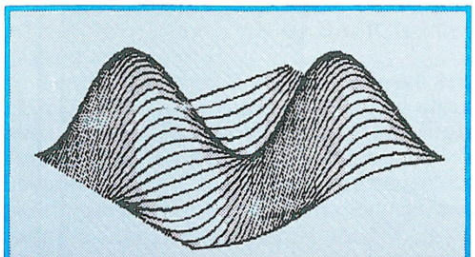
Le kellett tiltani a RUN/STOP billentyűt, mert ha olyankor állítjuk le a programot, amikor épp egy fájl nyitva van, akkor ez az adatok elvesztéséhez vezethet. A letiltás a POKE 788,52 utasítással érhető el, mely a C64 IRQ vonalának kezdőcímét EA31-ről EA34-re állítja. Ez kikerüli a RUN/STOP billentyű vizsgálatát. A STOP/RESTORE együttes leütése a bájt értékét az eredeti (49) értékre állítja vissza.

A program felépítése

2— 120	adatbeolvasás a szekvenciális állományokból
120— 400	egy városhoz havi hőmérsékletek + minimum, maximum és átlaghőmérséklet
400— 434	nyomtatási szubrutinok (gépi kódú hcopy meghívása)
6000— 6002	a gép alaphelyzetbe állítása (kilépés a programból)
6999— 8000	Magyarország-térkép a printer számára
8000— 9000	a program alaphelyzetbe állítása (visszatérés a kezdéshez)
9000— 9050	táblázatmagyarázat szubrutin
9050— 9100	minimum-, maximumszámítás
9100— 9200	1. üzemmódváltás
9200— 9300	egy konkrét időpontban mért adatok kiírása (a 20 város mellé az adatok)
9300— 9450	térképes üzemmód
9450— 9600	2. üzemmódváltás
9600— 9630	a húsz város átlaga egyszerre
9630— 9700	egy város átlaga
9700— 9900	oszlopdiagram üzemmód
9900—10000	lemezre írás üzemmód
50000—50012	diszkhiba esetén meghívott szubrutin

A program még két gépi kódú szubrutint is tartalmaz C000 memóriától kezdve. Az egyik egy hcpoy (képernyőtartalom kinyomtatása nyomtatóra), a másik egy PRINT AT utasítást helyettesítő rutin. A két program együtt sem több, mint 1 blokk a lemezen. A teljes program terjedelme meghaladja a 10 kbájt nagyságot, ezért listáját nem közöljük. Érdeklődők a HCC Commodore Szekciónál saját lemezükre átmásolhatják.

KOCSIS TAMÁS



Tass Zoltán grafikái (lásd még a 22., 35., 43. és 46. oldalon is)

FÓRUM

Micsoda kiszolgálás! 2.

Élvezettel olvastam a Mikroszámítógép Magazin 1985. évi 6. számában e címen megjelent derűs kis cikküket, amelyből megtudhattuk, hogy milyen csodás körülmények között vásárolt egy vékony pénzű magyar turista München egyik áruházában kisfia számára egy olcsó, sőt leértékelt számítógépet. Ha írásom az említett cikk karikatúrájaként sikeredne, arról nem én tehetek, hanem a tények, amelyekhez igyekszem szigorúan ragaszkodni.

1985. április. Örömmel értesültem az egyik volt diákomtól, hogy a Videoton elkészült olcsó személyi számítógépével, és TV-Computer néven hamarosan forgalomba is hozza.

1985. május. A BNV-n, a Videoton standjánál már sok számítógép-rajongó gyönyörködhetett a TV-Computer néhány példányában. A Mikroszámítógép Magazin, a Számítástechnika, sőt még a napilapok is írtak róla vásári beszámolóikban.

1985. június. Lelkes diákjaimmal úgy búcsúztunk a számítástechnikai szempontból is sikeres tanévtől, hogy az új tanév kezdetére már bizonyára ezzel az új, sokat ígérő géppel is gyarapodni fog iskolánk gépállománya.

1985. szeptember. Diákjaim megjöttek, de a gép sajnos nem. Helyette hozta az egyik diákom a géphez ígért könyvsorozat első két könyvét: a TV-Computer kezelési útmutatóját és BASIC programozási segédletét. A két csinos kiállítású könyv még jobban felcsigázta érdeklődésünket. Tartalmuk azt mutatta, hogy a gép sokat tud és sok mindenre használható. Örültünk, hogy a gépnek már megjelenésekor lesz használható irodalma. Most már egyre türelmetlenebbül vártunk.

1985. november. A SW '85 szoftvervásáron az egyik diákom dolgozhatott is a gépen. Nagyon tetszett neki. Reméltük, hogy karácsonyra mindannyian örülhetünk a gépnek. Alátámasztották ezt a reményt híreink is. A Mikroszámítógép Magazin 1985. évi 4. számában Videoton-újdonságok címen megjelent a TV-Computer fényképe és a hír, hogy a gyár az év második felében 2000 darabot szeretne belőle piacra hozni. A Számítástechnika novemberi számában pedig egy rövid hírben azt olvashattuk, hogy a TV-Computerek is Tabon készülnek. Tanítványaimnak kiadtam, hogy aki a karácsonyi vásáron lát valahol ilyen gépet, azonnal jelentse, hogy megvehessük.

1985. december. Az Ötletben teljes oldalt kitöltő hirdetés, hogy a Videoton forgalomba hozta a TV-Computert. Alatta telefonszámok, ahelyett, hogy a vásárlási helyek és a gép ára lennének felsorolva. (Nálunk még nincs gyakorlatban a telefon útján való rendelés!)

1986. január. Elmúlt a karácsony, de gépünk nincs. No de sebaj! Az Ötlet január 2-i számában ismét egy egész oldalas hirdetés — telefonszámokkal.

1986. január 5. A tv vasárnapi szép színes reklámjaiban kétszer is felvillant a TV-Computer hirdetése.

1986. január 8. Mivel diákjaim még mindig nem hoztak hírt arról, hogy valamelyik kirakatban látták volna a TV-Computert, a telefonra fanyalodtam. Felhívtam a hirdetésekben megadott telefonszámot. A vevőszolgálatnál barátságos férfihang fogadta kérdéseimet. Néhány dicsérő szót is mondott a gépről, majd kapcsolta az illetékes csoportot. Itt megkaptam a kért információkat: az egyik művelődési házban van ilyen gép, ott megnézhetem, és az egyik nagyáruházban érdeklődhetem a vétellel kapcsolatban.

Másnap kora délelőtt elmentem az áruházba. Kicsit reménykedtem, hogy ebben az előkelő, háromszintes épületben valami olyat látok, amit az előző cikk írója abban a müncheni külvárosi diszkontáruházban.

A karácsony utáni ismert kép fogadott. Alig lézengett néhány érdeklődő, de a polcok bőséget mutattak. Örültem, hogy kevés a vásárló, mert így figyelmesebb kiszolgálásra számíthattam. Végignéztem a műszaki osztályt, de a TV-Computerre vonatkozó, általam elképzelt nagy reklámokból semmit sem láttam, még magát a gépet sem. Az egyik polcon felfedeztem a Primót: egyedül szomorkodott, tévé és rivális nélkül. Ez felbátorított, és az arra sétáló eladótól érdeklődtem a Videoton Computer után. Kijelentette, hogy ilyenről még nem is hallott, de a biztonság kedvéért megkérdezte a kolléganőjét, tud-e róla. A hölgy végignézett rajtam, és azt mondta, hogy ha itt van nálam a számítógép ára, 20 ezer forint, akkor kihozza és megmutatja. Meglepetésemben alig tudtam megszólalni, de látszhatott az arcomon, hogy valóban nincs a zsebemben ennyi pénz, mert a hölgy diadalmasan elvonult a csatatérről. Így nemcsak bemutatásból, de még megmutatásból sem lett semmi.

Szégyenkezve kullogtam ki az áruházból. Restelltem, hogy nyugdíjas létemre nem 20-30 ezressel a zsebemben indulok nyomozásra egy olyan számítógép után, amelynek csak a reklámozására már legalább 100 ezer forintot költöttek.

Optimista vagyok, és remélem, hogy mire írásom megjelenik, minden áruházban és Keravillban lehet két-három fajta hazai gyártmányú számítógépet kapni. Ha így lesz, akkor szívesen elfelejtem mostani bosszankodásomat. De a propaganda, a vevőszolgálat és a kereskedelem tehetne valamit azért, hogy a vásárlás a vevő számára ne nyomozás, boszszúság és megaláztatás, hanem öröm legyen!

