sorok elejének, ill. végének törlése, a képernyő fel- és

A 128D – aktatászkaként



```

10 REM *****
20 REM *
30 REM *      FLOPPY      FIGYELES
40 REM *
50 REM *****
60 TRAP 1000 : REM      HIBA ESETEN UGR
AS AZ 1000-ES SORRA
100 DIRECTORY
200 IF DS <> 0 THEN PRINT"HIBA A FLO
PPYVAL !" : GETKEY A# : GOTO 60
900 END
1000 TRAP 1000 : IF ER <> 5 THEN 120
0
1100 PRINT"KAPCSOLJA BE A FLOPPYT !"
: GETKEY A#
1200 RESUME 60

```

```

10 REM *****
20 REM *
30 REM *      RUN/STOP FIGYELES
40 REM *
50 REM *****
60 TRAP 1000 : REM      HIBA ESETEN UGR
AS AZ 1000-ES SORRA
100 PRINT ". ";
200 :
300 :
900 GOTO 100
1000 TRAP 1000 : IF ER <> 30 THEN 12
00
1100 PRINT : PRINT "< RUN/STOP >"
1200 RESUME 60

```



## C 128-as 7.0-ás BASIC tokenjeinek listája

lefelé való görgetése. Az egyik legfontosabb funkció az idézőjeles módus kikapcsolása, melynek programlisták idézőjelek közötti szövegeinek javításakor van nagy jelentősége. Programok listázásakor célszerű a NO SCROLL billentyű használata, melynek lenyomásával a listázás megállítható, majd újbóli lenyomásával folytatható. A listázás lassítása a C 64-essel ellentétben a C= gomb lenyomásával érhető el.

**A Commodore 128 egyidejűleg két képernyőt tud kezelni. A 40 karakteres grafikus és a 80 karakteres képernyő között is egyszerű utasítással lehet mozogni, kiválasztani, hogy melyik képernyőre kívánunk írni. A 80 karakteres képernyőn lehetőség van a kurzor formájának megválasztására, a karakterek aláhúzására és villogtatására. A C 128-as két karakterkészlettel, egy vastag és egy vékony rajzolatúval rendelkezik, melyek az ASCII/DIN kapcsoló állásától függően kapcsolódnak be. A két karakterkészlet 64-es üzemmódban is használható.**

## C 128 műszaki adatok

### 128-AS ÜZEMMÓD

- 8502 mikroprocesszor (6510/6502 kompatibilis)
- 1 vagy 2 MHz órajel
- 6581 hang-chip
- 128 KByte RAM, bővíthető 512 KB-ig
- 48 + 16 KByte ROM
- BASIC 7.0 fejlesztett verzió
- beépített monitor program
- 40 karakter 25 sorban (320x200 képpont) vagy 80 karakter 25 sorban (640x200 képpont)
- 16 szín és 8 definiálható sprite (csak a 40 oszlopos felbontásban)

### CP/M ÜZEMMÓD

- Z 80 mikroprocesszor
- 2,04 MHz órajel
- CP/M Plus 3.0 verzió
- 128 KByte RAM, bővíthető 512 KB-ig
- 40 karakter 25 sorban (320x200 képpont) vagy 80 karakter 24 sorban (640x200 képpont)
- 16 szín

### KULCSSZAVAK

TOKEN	KULCSSZÓ	CÍM	TOKEN	KULCSSZÓ	CÍM	TOKEN	KULCSSZÓ	CÍM
80	END	4BCD	D6	RESUME	5F62	FE 02	BANK	6BC9
81	FOR	50F9	D7	TRAP	5F4D	FE 03	FILTER	7046
82	NEXT	57F4	D8	TRON	58B4	FE 04	PLAY	6DE1
83	DATA	528F	D9	TROFF	58B7	FE 05	TEMPO	6F07
84	INPUT#	5648	DA	SOUND	71EC	FE 06	MOVSPR	6CC6
85	INPUT	5662	DB	VOL	71C5	FE 07	SPRITE	6C4F
86	DIM	587B	DC	AUTO	5975	FE 08	SPRCOLOR	7190
87	READ	56A9	DD	PUDEF	5F34	FE 09	RREG	58BD
88	LET	53C6	DE	GRAPHIC	6B5A	FE 0A	ENVELOPE	70C1
89	GOTO	59DB	DF	PAINT	6186	FE 0B	SLEEP	6BD7
8A	RUN	5A9B	E0	CHAR	67D7	FE 0C	CATALOG	A07E
8B	IF	52C5	E1	BOX	62B7	FE 0D	DOPEN	A11D
8C	RESTORE	5ACA	E2	CIRCLE	668E	FE 0E	APPEND	A134
8D	GOSUB	59CF	E3	GSHAPE	658D	FE 0F	DCLOSE	A16F
8E	RETURN	5262	E4	SSHAPE	642B	FE 10	BSAVE	A1C8
8F	REM	529D	E5	DRAW	6797	FE 11	BLOAD	A218
90	STOP	4BCB	E6	LOCATE	6955	FE 12	RECORD	A2D7
91	ON	53A3	E7	COLOR	69E2	FE 13	CONCAT	A362
92	WAIT	6C2D	E8	SCNCLR	6A79	FE 14	DVERIFY	A1A4
93	LOAD	912C	E9	SCALE	6960	FE 15	DCLEAR	A332
94	SAVE	9112	EA	HELP	5986	FE 16	SPRSV	76EC
95	VERIFY	9129	EB	DO	5FE0	FE 17	COLLISION	7164
96	DEF	84FA	EC	LOOP	608A	FE 18	BEGIN	769C
97	POKE	80E5	ED	EXIT	6039	FE 19	BEND	528F
98	PRINT#	553A	EE	DIRECTORY	A07E	FE 1A	WINDOW	72CC
99	PRINT	555A	EF	DSAVE	A18C	FE 1B	BOOT	7335
9A	CONT	5A60	F0	DLOAD	A1A7	FE 1C	WIDTH	71B6
9B	LIST	50E2	F1	HEADER	A267	FE 1D	SPRDEF	7372
9C	CLR	51F8	F2	SCRATCH	A2A1	FE 1E	QUIT	-
9D	CMD	5540	F3	SELECT	A32F	FE 1F	STASH	AA1F
9E	SYS	5885	F4	COPY	A346	FE 20	-	-
9F	OPEN	918D	F5	RENAME	A36E	FE 21	FETCH	AA24
A0	CLOSE	919A	F6	BACKUP	A37C	FE 22	-	-
A1	GET	5612	F7	DELETE	5E87	FE 23	SWAP	AA29
A2	NEW	51D6	F8	RENUMBER	5AF8	FE 24	OFF	-
CB	GO	5A3D	F9	KEY	610A	FE 25	FAST	77B3
D5	ELSE	5391	FA	MONITOR	B000	FE 26	SLOW	77C4

### FUNKCIÓK

TOKEN	KULCSSZÓ	CÍM	TOKEN	KULCSSZÓ	CÍM	TOKEN	KULCSSZÓ	CÍM
B4	SGN	8C65	C2	PEEK	8005	D2	HEX\$	8142
B5	INT	8CFB	C3	LEN	8668	D3	ERR\$	80F6
B6	ABS	8C84	C4	STR\$	85AE	D4	INSTR	824D
B7	USR	1218	C5	VAL	804A	CE 02	POT	837C
B8	FRE	8000	C6	ASC	8677	CE 03	BUMP	82AE
B9	POS	94D0	C7	CHR\$	85BF	CE 04	PEN	8397
BA	SOR	8FB7	C8	LEFT\$	85D6	CE 05	RSPPPOS	831E
BB	RND	8434	C9	RIGHT\$	860C	CE 06	RSPRITE	8361
BC	LOG	89CA	CA	MID\$	861C	CE 07	RSPCOLOR	83E1
BD	EXP	9033	CC	RGR	8182	CE 08	XOR	8407
BE	COS	9409	CD	RCLR	819B	CE 09	RWINDOW	82FA
BF	SIN	9410	CF	JOY	8203	CE 0A	POINTER	477A
CO	TAN	9459	DO	RDOT	9B0C			
C1	ATN	94B3	D1	DEC	8076			