KOVÁCS MIHÁLY

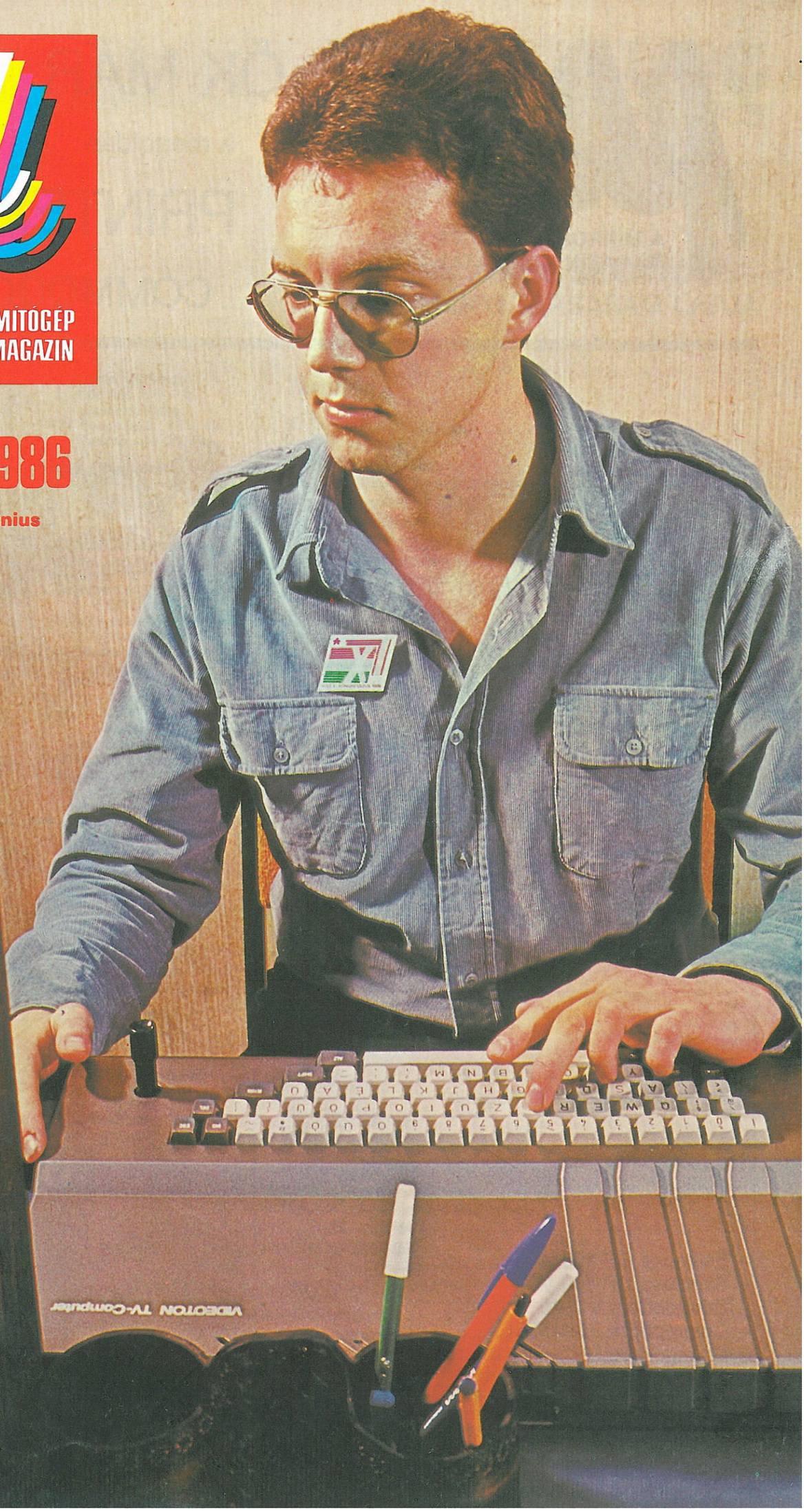
MIKROSZÁMÍTÓGÉP MAGAZIN

1986

június

Ára: 30 Ft

A NEUMANN JÁNOS SZÁMÍTÓGÉP TUDOMÁNYI TÁRSASÁG LAPJA



A VIDEOTON TV-COMPUTER

Az iskolaszámítógép-pályázaton a TVC (a gyártó által használt rövidítés) az általános iskolai kategóriában megosztva első díjat nyert, a középiskolai kategóriában második lett. A két kategóriában indított változat egyébként csak a RAM méretében különbözik.

A gyártó elsősorban tömeges felhasználásra: oktatásra, játékra, egyéb házi használatra készítette gépét, de a bővítés lehetőségeit kihasználva, tervezi egy kisebb, professzionális feladatok elvégzésére is alkalmas rendszer kiépítését.

A készülék felépítését vizsgálva szokatlannak tűnik a jobb oldalon látható botkormány. A hazai gyártmányoknál ugyancsak szokatlan, több buszcsatlakozóval ellátott kivitel és a botkormány miatt a gép az általában szokásos háziszámítógép-méretnél nagyobb. Az 1. és 2. képen bemutatott nézetek alapján ránézéssel is megállapítható, hogy a gép viszonylag sok periféria csatlakozását teszi lehetővé.

A billentyűzet

A billentyűzet 66 nyomógombot tartalmaz. *Ez az első, hazai tömeggyártásba került mikroszámítógép, amelyen az összes magyar ékezetes karakter fellelhető.* A billentyűzet részét képezi a fénypont nyolc irányú mozgatására szolgáló, de játékoknál is használható botkormány.

A billentyű lenyomásakor a billentyűzet alatt elhelyezett nyomtatott áramköri kártya aranyozott, fésűs érintkezőit vezetőgumi zárja. Így képezik azokat a jeleket, amelyeket a számítógép a billentyű azonosítására használ. A gyártó tájékoztatása szerint a billentyűk élettartama átlagos körülmények között több mint 5 millió leütés. Ellentétben az eddig itthon gyártott billentyűkkel, a feliratok kettős fröccsöntéssel készültek, így nem vesztik el tartós használat után sem olvashatóságukat.

A beépített botkormány a billentyűzethez hasonlóan membránkapcsolós, várható élettartama nem túl nagy. A bot letörésének megakadályozására alkalmazott felütköztetéses megoldás az üzletekben gyakran kapható legolcsóbb botkormányokénál várhatóan nagyobb élettartamot biztosít.

A billentyűzeten található vezérlőgombok közül a SHIFT funkciója közismert, a LOCK az írógépek egy részénél használt UPPER CASE rögzítő, a CAPS pedig az előző hatásához hasonló, de csak a kisbetűk váltódnak át. Teljesen újszerű az ALT gomb hatása: lehetővé teszi a karakterek egy részének átírását és további új karakterek készítését. 32 karakter írható át, és további 64 definiálható. Az átírás is és az új karakterek írása is oly módon történik, hogy az erre szolgáló utasítás segítségével közöljük a számítógéppel, hogy milyen kódszámú karaktert adunk meg, és e karakter tíz sorának milyen értékei lesznek. A karakterek mindegyike egy nyolcszor tízes pontmátrixban helyezkedik el, mely mátrix egyes pontjainak ki-, illetve bekapcsolt voltát kell meghatároznunk. Az általunk „elkészített" karakterek ezután a megadott kódszámra történő hivatkozással vagy az ALT gomb segítségével írathatók ki a képernyőre, sztring formában is.

A billentyűzet nyomógombjainak lenyomott helyzetét egy 8 × 10-es mátrix 8 × 8-as része érzékeli, a megmaradó 2 × 8-as mezőt a botkormány használja.

A csatlakozók

A használt csatlakozók aranyozottak, jó minőségűek. Az *1. képen* látható egy párhuzamos nyomtatóillesztő csatlakozója, a tápegység csatlakozója, két magnetofoncsatlakozó és egy összetett videojel-csatlakozó, az ún. RGB-kimenet (ez három főszín külön-külön történő kivitelére szolgál), valamint a hagyományos tv-csatlakozó. A *2. kép* két, általánosan használt külső botkormány csatlakoztatására szolgáló csatlakozót és az ún. programmodul-csatlakozót mutatja. Ezek közül különösen előnyösnek tartom az RGB-, a két független magnetofoncsatlakozó és a nyomtatóillesztő meglétét, és igen célszerűnek az 1. képen látható négy további buszcsatlakozót, melyek használatával különböző bővítőkártyák csatlakoztathatók további egységek kiépítése nélkül.

A csatlakozók minőségére vonatkozó általános megállapítás mellett meg kell jegyezni, hogy a tv-csatlakozó mechanikai felépítése (rögzítése) nem tűnik időt álló megoldásnak.

A készülék külsején helyezkedik el még egy, a színek ki-be kapcsolására szolgáló nyomógomb, mely megkönnyíti a fekete-fehér tévék használatát. A számítógép alsó felületén, rugalmas lemezként, a doboz anyagából van kialakítva a RESET gomb. Ennek elhelyezését és kialakítását nem tartom szerencsésnek; csak bizonyos gyakorlat után található meg könnyen és kezelhető biztonságosan. Az a megoldás, hogy „rövid idő"-n belül ismételten megnyomva, a memóriát is törli, veszélyesnek látszik, mert előfordulhat nem szándékos törlés is (a „rövid idő" a gyártó tájékoztatása szerint 0,5-0,6 másodperc). Programmegszakításra a CTRL + ESC gombokat használva, ez teljesen kizárható.

A színestévé-használatot bizonyos mértékben korlátozza, hogy csak a PAL-rendszerű, illetve a kétnormás tévék adnak színes képet. A legolcsóbb és igen elterjedt szovjet televíziók nem alkalmasak a gép színes képének előállítására, mivel csak SECAM-rendszerűek.

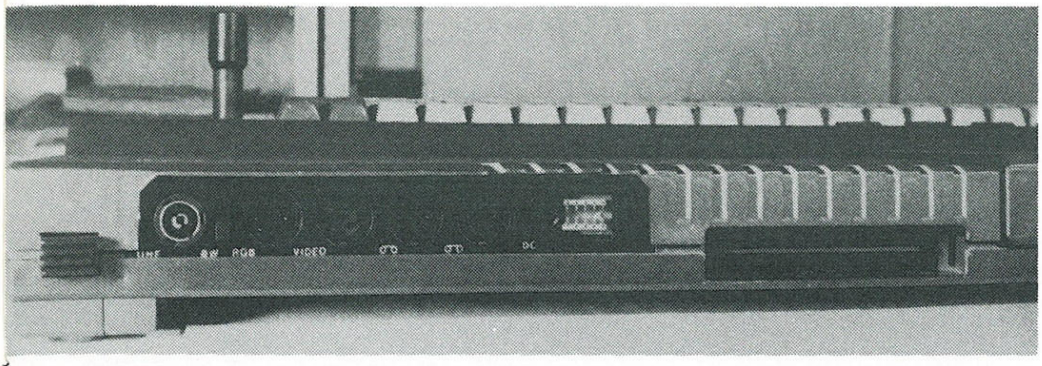
A lábazat rugós kialakítása nem tűnik szerencsés és tartós megoldásnak. A doboz relatív nagy mérete miatt lehetségesnek tartom, hogy az időjárás változásai miatt bekövetkező összehúzódások, tágulások olyan vetemedéseket okozhatnak, amelyek következtében a doboz megsérülhet, és kártyaközi zárlat léphet fel.

A belső felépítés

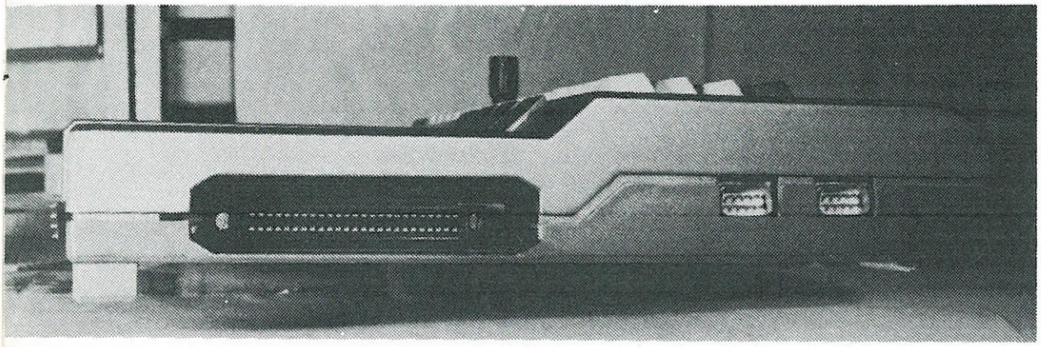
A számítógép belsejében egy nagy-, egy közép- és egy kisméretű kártya található. A középméretű (a 3. képen nem látható) a billentyűzet kiszolgálását végzi. Ez egy csatlakozóval kapcsolódik a nagy kártyához, és mechanikailag függetlenített attól. A kisméretű kártya a PAL kódolási funkciókat valósítja meg. Az RGB-kimeneteket használva erre a kártyára nincs szükség. Felépítésében egy szempontból lényegesen különbözik a hasonló célra kidolgozott megoldásoktól: az UHF-modulátor nem a világszerte szokásos Astec gyártmány, hanem egy speciális IC-t tartalmazó megoldás. A színsegédvivő olyan zavarásai ellen, amik például a C64 betűszéleinek „elszíneződését" okozzák, olyan megoldást dolgoztak ki, amely — saját tapasztalataim szerint — eredményes volt. A már említett 8 × 10-es méretű betűmátrix jobb, mint a Commodore és Sinclair gépek 8 × 8-as mátrixa. A nagyobb méret teszi lehetővé a magyar karakterkészletnek a helyesíráshoz közelebb álló megvalósítását. Ez azt jelenti, hogy a a teljes magyar betűkészlet megvan, de a betűk mérete, az ékezetek elhelyezése, a betűk alakja egyedi. A beállító képernyőkijelzési módok közül a 16 karakteres nem elégséges, a 64 karakteres csak közelről olvasható, és néhány betű, például a hosszú ű és hosszú ő összetéveszthető.

E megjegyzéstől eltekintve, a kép feketefehérben olvasható mind Junoszty 402D, mind Elektronika C—401m típusú televízókon is, nemcsak Videoton gyártmányú tévéken. Az ékezetekkel kapcsolatban eltérés a magyar helyesírástól, hogy az Ú kivételével az összes nagybetűn pont van (Á, É, Ó, Ű esetében is!). Amennyiben Y, Ú, illetve X, Ü vagy Y, U kerül egymás alá, úgy összetéveszthetők. Ugyanez érvényes az O, Ö, Ő és G betűkre is. A betűk itt is összenyomottak. A képernyőn a szöveges kijelzésen kívül háromféle grafikus kijelzés is megvalósítható. Ezek a címezhető képpontok és a használható színek számában különböznek.

Az órajeleket egy 25 MHz-es kvarcoszcillátor állítja elő. Ennek jeléből felezéssel kapják a képelőállításhoz szükséges alapórajeleket, melyekből további felezéssel lehet megkapni a különböző felbontású alapórajeleket. A megjelenítőegység a kép frissítéséhez a memóriából 640 ns-os egységenként olvas ki egy-egy bájtot. Az informáci-



1. kép



2. kép

ók képernyőre való kiviteléhez szükséges adatok a 16 színű üzemmódban másképp tárolódnak, mint a két másikban. Az elsőnél egy képponthoz négy bit tartozik, melyből az első három a három alapszínre vonatkozik, az utolsó a színintenzitást adja meg (teljes vagy fél); a két másik üzemmódban az egy, illetve két bit egy ún. palettaregiszterben történő tároláshoz szükséges sorszámot jelez. A kép keretszíne külön kerül tárolásra. Az említett négy bitből adódó három színjel kerül ki az RGB-kivezetésre. A PAL videojelhez színsegédvivő tartozik, melynek jelét fekete-fehér tévé használatakor kapcsolóval ki lehet iktatni, a kapcsoló mindenkori helyzetét a program érzékeli. A színsegédvivő a sorfrekvenciához szinkronizált.

A mikroprocesszor Z80A típusú, 3,125 MHz frekvencián járatott. Mivel a memória lehetséges maximális mérete meghaladja a mikroprocesszor által címezhető területet, ezért lapozásos technikát használnak. A felhasználói RAM 32-64 kbájt; a *3. képen* látható, 4116 típusú integrált áramkörökkel valósították meg. A képmemória 16 kbájt. A mikroprocesszor ehhez a memóriaterülethez közvetlenül is hozzáférhet, így azt módosítani is képes. A rendszer ROM 20 kbájt, és a BASIC-interpretert, valamint az azt kiszolgáló operációs rendszert tartalmazza. Ezt maximálisan 24 kbájtig lehet kiterjeszteni. Programmodulok („dobozolt EPROM") számára maximálisan 16 kbájt áll rendelkezésre (ilyenek például a játékprogramok, az UPM — CP/M 2.2 — operációs rendszer). A perifériákhoz tartozó memóriák a külső kártyákon helyezhetők el, kártyánként maximálisan 8 kbájt nagyságban. A BASIC használata esetén a ténylegesen felhasználható memóriaterület 25—41 kbájt. A négy lapra osztott memória-címterület első lapján a rendszer ROM, a felhasználói RAM eleje, a programmodul ROM helyezkedik el. A második és a harmadik lapon a felhasználói RAM további részei vannak, ez utóbbin a képernyő RAM is. Az utolsó lapon a maradék található.

Az összes órajelet egy alapfrekvenciából állítják elő, osztólánc segítségével. Az adatbusz pufferelt, a busz többi része nem. Az összes kártyát lakkozás védi a por és légszennyezés ellen.

A hangkeltő áramkör egy 12 bites programozható számlánc, mely a mikroprocesszor órajelét osztja le. A számláncot 16-os leosztás követi. A számlánc kimenete négybites amplitúdószabályozó áramkörön keresztül kerül a kimeneti csatlakozóra, illetve az UHF-modulátor bemenetére. Az amplitúdószabályozó bemenetére folyamatos „1"-es szint is adható, ekkor D/A konverterként működik.

A kazettás egység jelét billenőkör állítja elő. A jel frekvenciája adja meg információtartalmát. A kazettán a jel formátuma a következő: egy 4 s hosszúságú, 470 μs periódusidejű, négyszög alakú szinkronizáló jelsorozat, melyet egy 740 μs periódusidejű négyszögimpulzus zár. Az átviendő adatok 256 bájtonként 2 bájtos CRC-vel (Cyclic Redundancy Check) ellenőrzöttek. A „Ø" 555 μs periódusidejű négyszögimpulzus, az „1" pedig 395 μs. A szinkronsorozat célja átlagsebesség-meghatározás, ehhez állítja be a számítógép a frekvenciát. A két magnetofoncsatlakoztatási hely egymástól függetlenül, programból kiválaszthatóan működtethető. Bekapcsoláskor vagy RESET után a tápcsatlakozóhoz közelebb levőt címezi a BASIC interpreter. A másik csatlakozó START/STOP áramköre POKE utasítással működtethető.

Három magnetofonnal vizsgáltam, hogy a gyártó által a géphez adott DEMO-kazettáról hogyan olvas, illetve ír. A gyártótól kapott BRG MK—29 típusúval mindkettő hibátlan volt, egy Panasonic (japán) R 5090 típusúval 6 olvasásból 4 sikerült (az írás kábelhiba miatt nem volt vizsgálható), egy Centrokord (NSZK) 250 típusúval teljesen eredménytelen volt. Nem maximális hangerőnél semmit sem olvasott be, maximálisnál a fejlécet általában igen, de néhány blokk beolvasása után az ellenőrzőszám hibája miatt leállt.

Bővítőkártyák

A gép bővítésére négy beépített kártyacsatlakozó szolgál. Ezekre vezették ki a buszt; ezek a pozíciók BASIC utasítással érhetők el. Az egyes csatlakozók egy kétbites számmal közvetlenül megcímezhetők, vagy a BASIC-ből is elérhetők. Eddig RAM-bővítés, lemezegység-csatoló, soros illesztő, általános célú párhuzamos illesztő készült el.

Rendszerprogramok

A BASIC-interpretert támogató operációs rendszer számos hozzáférhető szubrutint tartalmaz. Ezek egy 8 bites funkciókód használatával elérhetők. A funkciókódból az alsó hét bit kerül kihasználásra, melyek közül a felső három a szubrutin osztályát határozza meg (például a 0 a video, az 1 a billentyűzet stb.), az alsó négy pedig a konkrét funkciót adja meg (például a 0 a megszakítást, az 1 a karakteres ki-be menetet).

BASIC

A BASIC számos tulajdonságában eltér a világszerte legelterjedtebb és itthon is gyakran használt MICROSOFT típustól. Bizonyos tulajdonságai hasonlítanak a Spectrum típushoz. Néhány előnyös tulajdonsága:

— a LIST parancsot tartománysorozat is követheti,

— a DELETE parancs létezése, melynél tartománysorozat is használható,

— a RESTORE sorszám létezése,

— a 250 karakter hosszúságú utasítássor és a 254 karakter hosszúságú karakterlánchossz,

— a sokféle paraméterrel (szín-, vonaltípus-, vonalkeresztezés-, időállandó-, holtidő-beállító, karakterdefiniáló) ellátható SET létezése,

— a PLOT utasításnál több vonalhúzás is összefogható (kár, hogy nem használhatók relatív elmozdulások, koordinátatengely és mellékirány felé tett elmozdulásokra egykarakteres alutasítások, elforgatás, nagyítás),

— 14-féle vonalhúzás állítható be, és a vonalkeresztezések eredménye logikai függvénykapcsolattal meghatározható,

— a botkormányok az INKEY$ utasítással olvashatók be.

Felsorolok néhány hátrányt is:

— a hangelőállításnál a szünet hossza nem szabályozható, a zenei skálahangok helyett frekvenciákat kell megadni,

— a kompatibilitás hiánya más BASIC-interpreterekkel.

Az alábbiak előnyös vagy hátrányos volta vitatható:

— a kiíratásnál nem törli a sort, amelybe ír, de törli az utána következő karaktert,

— ha a TAB értéke kisebb, mint az előző PRINT által kiadott karaktersorozat vége, akkor ebbe a sorozatba beleírja a következőt,

— a NEW és LOAD hatására a LOMEM, amellyel a gépi kódú programnak helyet biztosítottunk, alapértékére áll vissza, de a VLOMEM tartalmának átírásával ennek hátránya megszüntethető. NEW és LOAD esetén a LOMEM a VLOMEM értékét veszi fel,

— a RESET hatására a négyszínű üzemmódba ugrik, de CTRL+ESC-nél nem!

— az INPUT PROMPT forma,

— az INPUT-nál az, hogy hibás adatformátum „Ø", illetve „ " beadását jelenti,

— az egymásba ágyazott IF... THEN hatástalanítja (az első kivételével) az ELSE-t,

— a FOR... NEXT ciklusból a kiugrás megengedett,

— az ASC-függvény nevének ORD-ra változtatása.

A dokumentáció

A géphez két könyvet kap a vásárló: a Kezelési útmutatót és a BASIC programozási segédletet. Ezek a felhasználó részére igen sok, jól használható adatot tartalmaznak, az eddig itthon megszokottaknál lényegesen jobbak. A Kezelési útmutató általában jól követhető, érthető, alapos. Megemlítek azonban néhány olyan hibát, amely a hibajegyzékben nem szerepel és zavaró.