### MÁSODLAGOS KULCSSZAVAK

TOKEN	KULCSSZÓ	CÍM	TOKEN	KULCSSZÓ	CÍM	TOKEN	KULCSSZÓ	CÍM
A3	TAB(	-	A7	THEN	-	FC	UNTIL	-
A4	TO	-	A8	NOT	-	FD	WHILE	-
A5	FN	-	A9	STEP	-			
A6	SPC(	-	AA	USING	-			

### ARITMETIKAI OPERÁTOROK

TOKEN	KULCSSZÓ	CÍM	TOKEN	KULCSSZÓ	CÍM	TOKEN	KULCSSZÓ	CÍM
AA	+	8848	AC	*	8A27	AE	↑	8FC1
AB	-	8831	AD	/	8B4C			

### LOGIKAI OPERÁTOROK

TOKEN	KULCSSZÓ	CÍM	TOKEN	KULCSSZÓ	CÍM
AF	AND	4C89	BO	OR	4C86

### ÖSSZEHASONLÍTÓ OPERÁTOROK

TOKEN	KULCSSZÓ	CÍM	TOKEN	KULCSSZÓ	CÍM	TOKEN	KULCSSZÓ	CÍM
B1	>	-	B2	=	-	B3	<	-

Következő számtól elkezdjük közölni a C 128-as memóriatérképét, RAM listáját. A vállalkozás kimenetele még kérdéses, mert a lista bizonyos részeivel rendelkezünk, a teljes megszerzéséhez azonban C 128 tulajdonok segítségével lenne szükségünk. Ha esetleg Ön tud segíteni, kérjük jelentkezzen!



Természetesen mindig jól esik egy szerkesztőségnek, ha a lapot dicsérik, úgyhogy a jövőben is szívesen fogadunk minden ilyen témájú levelet. Ígérjük azonban, hogy ezeket többé nem fogjuk közölni. És egyben ígérjük azt is,

hogy a minket szidó leveleket mindig közölni fogjuk. Minthogy az első lapszámról van szó, most az egyszer kivételt teszünk.

Örömmel vettem kézbe a Commodore Újság első számát. Az oldalakat olvasva egyre nőtt az örööm. Ilyen magas színvonalú, érdekes magyar újság még nem volt a kezemben.

Különösen nagyot dobbant a szívem, mikor a "Plusz még mi jöhet?" című cikket olvastam. Itt Rényi Gábor, a Novotrade RT. ügyvezető igazgatója a következőket nyilatkozta: „ – Összesen 10000 darabra szól a behozatali engedélyünk, ebből 6500-at kap a Novotrade, amelynek egy részét megrendelésre iskoláknak, oktatási intézményeknek adjuk, **egy részét pedig a 2C boltokban a vásárlók kapják.**” (Kiemelés tőlem.) Kardos András, az ÁPISZ vezérigazgatója pedig a következőket mondta:

„ – A tavalyi akciónk után – reménykedve a jövőben – előrendeléseket vettünk fel C 16-osokra. Amennyiben mindenki elfogadja a Plusz 4-est, ezen az áron, úgy valamennyi gépre megvan a vevőnk, sajnos egyetlen új megrendelést sem tudunk elfogadni.”

E mondatokat jól megjegyezve indultam az első adandó alkalommal, augusztus 11-én előbb a Balzac utca 35. be, majd a Budafoki út 7. be, azaz a 2C, illetve az ÁPISZ-SZÁMALK boltba. A 2C-ben az udvarias pultőrő nem is engedte végigmondanom eleve reménytelen kérdésemet, kategorikusan kinyilatkoztatta – nyilván mint ostoba „vásárlónak” – a Novotrade magánszemélynek soha nem adott el egy Plusz 4-est sem, és nem is fog. Leforrázva, de törve nem, indultam az ÁPISZ üzletébe, hiszen egy előjegyzés boldog tulajdonosa vagyok. Talán tudnak valami biztatót mondani.

Leszállva a 86-os buszról magabiztosan nyitottam ajtót. Büszkén elővettem előjegyzési papiromat, melyet az elárusító hölgy tüzetesen elolvasott. Magabiztosságom pillanatok alatt szertefoszlott. Itt sem jártam szerencsével. Egyetlen vigaszom csak az maradt, hogy a bolt alkalmazottjának is újság volt az újságom, mivel a csodálkozására elővett fent nevezett cikkből ő is megtudhatta a vállalati pletykák újabb fejleményét. Így már felkészülve várhatja sorstársaimat.

Felhajtás volt a hír, vagy csupán egy rossz időzítés? Egy dolog biztosnak látszik: saját dolgozóikat nem készíteték fel egyik helyen sem a várható érdeklődésre.

Más...

Kíváncsian kerestem az eszlő számban a beigért 50 Ft-os vásárlási tikkett, melyet az Ötlet 21. számában a Commodore Egyesület toborzójában dőlt betűvel hirdettek. Még szerencse, hogy nem ezért lettem az Egyesület tagja.

### Szabados Pál

Az első témával kapcsolatban: úgy tűnik Ön volt a legszerencsétlenebb, az úttörő, aki elsőként kérdezte végig a boltokat, és így a legkevésbé határozott választ kapta. Az önt követők már a kitaposott ösvényen kapták a továbbra is határozatlan választ. Jó hírekkel mi sem szolgálhatunk, de az érdekelt vállalatok igazgatóinak magyarázatát a 11. oldalon olvashatja. A második témával kap-

csolatban: reméljük, hogy más sem csak a beigért 50,- forintos vásárlási utalvány miatt lett az egyesület tagja. A tény azonban tény, a beigért kedvezmények nem örökérvényűek, hogy egészen pontosan fogalmazzunk, havonta változnak. Úgy tűnik – utolsó oldalunkon Ön is olvashatja – ebben a hónapban nem lehet panasza senkinek. Biztosak vagyunk azonban abban, hogy lesznek szűkebb hónapok is.

## PÖTYÖGŐ SZOLGÁLAT

SORSZÁM	GÉPTÍPUS	ELNEVEZÉS	ÁRA
001	C 64	GOTO X Databecker	10,-
002	C 64	GOTO X Data Welt	10,-
003	C 64	Hangzó billentyűk	10,-
004	C 64	Memória kiírás	10,-
005	C 16	Rajzoló program	30,-
006	C 64	Átsorszámzó program	10,-
007	C 64	Help-trace	10,-
008	C 16	Hangmemória	40,-
009	C 64	Teke Homecomputer	20,-
010	C 64	Csak kezdőknek!	10,-
011	C 64	Billentyűkódok	10,-
012	C 64	Autonumber	20,-
013	C 16	BASIC bővítő	20,-
014	C 64	BASIC billentyűzet	20,-
016	C 64	Perifériavizsgálat	10,-
017	C 64	Sprite mozgítás IRQ	20,-
018	C 64	NO SCROLL	20,-
023	C 16	TURBOBASIC	50,-

Alulírott megrendelem a következő programokat a Pötyögő Szolgáltatótól:

PROGRAM SORSZÁMA	ÁRA
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

Összesen:                      db                      ,- Ft

- A programokat saját kazettámra kérem.
- A programokat saját lemezemre kérem.   
(a megfelelő kockát x-elje be!)
- A programokat a Szolgálat által biztosított Parrot SS, SD lemezre kérem (136,- Ft/db)
- 15' kazettára kérem. (26,- Ft/db)
- POSTA költség (25,- Ft)

Összesen:                      ,- Ft

A fenti összeget befizettem az Országos Commodore Egyesület számlájára:

OPT XIII. Visegrádi u. 7/b.