A 15. oldalon leírtak ellenére a SHIFT gomb lenyomása nem váltja vissza a fénypontot alapmódba, ezt csak a LOCK gomb lenyomása eredményezi.

A 22. oldal első példájában a LOG nem „10-es alapú logaritmus"-t jelent, hanem — ahogy ezt a példa eredménye is bizonyítja — ún. természetes alapú logaritmust.

Az Ismerkedés a BASIC-kel című fejezet valódi kezdőknek nehéz, a példák egy része egy átlag kezdő számára nem érthető. A már említett hibás példa is ilyen. Igen előnyösnek tartom a csatlakozókiosztás ismertetését (kár, hogy jelmagyarázat, terhelési adatok nélkül), az egyéb műszaki adatok közlését, a billentyűzetábrák és a karakterkód-táblázat megadását.

Megtévesztő lehet viszont a BASIC eltéréseit tartalmazó fejezet. Ezt olvasva ugyanis az olvasó azt hiszi, hogy itt megtalálja az összes eltérést, azonban nem szerepelnek olyan eltérések, mint például a perifériára írásnál a PRINT n: formátumban használatos „:", az INPUT PROMPT forma létezésének említése és ebben a „:" használata, az INPUT-nál az, hogy hibás adatformátum „Ø", illetve "" értéket okoz, az egymásba ágyazott IF... THEN hatástalanítja (az első kivételével) az ELSE-t, a FOR... NEXT ciklusból engedélyezett a kiugrás, nincs említve az ON utasításban

MIKROPROCESSZOR ÉS MEMÓRIA

Típus:	Z80A
Órafrekvencia:	3,125 MHz
Rendszer ROM (OS+BASIC):	20 kbájt (max: 24 kbájt)
RAM memória:	332 kbájt (64 kbájtig bővíthető)
Video RAM:	16 kbájt
I/O ROM+RAM:	max. 8 kbájt/kártya (max. 4 kártya)
Programmodul:	max. 16 kbájt ROM

KÉPERNYŐSZERVEZÉS

240 × 128 pont, 16 szín (24 × 16 karakter)
240 × 256 pont, 4 szín (24 × 32 karakter)
240 × 512 pont, 2 szín (24 × 64 karakter)
A két- és négyszínű üzemmódban az aktuális színek tizenhatos színkészletből választhatók. Külön állítható a képkeret színe (tizenhat szín).

KARAKTERKÉSZLET

10 × 8-as teljes pontmátrix
96 definiált, nem módosítható karakter
32 definiált, újradefiniálható karakter
64 nem definiált, a felhasználó által definiálható karakter

HANG

Frekvencia (Hz): 195312,5 (4096-P), ahol P=0...4094
Amplitudó: tizenhat fokozatban programozható

BILLENTYŰZET

57 alfanumerikus nyomógomb
kilenc szerkesztő-, funkció- és kódmódosító nyomógomb
beépített kis botkormány (négy irány)

KI- ÉS BEMENETEK

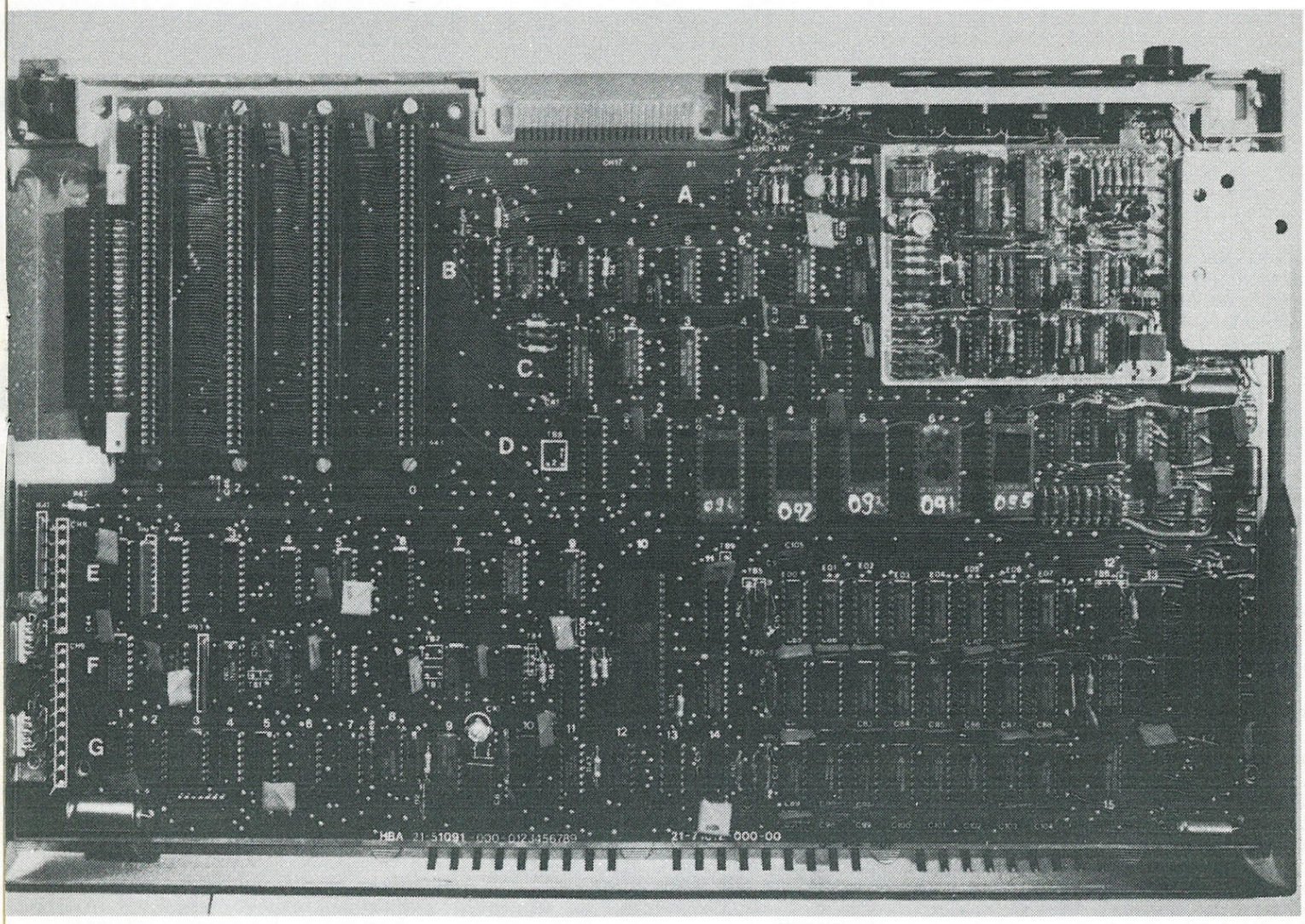
Antennacsatlakozó
Képvivő: UHF 25...40 csatorna között
Hangvivő: Képvivő+6,5 MHz, FM
Szín: PAL-rendszerű, a színsegédvivő kikapcsolható

RGB-kimenet
A G, R, B jelek szintje 75 ohmmal lezárva:
— a teljes fényű színeknél 1 V
— a sötét színeknél 0,6 V
— fekete szint: 0 V
Polaritásuk pozitív
Szinkronjel: 1,4 Vpp 75 ohmmal lezárva, polaritása negatív

Összetett videojel-kimenet
Videojel: — amplitudó: 1 Vpp 75 ohmmal lezárva
— polaritás: pozitív (szinkron negatív)
— színkülönbségi jel: PAL-rendszerű, kikapcsolható
— a nem leválasztott kimeneten a fekete szint 75 ohmos lezárás esetén kb. +2 V
Hang: — a maximális kimenőszint 1,1 Vpp

Magnócsatlakozók
Kimenet: 200 mVpp
Bemenet: 0,8...5 Vpp
Motorvezérlő jel: földelt emitteres tranzisztor kollektora van kivezetve,
a tranzisztor határadatai:
Ucmax: 30 V
Icmax: 300 mA

1. táblázat. A Videoton Elektronikai Vállalat TV-Computerének műszaki adatai



3. kép

az ELSE léte, az ASC-függvény nevének ORD-ra változtatása.

A BASIC programozási segédlet kezdőknek érthetetlen, mert számos fogalom ismeretét feltételezi. Más BASIC-változatokat ismerők számára azonban jól használható. A gyakorlottabb programozók számára előnyös a BASIC-változók és munkaterületek egy részének, az adatábrázolás módjának bemutatása. A gépi kódú programozást elősegíti a Z80 utasításainak ismertetése.

A BASIC egyik jellemző adata a futtatási sebesség. A közismert KILOBAUD-teszt eredményét a *2. táblázat* tartalmazza, a korábban már más helyen közölt néhány gépre vonatkozó adatokkal együtt.

GÉPTÍPUS	PROGRAMSORSZÁM 1	2	3	4	5	6	7	8
C64[1]	1,6	9,7	18,3	20,3	21,8	31,5	49,5	115,9
Spectrum[1]	4,8	8,7	21,1	20,4	24,0	55,3	80,7	253,0
Atari 400/800[1]	2,3	7,4	19,9	23,2	26,8	40,7	61,5	431,0
Aircomp 16 (gyors üzem)[1]	0,7	4,0	8,0	8,0	9,5	15,0	23,0	30,0
Primo[2]	1,4	9,2	17,6	20	20	32	51	84
Dragon (gyors üzem)[2]	0,7	5,1	9,9	10,7	11,7	17,9	25,0	64,0
TVC	1,9	9,0	21	28	30	44	58	250

[1] ÖTLET-BITLET 1984. január
[2] Mikromagazin 1984. 6. szám

2. táblázat. A KILOBAUD-teszt eredménye

Összefoglalás

A gép az itthon használt más gépekkel összehasonlítva egyes tulajdonságaiban, melyeket korábban ismertettem, lényeges előnyöket mutat, és különösebb hátrányát nem találtam. Meg kell azonban jegyeznem, hogy tartós üzemű vizsgálat végzésére még nem volt lehetőségem. A dokumentációt az itthon megszokottaknál lényegesen jobbnak találtam. A gyártók eddig már jó néhány programot kifejlesztettek, például játékprogramokat, műszaki, irodai alkalmazásokra szánt programokat. A hardverbővítésben megvalósított lemezegység-kezelőn keresztül a CP/M 2.2 egy változata (UPM) biztosítja a kompatibilitást a korábban erre a rendszerre kifejlesztett programokkal. A gyártó más gépekre kifejlesztett táblázatkezelőjét, szövegszerkesztőjét, formanyomtatvány-kezelőjét, formátumgenerátorát, adatrögzítés-vezérlőjét, levélnyilvántartóját, FORTH rendszerét kívánja ehhez a rendszerhez átdolgozni. A rendszer további perifériákkal, például a már gyártott nyomtatóillesztővel és soros illesztővel is ellátható. Úgy tűnik, hogy a gyártó teljes rendszert igyekszik kiépíteni.

PROGRAMOK

ÖT GÉP VIZSGÁLATA

Egy programmal

1985/6. számunkban cikket írtam „Melyiket vegyem?" címmel, melyben a legalacsonyabb árkategóriába tartozó 10 gépet összehasonlító táblázatokat közöltem. Ezekhez kiegészítést és egyúttal két változathoz programot ad ez a cikk.

Különböző gépek — igazában azok fordítóprogramjai, de mert ezek gyakran azok tartozékai, tehát mindkettő — összehasonlítására, adott programok futási sebességeit közli a szakirodalom. (Mi is közöltünk ilyet lapunk 1984/6. számában, a Primo-ismertetőben.) E módszer hátránya, hogy a legtöbb gépen futtatható programokat igényel. Ilyenek készítésénél viszont a gépek előnyös, egyedi tulajdonságait nem vehetik figyelembe.

Egy másik lehetőség az összehasonításra, ha nem a program, hanem a feladat adott. Így, ha a feladatok a leggyakrabban előforduló felhasználásokat felölelik, egy olyan összehasonlítást kapunk, ami reálisabb képet adhatna. A feltételes mód oka az, hogy itt az is feltétel, hogy a gépek tulajdonságait teljesen kihasználó programokat kellene összehasonlítani. Ehhez viszont „mindentudó" programozók kellenének. Ilyenek természetesen nincsenek.

Egyes nagy példányszámú nyugati lapok a legelterjedtebb géptípusok lehetőségeit viszonylag jól ismerő állandó szakírógárdával igyekeznek hasonló színvonalú programokat készíteni ugyanarra a feladatra. Ezek támpontot jelentenek a gépkiválasztásnál.

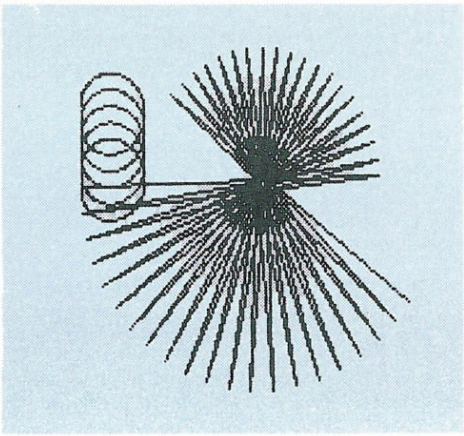
A COMPUTE! c. szaklap 1983. októberi számában közölt egy olyan programot, ami cikkem tíz gépe közül ötre, valamint két másikra (Apple II és VIC-20) ad változatot. A közölt változatok a két Atari, a C64 és a két Dragon gépen futtathatók. (A TRS 80 Color és a Dragon gépek BASIC-szinten azonosak.)

Maga a program a Spirograph nevű eszköz rajzolásmódját általánosítja. Egy olyan vonalhálózatot állít elő, amely két sugár mozgatásával jön létre. Az egyik sugár — a beadott értéktől függően — egyenletesen körbe fordul vagy egy egyenes mentén mozog; a másik sugár középpontja az első, szabad végpontja. Beadandó a két sugár hosszának aránya, a vonalhálózatok forgásának mértéke, az, hogy forgás közben zsugorodjon vagy sem, végül letörölhetők az eddigi rajzok, vagy egymásra rajzolhatók

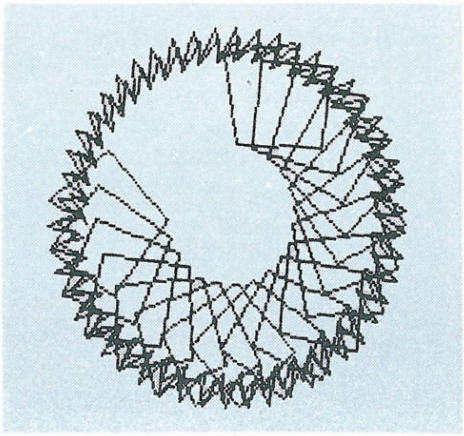
```
10 I=49152:CK=0
20 READA:CK=CK+A:PRINT";";CK:IF
   A=256THEN40
30 POKE I,A:I=I+1:GOTO 20
40 IFCK<>167356 THENPRINT
   "ERROR IN DATA STATEMENTS"
49152 DATA 173,234,207,141,236,207,173
49160 DATA 232,207,141,238,207,173,235
49168 DATA 207,141,237,207,173,233,207
49176 DATA 141,239,207,173,236,207,56
49184 DATA 237,230,207,141,240,207,173
49192 DATA 237,207,237,231,207,141,241
49200 DATA 207,176,24,173,240,207,73
49208 DATA 255,141,240,207,173,241,207
49216 DATA 73,255,141,241,207,238,240
49224 DATA 207,208,3,238,241,207,173
49232 DATA 238,207,56,237,228,207,141
49240 DATA 244,207,173,239,207,237,229
49248 DATA 207,141,245,207,176,24,173
49256 DATA 244,207,73,255,141,244,207
49264 DATA 173,245,207,73,255,141,245
49272 DATA 207,238,244,207,208,3,238
49280 DATA 245,207,56,173,236,207,237
49288 DATA 230,207,141,192,207,173,237
49296 DATA 207,237,231,207,13,192,207
49304 DATA 240,15,144,24,169,1,141
49312 DATA 222,207,169,0,141,223,207
49320 DATA 76,169,192,169,0,141,222
49328 DATA 207,141,223,207,76,169,192
49336 DATA 169,255,141,222,207,141,223
49344 DATA 207,56,173,238,207,237,228
49352 DATA 207,141,193,207,173,239,207
49360 DATA 237,229,207,13,193,207,240
49368 DATA 15,144,24,169,1,141,246
49376 DATA 207,169,0,141,247,207,76
49384 DATA224,192,169,0,141,246,207
49392 DATA141,247,207,76,224,192,169
49400 DATA255,141,246,207,141,247,207
49408 DATA173,230,207,141,248,207,173
49416 DATA228,207,141,250,207,173,231
49424 DATA207,141,249,207,173,229,207
49432 DATA141,251,207,56,173,244,207
49440 DATA237,240,207,141,211,207,173
49448 DATA245,207,237,241,207,13,211
49456 DATA207,176,15,173,240,207,141
49464 DATA252,207,173,241,207,141,253
49472 DATA207,76,40,193,173,244,207
49480 DATA141,252,207,173,245,207,141
49488 DATA253,207,173,252,207,141,254
49496 DATA207,173,253,207,141,255,207
49504 DATA56,173,252,207,237,244,207,
49512 DATA141,211,207,173,253,207,237
49520 DATA245,207,13,211,207,208,17
49528 DATA173,255,207,74,141,161,207
49536 DATA173,254,207,106,141,160,207
49544 DATA76,104,193,173,255,207,74
49552 DATA141,227,207,173,254,207,106
49560 DATA141,226,207,173,252,207,13
49568 DATA253,207,240,27,173,253,207
49576 DATA48,22,24,173,160,207,109
49584 DATA240,207,141,160,207,173,161
49592 DATA207,109,241,207,141,161,207
49600 DATA76,142,193,76,45,194,56
49608 DATA173,160,207,237,254,207,141
49616 DATA211,207,173,161,207,237,255
49624 DATA207,13,211,207,144,38,56
49632 DATA173,160,207,237,254,207,141
49640 DATA160,207,173,161,207,237,255
49648 DATA207,141,161,207,24,173,248
49656 DATA207,109,222,207,141,248,207
49664 DATA173,249,207,109,223,207,141
49672 DATA249,207,24,173,226,207,109
49680 DATA244,207,141,226,207,173,227
49688 DATA207,109,245,207,141,227,207
49696 DATA56,173,226,207,237,254,207
49704 DATA141,193,207,173,227,207,237
49712 DATA255,207,13,193,207,144,38
49720 DATA56,173,226,207,237,254,207
49728 DATA141,226,207,173,227,207,237
49736 DATA255,207,141,227,207,24,173
49744 DATA250,207,109,246,207,141,250
49752 DATA207,173,251,207,109,247,207
49760 DATA141,251,207,32,46,194,206
49768 DATA252,207,173,252,207,201,255
49776 DATA240,3,76,104,193,206,253
49784 DATA207,76,104,193,96,173,251
49792 DATA207,240,8,173,250,207,201
49800 DATA65,144,1,96,173,248,207
49808 DATA201,199,144,1,96,173,249
49816 DATA207,240,1,96,173,250,207
49824 DATA141,212,207,173,251,207,141
49832 DATA213,207,173,248,207,141,214
49840 DATA207,173,249,207,141,215,207
49848 DATA173,215,207,74,141,217,207
49856 DATA173,214,207,106,141,216,207
49864 DATA173,217,207,74,141,217,207
49872 DATA173,216,207,106,141,216,207
49880 DATA173,217,207,74,141,217,207
49888 DATA173,216,207,106,141,216,207
49896 DATA173,213,207,74,141,219,207
49904 DATA173,212,207,106,141,218,207
49912 DATA173,219,207,74,141,219,207
49920 DATA173,218,207,106,141,218,207
49928 DATA173,219,207,74,141,219,207
49936 DATA173,218,207,106,141,218,207
49944 DATA173,214,207,41,7,141,220
49952 DATA207,173,216,207,10,46,217
49960 DATA207,10,46,217,207,10,141
49968 DATA210,207,46,217,207,173,217
49976 DATA207,141,211,207,173,210,207
49984 DATA10,46,217,207,10,46,217
49992 DATA207,109,210,207,141,216,207
50000 DATA173,211,207,109,217,207,141
50008 DATA217,207,173,216,207,10,46
50016 DATA217,207,10,46,217,207,10
50024 DATA46,217,207,141,216,207,173
50032 DATA218,207,10,46,219,207,10
50040 DATA46,219,207,10,46,219,207
50048 DATA141,218,207,24,173,216,207
50056 DATA109,218,207,141,208,207,173
50064 DATA217,207,109,219,207,141,209
50072 DATA207,24,173,220,207,109,208
50080 DATA207,141,208,207,169,0,109
50088 DATA209,207,141,209,207,24,169
50096 DATA32,109,209,207,141,209,207
50104 DATA173,208,207,133,251,173,209
50112 DATA207,133,252,173,212,207,41
50120 DATA7,141,225,207,56,169,7
50128 DATA237,225,207,141,225,207,169
50136 DATA0,141,206,207,56,173,225
50144 DATA207,46,206,207,206,225,207
50152 DATA16,245,160,0,177,251,13
50160 DATA206,207,145,251,96,169,147
50168 DATA32,210,255,169,0,162,0
50176 DATA157,0,32,157,0,33,157
50184 DATA0,34,157,0,35,157,0
50192 DATA36,157,0,37,157,0,38
50200 DATA157,0,39,157,0,40,157
50208 DATA0,41,157,0,42,157,0
50216 DATA43,157,0,44,157,0,45
50224 DATA157,0,46,157,0,47,157
50232 DATA0,48,157,0,49,157,0
50240 DATA50,157,0,51,157,0,52
50248 DATA157,0,53,157,0,54,157
50256 DATA0,55,157,0,56,157,0
50264 DATA57,157,0,58,157,0,59
50272 DATA157,0,60,157,0,61,157
50280 DATA0,62,157,0,63,232,208
50288 DATA157,169,16,162,0,157,0
50296 DATA4,157,0,5,157,0,6
50304 DATA157,0,7,232,208,241,96,256
```