MNB 217-98292

OTP 565-3610

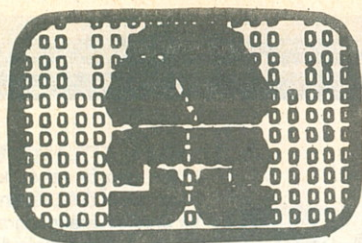
NÉV: \_\_\_\_\_

EGYESÜLETI TAGSÁGI SZÁM: \_\_\_\_\_

DÁTUM: \_\_\_\_\_

alíírás





STENCILFESTÉK · TUS *Apisz* TAPÉTA · HEGYEZŐ ·

VONALZÓ · ISKOLATÁSKA · DA...

**EXTRA KEDVEZMÉNY  
A COMMODORE EGYESÜLET  
TAGJAINAK!**

Szeptember hónapban az

**ÁPISZ "ÁSZ" SZAKÜZLETÉBEN  
VISSZAADJUK  
A VÁSÁROLT ÁRUK ÉRTÉKÉNEK  
10 %-at.**

**Kínálatunk:**

- számítástechnikai médiák, tartozékok,
  - mágneses háttértárak,  
(mágnesszalagok, merev és hajlékony mágneslemezek,  
rövid idejű magnókazetták),
  - floppytartó dobozok és mappák,
  - festékszalagok nyomtatókhoz és írógépekhez,
  - szoftver termékek,
  - játékprogramok,
  - tisztítóeszközök,
  - számítástechnikai könyvek,
  - speciális íróeszközök:  
fóliára író filcirónok, vonalzó, sablonok,  
és egyéb számítástechnikával kapcsolatos termékek.
- Szakmai tanácsadás.  
Mind a magánvásárlók, mind a vállalatok  
és intézmények rendelkezésére állunk.

**ÁPISZ ÁSZ  
SZÁMÍTÁSTECHNIKAI SZAKÜZLETE**

Budapest, XI., Budafoki út 7.

Telefon: 665-503

ENDŐ · INDIG

ITA *Apisz*

ESVONALZÓ

· SZINES IRI

Ö · ALBUM

IDŐ · INDIG

*Apisz*

VONALZÓ

SZINES IRI

GYEZŐ ·

Ö · INDIG

*Apisz*

VONALZÓ

NES IRC

ALBUM ·

LORENDŐ · MÁGN

KAJZSZÖG · KÖRZŐ · TINTA *Apisz*

KÉPESLAP · LYUKASZTÓ · FEJESVONALZÓ

YEGZŐ · MILTONKAPOCS · SZÁMOLÓGÉP · SZINES IRC

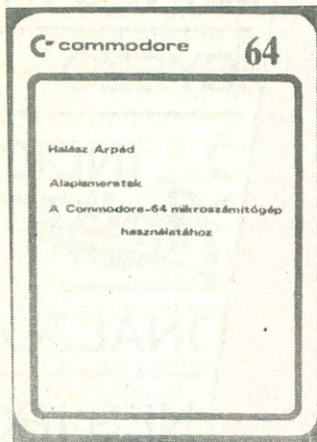
TA · STENCILFESTÉK *Apisz* HEGYEZŐ · ALBUM ·



A számítástechnikával foglalkozó hazai könyvek megjelenése meglehetősen késéssel követi a számítógépi piac változását – ebben persze bűnös a nyomdai átfutási idő is. A Commodore 64 géppel foglalkozó könyvek kínálata mostanában tetőzik, amikor már sokan elavultnak tartják a C-64-et, nagyobbra cserélik. A tény persze ettől még tény marad: a C-64 még mindig a legelterjedtebb géptípus, és ma is sokan ezen kezdik a programozás alapjainak megismerését. Ezért hasznosak, kellenek ezek a kötetek, – ha valóban hasznosak, vagyis jól használhatják, aki kíváncsi valamire a gép kezelésével kapcsolatban.

Az alábbi összeállításban azokból a kötetekből értékelünk néhányat, melyek az utóbbi hónapokban jelentek meg – frissen, utánnymásban vagy új kiadásban –, és amelyek szándékuk szerint a teljesen kezdőknek, a C-64-gyel és a programozással most ismerkedőknek szólnak.

## Halász Árpád: Alapismeretek a Commodore 64 mikroszámítógép használatához



Halász Árpád könyve, amely 1985 elején jelent meg először, rengeteg fontos ismeretet közöl a gépről és alapfokú programozásáról, annak ellenére, hogy terjedelme alig nagyobb, mint egy iskolai füzeté. Ehhez persze sokban hozzásegíti a mellékelt lemez a minta-programokkal. Első része logikusan felépítve mu-

tatja meg, gyakoroltatja a klaviatúra kezelését, a képernyőre való írást, a periféria-kezelést, a gép használatát direkt üzemmódban. A második rész a C-64 BASIC főbb utasításait ismerteti, jó magyarázatokkal – már amennyire a terjedelem engedi.

„Csupán” a szerző alapállása kérdőjelezhető meg: „Az eddigiek során azt igyekeztünk bemutatni, hogy ha valamit elkövetünk, akkor cselekedetünknek mi lesz a következménye, és nem azt, hogy miért pont az történik. A továbbiakban is ezt a módszert követjük, de javasoljuk az önálló próbálkozásokat is.” Nos, kétséges, hogyha valaki egy feladat megoldására programot akar írni, akkor ezt meg tudja tenni az utasítások működési módjának ismerete nélkül – illetve igen-csak sok „önálló próbálkozásra” lesz szüksége. Szerencsére a szerző nem tartja magát következetesen ehhez az alapelvhez – így az olvasónak mégis lehetősége van a BASIC alapjainak megismerésére. A kötet nyomdai tördelése nagyon jó, kiemeli a lényegét, hasznosak a bekezdéseket megelőző CSAK OLVASNI, GYAKOROLNI, TUDNI KELL, stb. megjegyzések.

## Pál Zsuzsanna–Révbíró Tamás: Hetedhét Commodore 64 Novotrade. 66,- Ft

A Hetedhét sorozat első darabja volt ez a könyv. Eredetileg három

különálló füzet formájában jelent meg, az új kiadás a füzetek összefü-



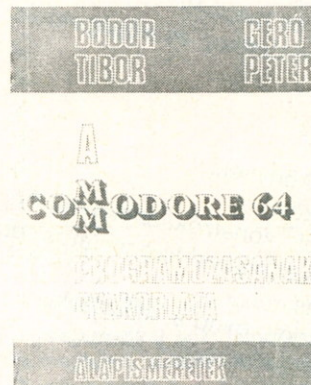
zött, javított változatát tartalmazza. A könyv célja a programozás iránti érdeklődés felkeltése, és e célkitűzésnek eleget is tesz. Ahogy a szerzőpáros az előszóban fogalmazza: a könyv arról szól, hogyan lehet a számítógéppel játszani. Já-

tékon persze azt értik, hogy a gép használója ne csupán a gyári, profi által írt játékprogramokat töltsse be, és a képernyő előtt vívjon úrháborút, hanem próbáljon maga is bepötyögni valamit a gépbe, ami lefuttatható, és látványos eredményt produkál.

A kétmondatos előszóhoz hasonlóan tömör – de ezzel együtt jól követhető, világos – a kötet érdemi része is. A 3x7 leckére bontott anyag egy-egy rövid témakört ölel fel, melyek ismeretei fél-háromnegyed óra alatt könnyen – tényleg játszva – elsajátíthatóak. Az olvasó a képernyőre való írástól eljuthat az egyszerű szubrutinokat tartalmazó programok megszerkesztéséig. Az érdeklődés ébrentartásához hozzájárulnak a jópofa illusztrációk, a tanulhatóságot pedig teljes, futtatásra kész programlisták és ellenőrző kérdések segítik.

A kellemes stílus, a logikus szerkezet jó esélyt ad arra, hogy aki túl jut a kötet anyagán, az nem áll meg a programozási tudásnak ezen a szintjén.

## Bodor Tibor–Geró Péter: A Commodore 64 programozásának gyakorlata Alapismeretek 1. kötet SZÁMALK, 55,- Ft.



A SZÁMALK C-64 sorozatának első kötete a szerzők szerint szintén azokhoz szól, akik nem rendelkeznek semmiféle előzetes számítástechnikai ismerettel. Az előszó fokozatosságot és didaktikus felépítést ígére. Nézzük, mi teljesül ebből!

Az ISMERKEDÉS című rész a gép összeállításának, bekapcsolásának leírása után máris egy FOR-NEXT ciklust gépelltet be az abból mit sem értő olvasóval, hogy ezen mutassa be a programlistázást, futtatást, törlést. Erre persze jó ez a program (mint bármely másik) – feltéve, hogy a billentyű-

zettel először találkozáva szintaktikai hiba nélkül be tudjuk vinni a gépbe. Ezután következnek a program elemeinek ismertetése, meglehetősen önkényes sorrendben. Félvezető fejezetcímek követik egymást: „Újítások” alatt találjuk például a sorszámozást, vagy a „Megjegyzések, kiegészítések, tanácsok” ismerteti a változó típusokat. Az elemi tevékenységeket természetesen a klasszikus másodfokú egyenlet illusztrálja.

A második rész témája az adattárolás. Ez jól követhető, világos, sok ábra illusztrálja – de mikor tanulunk meg programot írni, amit tárolhatunk?

Harmadik rész: A PROGRAMOZÁS ELEMELI. Végre! Az elcsigázott olvasó azt hiszi, hogy legalább itt megtud valamit a programozás mikéntjéről – de nagyot téved. Blokkdiagramok tömege, ami önmagában még nem volna baj, de kíséző szövegükben hemzsegnék a szekvenciális, feltételes és alternatív esetválasztó tevékenységek, elől- és hátultesztelő ciklusok és társaik. Ha a számítástechnikában teljesen járatlan olvasó nem csukná itt be végleg a könyvet, azon nagyon csodálkoznék. Pedig türelmének utolsó megmérése csak ezután következik: ömlesztve a C-64 függvényei, végezetül hét oldalon letárgyalva a tömbök kezelése. Egyben reménykedhetünk csak: hogy a sorozat további kötetei nem ebben a szellemben – illetve rendszertelenségben – fogják bemutatni tárgyukat.



**Dr. Ferenczy Antal:**

**C-64 Start!**

**Alapfokon a Commodore 64-ről**

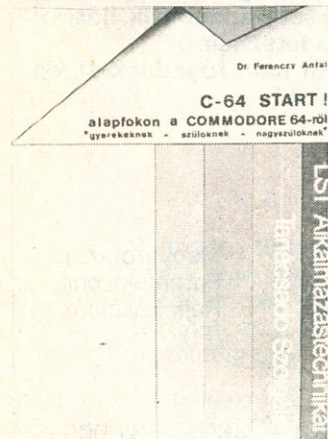
**gyerekeknek – szülőknek**

**– nagyszülőknek**

**LSI Alkalmazástechnikai**

**Tanácsadó Szolgálat.**

**136,- Ft.**



Az LSI ATSZ kiadványának szerzője szintén ígéretet tesz az előszóban. A hangvétel és a forma újdonságát ígéri: a hangvétel könnyedebb lesz a máshol megszokott, tudományosan precíz megfogalmazásoknál, a forma pedig a nyelvlecke-szerű feldolgozásban nyújt újat. Nos, a formai „újítás” nem a szerző találmánya, hiszen nem egy könyv építkezik hasonlóan, – ha nem is nevezi fejezeteit nyelvleckének. Bármely könyv, amely egy nyelvet tanít meg (legyen az akár programnyelv), egymásra épülő, az előzőek anyagát felhasználó fejezeteket kell, hogy tartalmazzon, különben használhatatlan. Ami pedig a hangvételt és a tartalmat illeti... Majd kiderül. Első fejezet: Felhasználói géphasználat. Jól használható információk gyűjteménye a gép üzembelhelyezéséről, ha az olvasó hajlandó átrágni magát az „aktuális konfiguráció” és hasonló fogalmak definíci-

ón. Ez lenne az új hangvétel? Köszönöm, inkább a régít. A második fejezet a „Számítástechnikai munkafolyamat” hangzatos címet viseli, a szöveg itt is hasonló stílusú. Pedáns(kodó) magyarázatok, pontokba szedett tennivalók – az olvasó elrémül a rá váró feladatoktól. Oldódásképpen elszórakozhat a „munkafolyamatot” illusztráló feladatokon. A szerző szerint azért van ezekből három, hogy mindenki ízlésének megfelelően választhasson közülük. Mindhárom egy téglalapot térfogatának kiszámítását tűzi ki célul, – először egy furdómedencéét, azután egy telekre terítendő földrétegét, majd egy lakás betonozásához szükséges betonét. Tényleg, melyiket szeresem?

A 3. fejezet – Röviden a C-64-ről. Persze, hasznos dolog a számrendszerek ismerete – de miért kell ezt az olvasó nyakába zúdítani akkor, amikor ennek semmi hasznát nem fogja venni az alapszintű BASIC programok írásakor? És miért kell tudnia a tokenekről, amikor még egy BASIC-utasítást sem ismer?

Végre elérkeztünk – mintegy hatvan oldal átrágása után a 4. fejezethez: „C-64 BASIC”. E fejezet – mely szerencsére egymagában hosszabb, mint az előzőek együttvéve – igyekszik mindent megtanítani a BASIC nyelvű programozásról. Felépítése jónak mondható, teljességre törekvése már kevésbé: a PRINT utasítás bemutatásától el akar jutni az adatállományok kezelésének megismertetéséig. E cél elérése egy többszáz oldalas programozási tankönyvnek is becsülete válna. Talán, ha a szerző nem

eresztette volna olyan bő levűen szabatosra az előző fejezeteket, több helye maradt volna a C-64

BASIC sajátosságainak leírására – és a laikus felhasználó is több szonnal forgathatná a kötetet.

**Ismerd meg a BASIC nyelvjárásait!**  
**Commodore 64,**  
**Commodore VIC 20, SHARP PC-1500**  
**szerkesztette: Kóhegyi János**  
**Műszaki Könyvkiadó**  
**65,- Ft.**



Műszaki Könyvkiadó

Koncepciója első pillantásra nagyon jónak tűnik: Donald Alcock méltán népszerű könyvének. – Ismerd meg a BASIC nyelvet! – ismereteit átültetni, igazítani a típusoként más-más BASIC-nyelvjárást értő gépekhez. Amit az Alcock-könyv formai megjelenéséből átvettek, az tényleg értékessé teszi a köteteket. Így a kézírásos megjelenítés – és az ebből adódó közvetlen hangneme –, vagy a sok grafikus elemet tartalmazó, jól megjegyezhető jelölésrendszer. Ami viszont nehezen forgathatóvá teszi e könyveket: Kóhegyiék me-