1/a. program

1. ábra



2. ábra



```
10 POKE53281,1:PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";
20 PRINT"▒  /──────────────\"
30 PRINT"          B                B"
40 PRINT"          |  SPIRALIZER    |"
50 PRINT"          |                |"
60 PRINT"          |                |"
70 PRINT"          \──────────────/"
80 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒PRESS SPACE";:POKE198,0:FORT=1TO300:NEXT
100 WAIT197,64,64:POKE53272,21:POKE53265,27:PRINT"▒";
110 Z=5:PRINT"SPEED(-50 TO 50)? ";"   ▒▒▒▒";:GOSUB500:K=Z
115 IFZ<-50OR Z>50THEN110
125 K=K-1
127 Z=35:PRINT"RADIUS (1 TO 60)? ";"   ▒▒▒▒";:GOSUB500:R=Z
130 R=R+13:S=1
135 Z=1:PRINT"SPIN(1 TO 18)? ";"   ▒▒▒▒";:GOSUB500
137 IFZ<1ORZ>18THEN135
140 A=1/Z:IFZ>1ANDZ<9THEN200
143 SM=1:M=2:PRINT"MOVEMENT OR DECREMENT (M/D)? ";"   ▒▒▒▒";
145 GETX$:IFX$="" THEN145
148 IFX$=CHR$(13) THEN M=0:SM=0:PRINT:PRINT"▒NONE";:GOTO200
150 IFX$="M" THEN SM=0:GOTO190
160 IFX$="D" THEN M=0:GOTO190
170 GOTO145
190 PRINTX$
200 PRINT:PRINT"CLEAR (Y/N)?
201 GETT$:IFT$="" THEN201
203 IFT$<>"N" THENPRINT"YES":PRINT"▒";:SYS50012
205 IFT$="N" THENPRINT"NO":SYS50120
210 W=1:Z=139:IFM=2THEN Z=80:IFA=1THEN W=5:M=1:Z=122
215 IFSM-A=0THEN W=5
220 IFA<1THEN K=K+A
230 C=0.001:IFA<1/9 THEN M=M/2:C=C/2
240 J=R:I=79-R
250 X0=Z+20:Y0=30:REM:SYS49152
260 REM MAIN LOOP
270 FORT=0TO6.2831/A*W STEP 0.06283
280 IFSM THEN J=R*S:I=79*S-J:S=S-C
290 X1=Z+T*M-SIN(T)*J+SIN(T*K)*I  :X1=X1+20
300 Y1=79-COS(T)*J-COS(T*K)*I:Y1=Y1+30
308 GOSUB1000:X0=X1:Y0=Y1
310 NEXT  :GOTO100
500 L0=0:L1=1:B$=""
510 PRINTH$;
520 GETA$:IF A$=""THEN520
530 IF A$="-"AND L0=0THEN PRINTA$;:B$=A$:L0=1:L1=2:GOTO510
540 IFA$= CHR$(13) AND L0>0THENZ=VAL(B$):PRINT:RETURN
550 IF A$=CHR$(13) THEN PRINTZ:RETURN
560 IF A$=R$ ANDL0>1THEN PRINTA$;:B$=LEFT$(B$,LEN(B$)-1):L0=L0-1:GOTO510
570 IF A$=R$ ANDL0>1THEN PRINTA$;:B$="":L0=0:GOTO510
580 IFL0>L1THEN510
590 IFA$<"0"ORA$>"9" THEN510
600 PRINTA$;:B$=B$+A$:L0=L0+1:GOTO510
900 POKE53272,21:POKE53265,27:PRINT"▒";:END
1000 POKE53272,PEEK(53272)OR8:POKE53265,PEEK(53265)OR32
1005 X0=INT(X0):IFX0<=255THEN POKE53220,X0:POKE53221,0
1010 IF X0>255THEN POKE53220,(X0-256):POKE53221,1
1030 POKE53222,Y0:POKE53223,0
1040 X1=INT(X1):IFX1<=255THEN POKE53224,X1:POKE53225,0
1050 IF X1>255THEN POKE53224,(X1-256):POKE53225,1
1060 POKE53226,Y1:POKE53227,0
1200 SYS49152 :RETURN
```

1/b. program

2. program

```
10 CLS 0
20 PRINT@200,"SPIRALIZER";
25 FORT=1 TO 1000:NEXT
30 CLS 3
35 K=5
40 INPUT "SPEED (-50 TO 50)";K$
41 K=VAL(K$)
42 IF K$=" "THEN K=5
45 IFK<-50ORK>50THEN40
47 K=K-1
48 R=35
50 INPUT "RADIUS (1-60)";R$
51 R=VAL(R$)
52 IF R$=" "THENR=35
55 IF R<1 OR R>60 THEN50
57 R=R+13:S=1
60 INPUT "SPIN 1-18";Z$
61 Z=VAL(Z$)
63 IF Z$=" "THENZ=1
65 IF Z<1OR Z>18THEN 60
70 A=1/Z:IF Z>1AND Z<9THEN200
80 SM=1:M=2
85 INPUT " MOVEMENT OR DECREMENT";X$
90 IF LEFT$(X$,1)<>"M" ANDLEFT$(X$,1)<>"
D"THENM=0:SM=0:PRINT:PRINT"NONE";:GOTO20
0
95 IF LEFT$(X$,1)="M"THENSM=0:GOTO 190
100 IFLEFT$(X$,1)="D"THENM=0:GOTO190
190 PRINT X$
200 PRINT:INPUT"CLEAR (Y/N)";T$
205 IF LEFT$(T$,1)<>"N"THENPRINT"YES";:E
=0:REMPUT CLEAR OF HIRES
210 IF LEFT$(T$,1)="N"THENPRINT"NO"

220 W=1:Z=139:IF M=2 THEN Z=80:IF A=1 TH
ENW=5:M=1:Z=122
230 IF SM-A=0THENW=5
235 IF A<1THENK=K+A
238 C=0.001:IFA<1/9THENM=M/2:C=C/2
240 J=R:I=79-R
250 X0=Z:Y0=20
260 REM MAIN LOOP
270 FOR T=0 TO 6.2831/A*W STEP 0.06283
280 IF SM THENJ=R*S  :I=79*S-J:S=S-C
290 X1=Z+T*M-SIN(T)*J+SIN(T*K)*I

300 Y1=79-COS(T)*J-COS(T*K)*I:Y1=Y1+20
310 GOSUB 1000:X0=X1:Y0=Y1
320 NEXT
330 A$=INKEY$:IF A$=" "THEN330
340 GOTO30
1000 SCREEN 1,1
1010 IF LEFT$(T$,1)="Y"ANDE=0THENPCLS:E=
1
1020 LINE (X0,Y0)-(X1,Y1),PSET
1030 RETURN
```

AND	C8	ON	88
AUDIO	A3	OPEN	9B
CIRCLE	B1	OR	C9
CLEAR	96	PAINT	B2
CLOAD	99	PCLEAR	AF
CLOSE	9C	PCLS	AB
CLS	A0	PCOPY	B6
COLOR	B0	PLAY	B8
CONT	94	PMODE	B7
CSAVE	9A	POKE	93
DATA	86	PRESET	AD
DEF	98	PRINT	87
DEL	A6	PSET	AC
DIM	8C	PUT	B4
DLOAD	B9	READ	8D
DRAW	B5	REM	82
EDIT	A7	RENUM	BA
ELSE	84	RESET	9F
END	8A	RESTORE	90
EXEC	A4	RETURN	91
FN	BE	RUN	8F
FOR	80	SCREEN	AE
GET	B3	SET	9E
GO	81	SKIPF	A5
IF	85	SOUND	A2
INPUT	89	STEP	C1
LET	8E	STOP	92
LINE	AA	SUB	BD
LIST	95	TABC	BB
LLIST	9D	THEN	BF
MOTOR	A1	TO	BC
NEW	97	TROFF	A9
NEXT	8B	TRON	A8
NOT	C0	USING	CD
OFF	C2		

1. táblázat. Parancsok, utasítások BASIC kódjai (69)

+	C3	↑	<7
–	C4	=	CB
'	83 (REM)	>	CA
*	<5	<	CC
/	<6	!	<E

2. táblázat. Jelek (10)

az egyes hálózatok. A változók értékének változtatásával rendkívül érdekes ábrák adódnak.

A VIC—20 változat csak Super Expander használatával valósítható meg! A C64 változat — a grafikus utasítások hiányában — egy grafikus utasításokat előállító programot is tartalmaz. Szívesen közölnénk C16, Spectrum, Primo, HT változatot is! Ilyeneket várunk!

Számszerűen összehasonlítva a változatokat, az Apple II változat 115, a VIC—20 változat 142, az Atari-változat 122, a Dragon-változat 78, a C64 változat 143 és a grafikus program 154 utasításból áll. Látható egyrészt az, hogy a Dragon-változat nagyon rövid, másrészt az, hogy a C64 még a grafikus programot figyelmen kívül hagyva is igen hosszú!

Az összehasonlítás és a géphasználat megkönnyítésére közlöm a C64 (*1/a.* és *1/b. program*)és a Dragon 64 (*2. program*) változat listáit és kulcsszó-kódtáblázatait. A Dragon rajztudásának szemléltetésére pedig bemutatok néhány, a rajzolóprogram készítette ábrát (*1-2. ábra*).

— nyi —

PROGRAMOK

Öt gép vizsgálata egy programmal

Az azonos című, ez évi 4. számunkban közölt program C16, PRIMO és TVC változatával könnyen rajzolhatunk nagy felbontású spirálokat és sokszögeket.