reven ragaszkodnak az Alcock-könyv szerkezetéhez is. Elképzelésük az, hogy az olvasó párhuzamosan lapozhatja a két kötetet – Alcockét és az adott gépet bemutató nyelvjárást; ők csak a gépnek az Alcock által leirtakról eltérő viselkedését mutatják be. Csakhogy ez ellen két komoly kifogás merül fel. Az egyik, hogy Alcock egy programozási nyelvet akar megismertetni, nem pedig valamely gép programozását. Így olyan alapvető dolgokra, mint például a gép bejelentkezése, csak nagyon későn kerül sor, amikor az olvasó már többé-kevésbé elsajátította a programírást. Hasonlóképpen, a PRINT utasítás is az Alcock-könyv vége felé szerepel (együtt a TAB-bal és a PRINT USING-gal, ami logikailag helytálló), de a kezdő felhasználó számára a képernyőre írás volna az első alkalom, hogy valamilyen „cselekvésre” rábírja gépét. Emiatt a sorozat teljesen kezdőnek nem ajánlható jó szívvel, – inkább azoknak, akik már konyítanak valamit a BASIC-hez. A másik kifogás szintén a szerkezettel kapcsolatos. A legtöbb elterjedt, és a sorozat kötetében bemutatott gép nyújt valamit hangban és grafikában, adatokat tárol, nyomtat, ... Mindez természetesen nem fért az Alcock-könyv kereteibe, hiszen egyes gépeknél ezek a szolgáltatások annyira eltérőek, hogy azokat nem lehet általánosan tárgyalni. Ennek köszönhetően a nyelvjárás-sorozat függelékei mind terjedelmesebbek, hogy ezeknek az információknak is helyet adhassanak. Így az olvasó valamilyen konkrét kérdésre keres választ, kénytelen lázasan előre-hátra lapozgatni a két könyvben. Az Alcock-könyv szerkezetének szabadabb átvétele, a függelék részeinek az alapszövegbe való szerkesztése használhatóbbá, jobb felépítésűvé tette volna a sorozatot.

Összegzésként a táblázat mutatja a könyvek összehasonlítását öt szempont alapján, egytől tízig pontozva. Átlagot nem számoltam, mert egyrészt az értékelés szubjektív, másrészt pedig – és ez a fontosabb – hogy a szempontokat minden olvasó tetszése szerint súlyozhassa.

**Halász Pál – Bodor – Ferenczi Kóhegyi**  
**Rév – Gerő**  
**bíró**

Laikusoknak szól-e	9	10	3	4	8
Felépítés, szerkesztés	9	9	4	5	7
Feldolgozott ismeretanyag	7	4	6	8	7
Stílus	8	10	6	3	10
Forma, megjelenítés	7	9	7	7	10
Ár	140 Ft	66 Ft	55 Ft	136 Ft	65 Ft

Tallér József







# CHIP

**szébb és szolidabb**



A házi-számítógép történelemben a C-64 sikere élő legendává vált. Egyedül 1985-ben több mint egy millió darab talált gazdára, és a közönség rajongása még mindig tart.

Az új modell fejlesztésének legfontosabb alapja a száz százalékos régi C-64 kompatibilitás. Ezért csak mechanikai változtatásokat alkalmaztak a használhatóság javítására. Az új C-64 bátyjára a C-128-ra hasonlít. A billentyűzetet közelebb hozták az áramköri laphoz, így a doboz magassága körülbelül a felére csökkent. A billentyűzet felépítése egyébként változatlan maradt.

A változtatások a gépen belül is megtalálhatók, a rosszul árnyékoló alumínium fólia helyett egy igazi fém árnyékolót használnak, mely a chippek hűtése

sére is kiválóan alkalmas. Így a gép jobban bírja a hosszú órákon át tartó igénybevételt. A C-64 I/O portjai a megszokott sorrendben megtalálhatók. A beépítésnek nagy hátránya, hogy az operációs rendszert tartalmazó ROM-okat nem tokozták, így esetleges cseréjük igen nehéz és veszélyes.

Mi történt a szoftver kompatibilitással? Ebben az esetben semmi ok az aggodalomra. Összehasonlítva a régi és új C-64-es ROM-okat, nincs semmi különbség.

De ezzel a Commodore fejlesztőinek módosításai nem értek véget. A rendszerfunkciók felhasználó-barátságossá tételéhez felhasználták a 16 bites PC-knél elterjedt GEM szoftvert.

A GEOS használatával az embernek most már nem az az érzése, hogy egy Commodore elé ül le, hanem az, hogy egy nagy teljesítményű rendszer elé. A C-64 minden funkcióját tartalmazó menütechnikával és külön rendszerüzenet ablakkal a diszkrét formattáláshoz például csak néhány joystick mozdulatra és gombnyomásra van szükség. A tartalomjegyzék elérése egy

képszimbólumon keresztül történik. A GEOS-sal az adateltérési sebesség az öt-hétszeresére növekszik, gyorsasága eléri a hyperload sebességét. A Geowrite olyan szövegszerkesztő program, mellyel direkt a grafikus memóriába írhatunk be. Így elérhető, hogy egy sorban különböző betűtípusokkal írhatunk, és egy aláhúzott szó pontosan úgy néz ki a képernyőn, mint ahogy a printer papírján fog. A rendszer külön szolgáltatása a „Preview” funkció. Ez azt jelenti, hogy a nyomtatásra kész oldalakat papír pazarlása nélkül tanulmányozhatjuk és esetleg még itt javíthatjuk.

Ezzel még nincs vége az újdonságoknak. A rendszer fejlesztői a GEOS-t ún. nyílt rendszernek alkották meg, azaz a szoftver fejlesztőknek nyitott a lehetőség a rendszer továbbfejlesztésére. Így a GEOS rendszer a Commodore számára egy új szabvány lehet. Az operációs rendszer külön is megvásárolható, így a régi C-64 modellekben is használható.

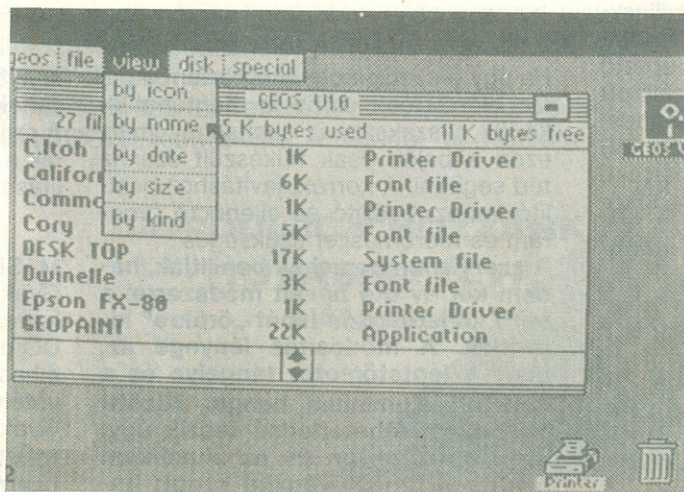
Chip, 1986. július

## GEOS C-64-re a Berkely Softworks-tól

A GEOS (Graphic Environment Operating System) a C-64-esek operációs rendszere, amely az új 64-eshez, a 64C-hez kapcsolódik, de külön is megvásárolható 60 dollárért.

A GEOS tulajdonságai hasonlóak az ATARI ST (GEM) és az Amiga (Intuition) operációs rendszeréhez. Bár jelenleg a pointer csak a joystick-kel mozgatható, de tervezik a fényceruzával és az egérrel való mozgatását is.

A GEOS-hoz tartozik még néhány igen hatékony programcsomag, a GEOS Write, GEOS Paint és egy tele-



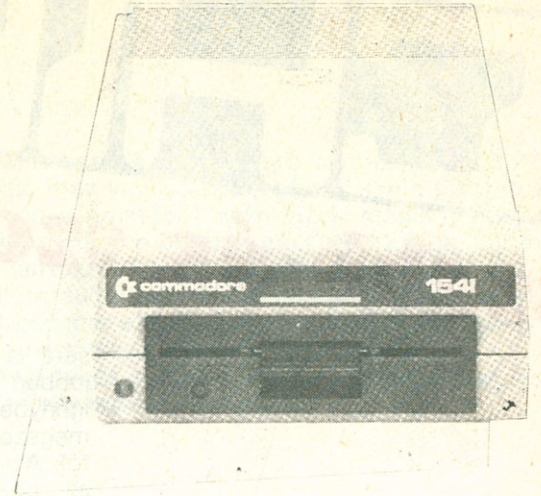
kommunikációs program a Q-Link. Ez utóbbi lehetőséget biztosít a Quantum-Link (az Egyesült Államokban működő) hálózat elérésére.

A nem GEOS programok is futhatnak GEOS alatt, de természetesen nem használhatják a GEOS előnyeit. A GEOS támogatja különböző printerek használatát, beleértve az Epson-t is. Úgy tűnik, hogy a GEOS a 64-esek standard operációs rendszerévé válhat.

Commodore Computing International,  
1986. augusztus



# FLOPPY -DOKTOR



A VC-1541 típusú floppy egység a COMMODORE C-64 személyi számítógép sokak által kedvelt és színdott mágneslemezes háttértárolója. A tároló elem egy állandó sebességgel forgó, cserélhető mágneslemez, ún. floppy-disk. A mágneslemezen levő információk leolvasására, ill. írására egy léptető motor által mozgatott kombinált író-olvasó fej szolgál. A léptető motor tengelyének körbefordulását egy fémszalag alakítja át lineáris mozgássá. A tengely órajárásnak megfelelő irányú elfordulása a kombinált író-olvasó fejet a mágneslemez közepétől kifelé, ellenkező irányú mozgása befelé mozgatja, miközben a mágneslemez állandó sebességgel forog. A léptető motor két helyzete közötti mozgása a kombinált író-olvasó fej kb. 0,5 mm-es elmozdulását eredményezi. Ebből következik, hogy az információt hordozó mágnespályák egymástól mért távolsága 0,5 mm.

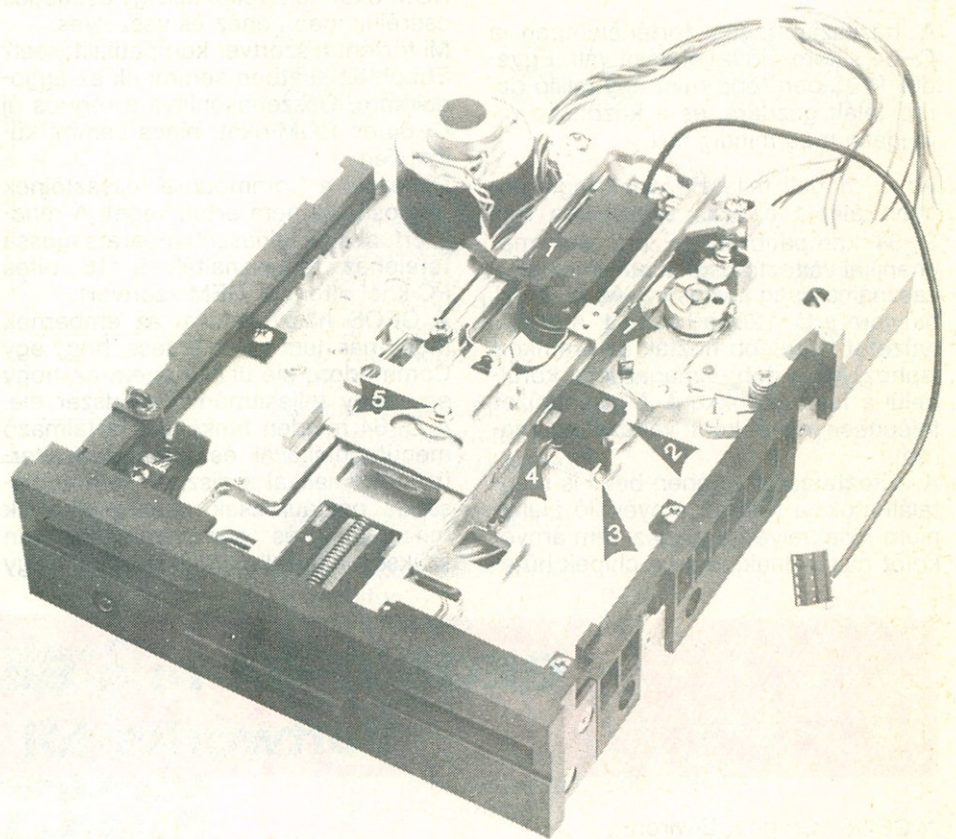
A kombinált fejet mozgató fémszalag a léptető motor tengelyéhez egy rásajtott alumínium henger segítségével kapcsolódik. Ezen alumínium henger kettős funkciót lát el. A fent leírtakon kívül további szerepe a mágnespályák azonosítása oly módon, hogy a kombinált fej szélső helyzetében az alumínium henger homloklapján kiképzett ütköző sík egy állítható és csavarral rögzített helyzetű lemezkéhez ütközik.

## ÁLTALÁBAN

A kiválóan működő egyszerű szerkezetnek óriási hiányossága az, hogy a tengely és a rásajtott henger az illesztés mértékétől függően el tud fordulni egymáshoz képest, így a léptető motor már nem a gyárilag beállított helyre viszi a fejet. Mivel a mágnespályák távolsága igen kicsi (0,5 mm) könnyű elképzelni, hogy a kismértékű elfordulás is zavarokat okoz a működésben.

Az elfordulás mértékétől függően bekövetkező hibák:

- Szekvenciális file-ok írásában, olvasásában bizonytalanságok lépnek fel.
- Jól beállított floppyval formatált lemezre a fej nem tud írni, (bizonytalanul olvassa)
- A floppyval régebben felvett lemezeket nem vagy nehezen olvassa.



Ha úgy érezzük, hogy megtörtént a baj (az előrelátóak előbb is megtehetik), kérjük a szakszerviz segítségét, mert ezen a bajon csak felkészült szerviz tud segíteni. A korrekt javításhoz beállító lemez, beállító és ellenőrző program és mérőműszer szükséges.

**A szervizben nemcsak beállítják, hanem két év óta bevált módszerrel a fenti konstrukciós hibát „örökre” kijavítják. A módosítás lényege az, hogy a léptetőmotor tengelye és a rásajtott alumínium henger közötti elfordulást lehetetlenné teszik úgy, hogy a tengelyen és az alumínium hengeren tengelyirányba vágott ho-**

**ronyba egy fémhuzaldarabot ragasztanak.**

A ragasztó megszilárdulása után történik a léptetőmotor és az ütköző beállítása gyári beállítólemez segítségével.

## MÁSKOR

Gyakran okoz üzemzavart a mágneslemezt körbeforgató kommutátoros DC motor meghibásodása is. A motor elhasználódása fokozatos, az áramfelvétele észrevétlenül emelkedni kezd. Gyakorlati tapasztalat, hogy 300 mA feletti áramfelvétel esetén motort kell cserélni. Az előzőekben leírt hibajelen-



séget (írás-olvasási bizonytalanságot) okozhat a hálózati egyenirányító diódák (Graetz) meghibásodása is. A meghibásodott, elhasználódott DC motor nagyobb áramfelvétele miatt gyakran felelős az egyébként is nehéz körülmények között dolgozó diódák háláláért. Ilyenkor mindkét alkatrészt cserélni kell.

**Itt kell még megemlíteni egy, az előzőekhez hasonló, rokon problémát. Az országba bekerült gépeknek jelentős hányada 240 V-os készülék. A gép hátoldalán az adatlapon ellenőrizhető, hogy 220 V vagy 240 V-os készülékkel rendelkezünk-e. A hálózati feszültségtől, annak ingadozásától, a DC motor áramfelvételétől függően előbb-utóbb jelentkezhetnek ezzel kapcsolatos problémák. A tápellátást biztosító feszültségstabilizátor nem tudja az esetenként 15–20 % hálózati feszültség hiányát semlegesíteni – főleg akkor, ha a nagyobb áramfelvétel miatt tovább csökken a szekunder feszültség és nő a puffer feszültség hullámossága.**

Panaszok jelentkezése esetén feltétlenül javasolt a gép 220 V-ra történő átalakítása.