C16

Először be kell állítani, hány szögű sokszöget akarunk rajzolni (−50-től +50-ig). A „+" számoknál homorodik, a „−" számoknál domborodik az alakzat. A második adat a szögeket alkotó spirálok sugarát adja meg; minél nagyobb, annál pontosabb az alakzat, kisebb a sugár. A harmadik adat azt mutatja, hányszor forduljon körbe a spirál (1—18). Végül meg kell adnunk, hogy az alakzat mozogjon-e, csökkenő sugarú legyen-e és a képernyőt kívánjuk-e előtte törölni.

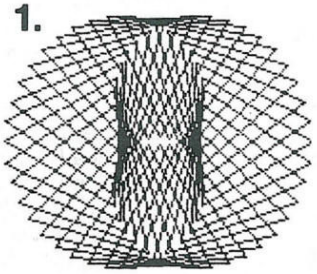
Az ábrák paramétereinek értékei a táblázatból olvashatók le.

BALÁZS GYÖRGY

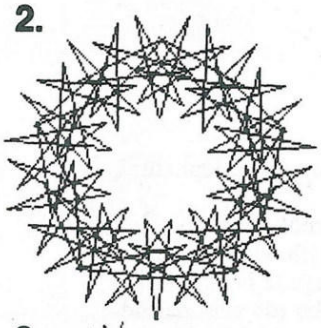
Változónév	K	R	F	X$	Y$	T$
Rajzsorszám						
1.	34	28	1	N	N	I
2.	45	40	1	N	N	I
3.	48	60	10	I	I	I
4.	35	1	1	N	N	I

```
10 PRINT"▽"
20 PRINT"▶▶▶▶▶▶▶▶▶▶▶▶SPIRALIZER"
30 FORT=1TO1000:NEXT
40 PRINT"▽"
50 K=5
60 INPUT"SPEED (-50- 50)";K$
70 K=VAL(K$)
80 IFK$=""THENK=5
90 IFK<-50ORK>50THEN60
100 K=K-1
110 R=35
120 INPUT"RADIUS (1- 60)";R$
130 R=VAL(R$)
140 IFR$=""THENR=35
150 IFR<1ORR>60THEN120
160 R=R+13:S=1
170 INPUT"SPIN (1- 18)";Z$
180 Z=VAL(Z$)
190 IFZ$=""THENZ=1
200 IFZ<1ORZ>18THEN170
210 A=1/Z:IFZ>1ANDZ<9THEN280
220 SM=1:M=2
230 INPUT"MOVEMENT OR DECREMENT";X$
240 IF LEFT$(X$,1)<>"M"AND LEFT$(X$,1)<>"D"THEN
0   M=0:SM=0:PRINT:PRINT"NONE":GOTO2
250 IF LEFT$(X$,1)="M"THEN SM=0:GOTO270
260 IF LEFT$(X$,1)="D"THENM=0:GOTO270
270 PRINTX$
280 PRINT:INPUT"CLEAR (Y/N)";T$
290 IF LEFT$(T$,1)<>"N"THENPRINT"YES";:E=0
300 IF LEFT$(T$,1)="N"THENPRINT"NO";
310 W=1:Z=160:IFM=2THENZ=80:IFA=1THENW=5:M=1:Z=140
320 IFSM-A=0THENW=5
330 IFA<1THENK=K+1
340 C=0.001:IFA<1/9THENM=M/2:C=C/2
350 J=R:I=80-R
360 X0=Z:Y0=20
370 FORT=0TO6.2831/A*WSTEP.06283
380 IFSMTHENJ=R*S:I=80*S-J:S=S-C
390 X1=Z+T*M-SIN(T)*J+SIN(T*K)*I
400 Y1=80-COS(T)*J-COS(T*K)*I:Y1=Y1+20
410 GOSUB450:X0=X1:Y0=Y1
420 NEXT
430 GETKEYA$
440 GRAPHIC0:GOTO40
450 GRAPHIC1
460 IFLEFT$(T$,1)="Y"ANDE=0THENGRAPHIC1,1:E=1
470 DRAW1,X0,Y0TOX1,Y1
480 RETURN
```

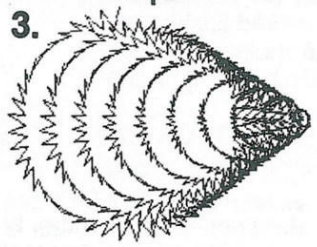
1.



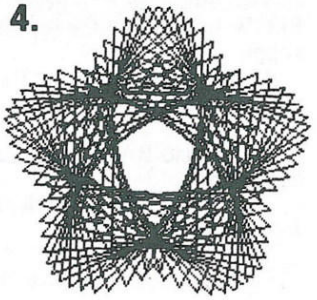
2.



3.



4.



PRIMO

A program két Primo változata közül az egyik a grafikus programcsomag használatával, a másik anélkül készült.

HORVÁTH ZOLTÁN

```
5 'NEV:spiralizer&G
10 CLS
20 PRINTTAB(16)CHR$(5)"SPIRALIZER"CHR$(21)
25 FOR T=1TO1000:NEXT
35 PRINT:PRINT:K=5
40 INPUT"SPEED (-50 TO 50)";K$
41 K=VAL(K$)
42 IF K$="" THEN K=5
45 IF K<-50 OR K>50THEN40
47 K=K-1
48 R=35
50 INPUT"RADIUS (1-60)";R$
51 R=VAL(R$)
52 IF R$="" THEN R=35
55 IF R<1 OR R>60 THEN 50
57 R=R+13:S=1
60 INPUT"SPIN 1-18";Z$
61 Z=VAL(Z$)
63 IF Z$="" THEN Z=1
65 IF Z<1 OR Z>18 THEN 60
70 A=1/Z:IF Z>1 AND Z<9 THEN200
80 SM=1:M=2
85 INPUT"MOVEMENT OR DECREMENT";X$
90 IF LEFT$(X$,1)<>"M" AND LEFT$(X$,1)<>"D
    " THEN M=0:SM=0:PRINT:PRINT"NONE";:GOTO200
95 IF LEFT$(X$,1)="M" THEN SM=0:GOTO 200
100 IF LEFT$(X$,1)="D" THEN M=0:GOTO 200
200 PRINT:INPUT"CLEAR (Y/N)";T$
205 IF LEFT$(T$,1)<>"N" THEN PRINT"YES";
    :E=0:'PUT CLEAR OF HIRES
210 IF LEFT$(T$,1)="N" THEN PRINT"NO"
220 W=1:Z=139:IF M=2 THEN Z=80:IF A=1 THEN
     W=5:M=1:Z=122
230 IF SM-A=0 THEN W=5
235 IF A<1 THEN K=K+A
238 C=0.001:IF A<1/9 THEN M=M/2:C=C/2
240 J=R:I=79-R
250 X0=Z:Y0=171
260 'MAIN LOOP
270 FOR  T=0 TO 6.2831/A*W STEP 0.06283
280 IF SM THEN J=R*S:I=79*S-J:S=S-C
290 X1=Z+T*M-SIN(T)*J+SIN(T*K)*I
300 Y1=112+COS(T)*J+COS(T*K)*I:Y1=Y1-20
310 GOSUB1000:X0=X1:Y0=Y1
320 NEXT
330 A$=INKEY$:IF A$="" THEN 330
340 GOTO35
1000 IF LEFT$(T$,1)="Y" AND E=0THEN CLS:E=1
1010 CMDJ X0,Y0,X1,Y1,1
1030 RETURN
```

```
5 'NEV:spiralizer3
10 CLS
20 PRINTTAB(16)CHR$(5)"SPIRALIZER"CHR$(21)
25 FOR T=1TO1000:NEXT
35 PRINT:PRINT:K=5
40 INPUT"SPEED (-50 TO 50)";K$
41 K=VAL(K$)
42 IF K$="" THEN K=5
45 IF K<-50 OR K>50THEN40
47 K=K-1
48 R=35
50 INPUT"RADIUS (1-60)";R$
51 R=VAL(R$)
52 IF R$="" THEN R=35
55 IF R<1 OR R>60 THEN 50
```

```
57 R=R+13:S=1
60 INPUT"SPIN 1-18";Z$
61 Z=VAL(Z$)
63 IF Z$="" THEN Z=1
65 IF Z<1 OR Z>18 THEN 60
70 A=1/Z:IF Z>1 AND Z<9 THEN200
80 SM=1:M=2
85 INPUT"MOVEMENT OR DECREMENT";X$
90 IF(LEFT$(X$,1)<>"M"ORLEFT$(X$,1)<>"m")A
   ND(LEFT$(X$,1)<>"D"ORLEFT$(X$,1)<>"d")THEN
      M=0:SM=0
92 PRINT:PRINT:"NOME";:GOTO200
95 IF (LEFT$(X$,1)="M" OR LEFT$(X,1)="m")
   THEN SM=0:GOTO 200
100 IF (LEFT$(X$,1)="D" OR LEFT$(X$,1)="d"
   ) THEN M=0:GOTO 200
200 PRINT:INPUT"CLEAR (Y/N)";T$
205 IF (LEFT$(T$,1)<>"N" OR LEFT$(T$,1)<>"
   n") THEN PRINT"YES";:E=0
210 IF (LEFT$(T$,1)="N" OR LEFT$(T$,1)="n"
   ) THEN PRINT"NO"
220 W=1:Z=139:IF M=2 THEN Z=80:IF A=1 THEN
   W=5:M=1:Z=122.
230 IF SM-A=0 THEN W=5
235 IF A<1 THEN K=K+A
238 C=0.001:IF A<1/9 THEN M=M/2:C=C/2
240 J=R:I=79-R
250 X0=Z:Y0=171
260 'MAIN LOOP
270 FOR  T=0 TO 6.2831/A*W STEP 0.06283
280 IF SM THEN J=R*S:I=79*S-J:S=S-C
290 X1=Z+T*M-SIN(T)*J+SIN(T*K)*I
300 Y1=112+COS(T)*J+COS(T*K)*I:Y1=Y1-20
310 GOSUB1000:X0=X1:Y0=Y1
320 NEXT
330 A$=INKEY$:IF A$="" THEN 330
340 GOTO35
1000 IF (LEFT$(T$,1)="Y" OR LEFT$(T$,1)="y
   ") AND E=0THEN CLS:E=1
1010 GOSUB1100
1030 RETURN
1100 ' VONAL HuZo RUTIN
1110 IF X0=X1 AND Y0=Y1 THEN SET(X0,Y0):
   GOTO1155
1115  IF ABS(X1-X0)>=ABS(Y1-Y0) THEN ST=
   (X1-X0)/ABS(X1-X0) ELSE 1135
1120 FOR XX=X0 TO X1 STEP ST
1125  YY=Y0+(XX-X0)*(Y1-Y0)/(X1-X0)
1130  GOSUB 1160:NEXT XX
1135 IF ABS(X1-X0)<ABS(Y1-Y0)THEN ST=(Y1-Y
   0)/ABS(Y1-Y0)
1140 FOR YY=Y0 TO Y1 STEP ST
1145 XX=X0+(YY-Y0)*(X1-X0)/(Y1-Y0)
1150 GOSUB 1160:NEXT YY
1155 RETURN
1160 IF XX>255 OR XX<0 OR YY>191 OR YY<0
   THEN 1670
1165 SET(XX,YY)
1170 RETURN
```

TVC

A következő módosításokkal programunk Dragon-változata a TVC-n is futtatható.