#### TISZTÍTÁS

Néhány szó a karbantartásról. Az első jótanács, hogy ne éljünk vissza azzal, hogy készülékünk sokat kibír. Nyári melegben ne tegyük napsütötte ablakba, lehetőleg óvjuk a portól, ne ejtsük le – szóval vigyázzunk rá.

Az alapgépnél leirtak itt is érvényesek – feszültség alatti csatlakoztatás a soros vonal meghibásodásához vezet!

**A készülék a használat időtartamától és körülményeitől függően, de legalább félévenként rendszeres karbantartást igényel. A szervizek vállalkoznak erre a feladatra, de ha valamilyen oknál fogva nem akarjuk másra bízni, a karbantartást elvégezzük magunk is.**

A fedél eltávolítása után az elektronika nyomtatott áramköri lapját néhány csavar kicsavarása és a csatlakozók kihúzása után levesszük, így hozzáférhetővé tettük a mechanikát. Először tiszta és száraz ecsettel (esetleg porszívóval is) portalanítjuk. A fejet vezető két hengeres rudat (lásd a fotón az 1. sz. nyilat) egy-egy csepp műszerolajjal és egy papírsebkendővel vagy fültisztítóval fényesre tisztítjuk, miközben a fejet többször kézzel előre-hátra mozgatjuk.

A hengereken maradó olajfilm elegendő a kenésre is! Vigyázzunk, nehogy a kombinált fejre olaj kerüljön, tönkretelheti lemezeinket.

Nagyon kevés olajat tegyünk a rajzon jelölt, egymáshoz képest elmozduló felületek közé, és a felesleget itassuk fel.

– fémszalag feszítő-tárcsa csapágy (fotón a 2. sz. nyíl)

– fémszalag feszítő alsó megvezető felület (fotón a 3. sz. nyíl)

– fémszalag feszítő szerkezet és vezető rúd közé (fotón a 4. sz. nyíl)

– lemezforgató felső csapágy (fotón az 5. sz. nyíl)

**Szereljük vissza a panelt és gondosan dugjuk vissza a csatlakozókat a megfelelő helyre. Ha semmit nem tévesztettünk el, floppyknk újra működésre kész, és megbízhatóságával meghálálja gondoskodásunkat.**

**FLOPPY-  
DOKTOR**

**MŰSZI Személyi Számítógép Osztály**  
1118 Budapest, Sasadi köz 4.

Telefon: 668-823  
Telex: 22-7114



#### **HA MUNKÁJÁBAN GONDJA VAN...**

- személyi számítógéppel (COMMODORE 64 és 128) segítünk
- személyi számítógépesítéssel kapcsolatban szaktanácsot adunk

#### **HA GONDJA VAN, MERT COMMODORE-JA NINCS...**

- egy év garanciával szállítunk C-64 és C-128 számítógépet és ezek perifériáit. (A 128-as típust is állóeszköz-értékhatár alatt)

#### **HA GONDJA VAN, MERT COMMODORE-JA VAN...**

- megtanítjuk a kezelésére, programozására
- jótállási időn túl országos hálózattal átalánydíjas szervizt biztosítunk
- közel 100 alkalmazási és felhasználói programot szállítunk, adaptálunk.
- egyedi COMMODORE- programját elkészítjük
- áramkimaradás esetére szünetmentes áramforrást biztosítunk
- lépjen be COMMODORE klubunkba! A tagsági díj a belépéskor me...

**FORDULJON HOZZÁNK**  
**TOVÁBBI FELVILÁGOSÍTÁSRA**  
**ÁLLUNK ÉLŐRE**





# KEDVEZMÉNYEK

Tájékoztatjuk egyesületi tagjainkat, hogy a különböző cégek, vállalatok, kisvállalkozások által felajánlott kedvezmények, melyek az egyesület tagjait illetik, havonta változni fognak. Ez nem bosszantásból van, hanem azért, mert a cégek természetesen szeretnék felhívni a figyelmet bizonyos árucikkekre, szolgáltatásokra, így ezeket fogják havonta kiemelni. Így lapunk borító előtti utolsó oldala mindig az aktuális kedvezménylistát tartalmazza majd.

A Newline Számítástechnikai Vállalkozás 10 % kedvezményt ad az egyesület tagjainak:

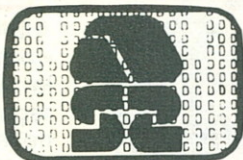
C 16 beépíthető 64 KByte memóriabővítő	150,- Ft
16-64-es átkapcsoló	490,- Ft
beépítés munkadíja	770,- Ft
ROMTURBO 16	3400,- Ft
együttes megrendelése esetén	3060,- Ft
árengedménnyel:	

**Jogosultak:** a Plusz- és a Szuperpáholy tagjai  
**Határidő:** 1986. szeptember 1-től november 30.  
**Igazolás:** ennek a tikketnek postai elküldésével  
**Cím:** Newline, 2220 Vecsés, Diófa u. 15.

**NEWLINE**  
 HARDWARE SOFTWARE

**10% 10% 10% 10% 10% 10%**

Az ÁPISZ-SZÁMALK számítástechnikai boltjaiban 10 %-os kedvezményt adnak mindenféle vásárlás után.



**Jogosultak:** valamennyi egyesületi tag

**Határidő:** szeptember 1-től 30-ig

**A kedvezményt nyújtó bolt:** Budapest, XI. Budafoki út 7.

**Igazolás:**

egyesületi igazolvánnyal

A kedvezmény

többször is

igénybe

vehető.

ne fenti javítják, hogy a le rásajtott al elfordulást len, hogy a tengely hengeren tengelyir

i. niu. ütköz. rögzítet.

ÁLTALÁBAN

A kiválóan műzetnek óriási hiány, tengely és a rásajto, néis mértékétől függőe. ni egymáshoz képest, motor már nem a gyárila helyre viszi a fejet. Mivel a pályák távolsága igen kicsi (0, könnyű elképzelni, hogy a kismér, elfordulás is zavarokat okoz a műkésben. Az elfordulás mértékétől függően bekövetkező hibák:

- Szekvenciális file-ok írásában, olvasásában bizonytalanságok lépnek fel.
- Jól beállított floppyval formatált lemezre a fej nem tud írni, (bizonytalanul olvassa)
- A floppyval régebben felvett lemezeket nem vagy nehezen olvassa.

**NOVOTRADE**

A Novotrade Rt. 2C áruházában 1986 szeptemberében a következő kedvezményeket adják:

CBM 610-es szoftverek:

7.15 Bérelszámolás 92.500 - Ft

7.10 Személyzeti nyilvántartás 20.000 - Ft

7.12 Vezinform 45.000 - Ft

7.14 Ranyi 30.000 - Ft

A fenti programokból

a szuperpáholy tagoknak 30 %

a pluszpáholy tagoknak 10 % árengedmény.

PC 20-as szoftver:

806 Bér- és munkaügyi rendszer 200.000 - Ft

Ebből a programból

a szuperpáholy tagoknak 20 %

a Pluszpáholy tagoknak 10 % árengedmény.

**Határidő:** 1986. szeptember 1-től szept. 30-ig.

**A kedvezményt nyújtó boltok:**

2C áruház, Budapest, XIII. Balzac u. 35.

2C pavilon a BNV területén

**Igazolás:** egyesületi igazolvánnyal

**A kedvezmény többször is igénybe vehető.**



**10%**

**A TECHNOCOMP Kiszövetkezet 10 %-os árengedményt ad C 16-os, C 64-es gépekre készült angol és német nyelvoktató programjaiból.**

**Részletes felvilágosítás és megrendelés:** 1476 Bp. Pf. 196. Kárpát u. 40.

**Jogosultak:** valamennyi egyesületi tag

**Határidő:** 1986. szept. 1-től nov. 30-ig.

**Igazolás:** egyesületi igazolvánnyal

**A kedvezmény többször is igénybe vehető.**

**10%**

A Novotrade-Fotoelektronik GT. az alábbi felsorolt szervezeiben mindenféle szervizszolgáltatás munkadíjából 10 % kedvezményt ad egyesületi tagjainknak.

**Jogosultak:** valamennyi egyesületi tag

**Határidő:** nincs

**A kedvezményt nyújtó szervizek:**

Budapest V., Magyar u. 12-14. Telefon: 173-551

Pécs, Kolozsvár u. 20. Telefon: (72) 11-812

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

Szeged, Székelysor 13.

Békéscsaba, Bartók B. u. 37.

Miskolc, Vologda u. 4. Telefon: (46) 17-011

**Igazolás:** a javítandó berendezés

leadásakor egyesületi igazolvánnyal

**A kedvezmény**

**többször is**

**igénybe vehető.**







**NYISSON BE  
HOZZÁNK!**



**NOVOTRADE** 2c

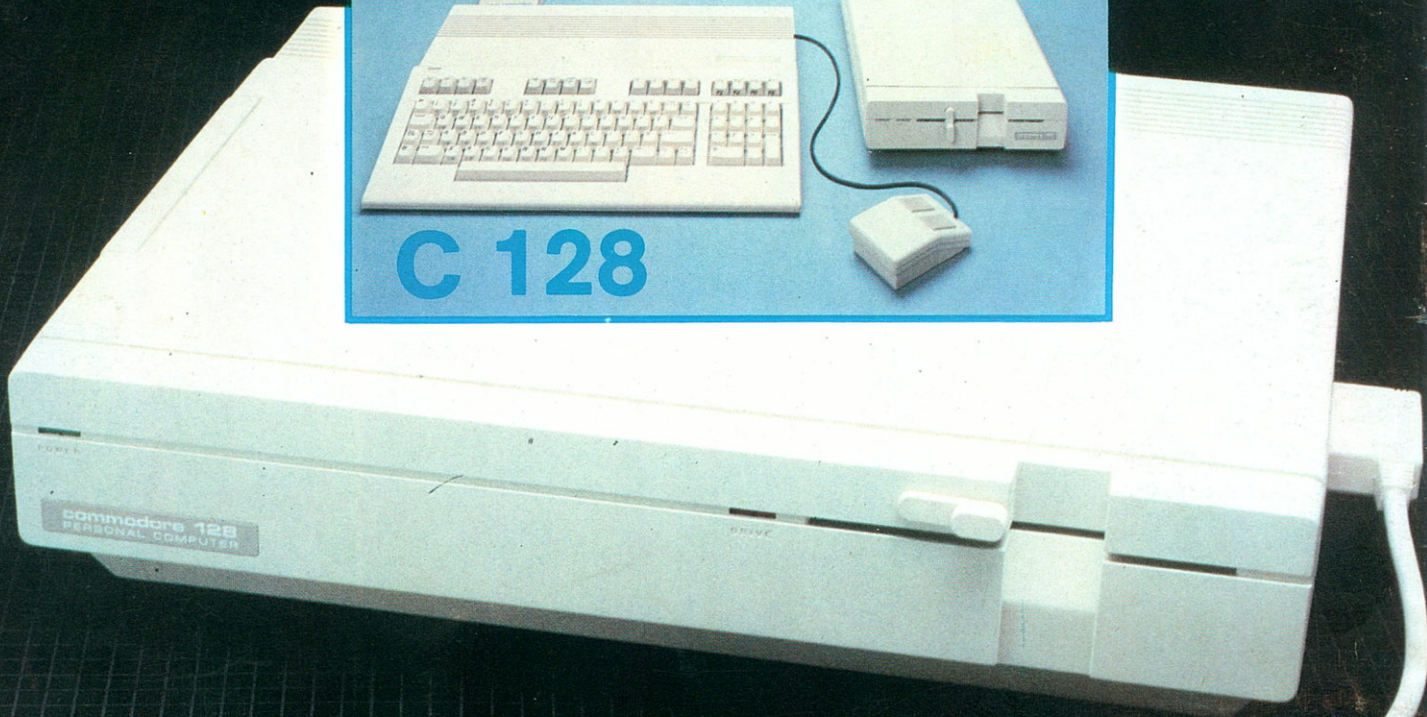
1136 Bp., Balzac u. 35. Tel. 402-954



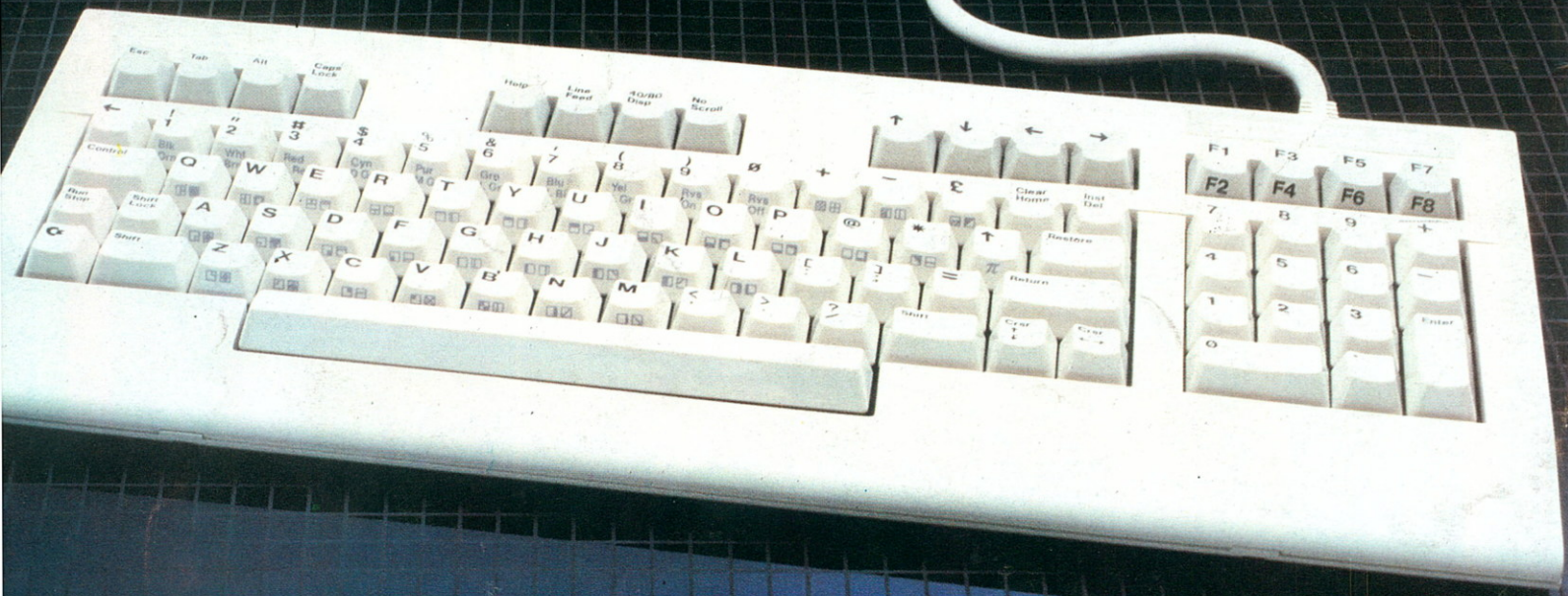




C 128



Commodore 128  
PERSONAL COMPUTER



COMMODORE 128  
PERSONAL COMPUTER