20 PRINT AT7,14,"SPIRALIZER";
A 25. sor lesz a 30.
A 40;50;60;85;200 soroknál INPUT helyett INPUT PROMPT és ; helyett : áll.
A 90,95,100,205,210,1010. sorokban LEFT$ (változó, szám) helyett változó (: szám) lesz.
A 290. sor végére kerül : X1=4* X1
300 Y1=161+COS (T) *J+COS (T*K) *
I : Y1=Y1-20:Y1=4 *Y1
310 1000 helyett 1010
1000 nincs
1010 PCLS helyett CLS van.
1020 PLOT, X0,Y0;X1, Y1
K=5
R=35
Z=15
D
Y
eset teljesen más képet adott, mint a COMPUTE! c. újságban APPLE II gépre közöltek.
Ha a DRAGON-változatnál a következő módosításokat beírjuk:
A 270. sor elejére
PMODE4,1:PCLS:SCREEN1,1
1000 nincs
akkor a kép ugyanaz lesz, mint a TVC-n.

SIMONYI ZSUZSA

HOMELAB 2 (AIRCOMP 16)

Morze adóprogram

A rádióamatőr-forgalomban jelentős feladatot tölt be Samuel Morse 1837-ben kidolgozott kódrendszere. Jelentősége abban van, hogy segítségével igen nagy távolságú kapcsolatokat lehet létrehozni annak veszélye nélkül, hogy a kiküldött jeleket a háttérzaj elnyomja.

Az amatőr adástechnikában az egyszerű kopogtatót már régen felváltották a gyorsbillentyűk, morzeírógépek és egyéb nagyobb teljesítményű berendezések. Napjainkban a számítástechnika rohamos elterjedésével lehetőség nyílt a továbblépésre. A számítógép nagy információtárolási kapacitása lehetővé teszi több oldalnyi gépelt szöveg egyidejű tárolását, gyors hozzáférését. A gép memóriájába bevitt adatok egyszerű programmal átkódolhatók morzejelekké. Ezt a feladatot hajtja végre ez a kis program.

A program három fő részre bontható. A BASIC programnak csak annyi a feladata, hogy ellássa a felhasználót a szükséges információval, és elindítsa a gépi kódú rutint. A BASIC rész működése egyszerű, ezért magyarázatára nem térek ki.

A gépi kódú rutin végzi a begépelt szöveg átkódolását, majd a morzekód kiküldését a magnókimenetre. A rutin első része lefordítja a bevett szöveget olyan speciális kódra, amit az adórutin külön vizsgálat nélkül képes értelmezni. Ha a leadandó szövegben olyan írásjelet talál, aminek a kódját az adó nem ismeri, jelzést ad, és a további fordítás megszakad. A lefordított szöveg a memóriába kerül, ahol az adórutin hozzáfér. A fordítás elvégzése után továbbadja a vezérlést az adóprogramnak.

Az adórutin végzi a lefordított kód morzejelekké alakítását, majd azok kivitelét a magnókimenetre. A rutin addig vár, míg meg nem nyomjuk az alsó SHIFT gombot. Csak ekkor kezdődik az adás. Erre azért van szükség, mert a szöveg begépelésekor keletkező hangimpulzusok kijutnak a kimenetre. A rádióadó „élesítése" után már nem lehet további szöveget bevinni. Az alsó SHIFT gomb megnyomásával — zavaró jel nélkül — indul az adás. A felső SHIFT gomb segítségével az adás bármikor megszakítható.

A program harmadik részében van tárolva az a kódtáblázat, ami alapján a fordító dolgozik. Egy-egy morzejelet három bájt ábrázol, amiből kettő vonatkozik a morzekódra. Az első bájt az írásjel ASCII kódja, a második bájt az elemi jelek számát tartalmazza, a harmadik bájt a morzejel, ahol 0 bit=pont és 1 bit=vonás. Például a Q betű morzejele ——.— A táblázatban a következőt találjuk: 51H 04H 0BH, ahol 51H a Q betű ASCII kódja, 04H: négy alapelemből álló jel következik és 0BH 00001011B, ami a Q betű bináris jelölése a morzejelnek megfelelően.

A morzejelek bájtos tárolási módjából adódóan egy jel nem lehet nyolc alapelemnél hosszabb. Kivétel ez alól az a morzejel, amely nyolc alapelemenként periodikus. Mivel a szabványos jelkészlet ilyen kódot nem tartalmaz, ezért egyes bájtcsoportok második bájtja nem lehet 08H-nál nagyobb.

A program használatba vétele előtt be kell állítani a H és L változók értékét. A H változó adja a jelek hangmagasságát, az L változó pedig egy alapelem hosszát. A megadott számokat a rutin számlálóként használja, így az FFH érték eredményezi a legmélyebb hangot, illetve a leghosszabb alapelemet (pontot).

A BASIC program listája:

```
10 HM=20480:PRINT "■↓↓", "MORSE ADOPROGRAM"
20 PRINT „UTASITASOK:↓"
30 PRINT ," LN=N JELHOSSZ (1—FFH)"
40 PRINT ," HN=N HANGMAGAS SAG(1—FFH)"
50 PRINT ," PN K=K ADASA N-SZER↓"
60 PRINT "(ALSO SHIFT)=ADAS MEGKEZDESE"
70 PRINT "(FELSO SHIFT)=ADAS-MEGSZAKITAS↓"
80 PRINT ," R=VISSATERES BASIC-BE↓↓"
90 PRINT "AZ UTASITASOK FORMAJA MEGEGYEZIK A MONITOR UTASITASAIVAL"
100 CALL HM: PRINT "■"
```

A gépi kódú program listája:

```
5000  32 3F 3F CD BE 01 CD 0B
      50 18 F5 FE 52 20 02 E1
      C9 FE 48 20 05 7D 32 91
      40 C9 FE 4C 20 05 7D 32
      90 40 C9 FE 50 C0 E5 D5
      DD E1 11 4D 52 DD 7E 00
      DD 23 FE 60 28 26 21 CD
      50 01 35 00 ED A1 20 06
      0E 02 ED B0 18 E7 23 23
      EA 3C 50 E1 F5 3E 0D EF
      3E 20 EF F1 11 BE 50 EF
      E7 C8 18 FB 12 3A FE 3A
      1F 38 FA 32 3E 3E 21 4D
      52 7E FE 60 28 2A 47 23
      4E 23 3C 28 1B 11 01 3C
      CB 09 30 02 1E 03 CD 9E
      50 11 01 3B CD 9E 50 10
      EC 1E 02 CD 9E 50 18 D9
      11 07 3B CD 9E 50 18 D1
      E1 2D C8 E5 18 C8 E5 2A
      90 40 7C 12 3D 20 FC 15
      7C 12 3D 20 FC 14 2D 20
      F1 1D 20 EB E1 3A FE 3A
      E6 20 C0 E1 E1 C9 2D 54
      20 4E 45 4D 20 49 53 4D
      45 52 45 4D 60
```

A hexadecimális kódtáblázat:

```
50CD 20 FF FF 30 05 1F 31 05
     1E 32 05 1C 33 05 18 34
     05 10 35 05 00 36 05 01
     37 05 03 38 05 07 39 05
     0F 2E 06 2A 2C 06 33 3F
     06 0C 3A 06 07 3B 06 15
     2D 06 21 3D 05 11 27 06
     1E 2F 05 09 22 06 12 41
     02 02 42 04 01 43 04 05
     44 03 01 45 01 00 46 04
     04 47 03 03 48 04 00 49
     02 00 4A 04 0E 4B 03 05
     4C 04 02 4D 02 03 4E 02
     01 4F 03 07 50 04 06 51
     04 0B 42 03 02 53 03 00
     54 01 01 55 03 04 56 04
     08 57 03 06 58 04 09 59
     04 0D 5A 04 03 5B 05 16
     5C 05 04 5D 04 07 5F 04
     0C 2B 05 0A 23 06 2D 00
```

KANICS MIKLÓS

Néhány képernyőkezelő rutin

A program először helyet biztosít a gépi kód számára, majd az öt rutint elhelyezi a 6639-es címtől kezdődően. A rutinok feladata sorrendben a következő:

1. EXT Ø utasítás: a teljes képernyő tartalmát jobbra lépteti.
2. EXT 1 utasítás: a teljes képernyő tartalmát balra lépteti.
3. EXT 2 utasítás: a teljes képernyő tartalmát felfelé lépteti.
4. EXT 3 utasítás: a teljes képernyő tartalmát lefelé lépteti.
5. EXT 4 utasítás: a papír és a tinta színét állítja be. A rutinok az ötödik kivételével a botkormány mozgatásával aktivizálódnak.

A függőleges mozgatás egyszerű blokkáthelyezés. Ebből következik, hogy a mozgatás gyorsítható.

Felfelé léptetés gyorsítása: a H regiszterpár tartalmának növelése 64-gyel vagy többszörösével, a BC regiszterpár tartalmának csökkentése ugyanannyival.

Lefelé léptetés gyorsítása: a HL és BC regiszterpárok tartalmának csökkentése 64-gyel vagy többszörösével.

Az ötödik rutinban a 1ØØ-as, illetve 1Ø1-es portcímekre más adatok írásával a papír, illetve a tinta színe megváltozik.

A gép mellé kapott segédletekből kiderül, hogy a memória 80 kbájt méretű lehet. Emiatt a VIDEO RAM-ot gépi kódban be kell „lapozni". Erre szolgál a DATA sorok elején szereplő 62,8Ø,211,2 kódsorozat. Magyarul: a 2-es portcímre 80-at kell beírni.

A rutinok könnyen átírhatók úgy, hogy a képernyőtartalomnak csak egyes részei mozduljanak el.

GULYÁS KIS PÉTER

```
10 LOMEM 6800
20 GRAPHICS 2
30 GOSUB 500
35 REM USRTAB táblázat feltöltése
40 POKE 33,239:POKE 34,25
50 POKE 35,8:POKE 36,26
60 POKE 37,33:POKE 38,26
70 POKE 39,51:POKE 40,26
80 POKE 41,69:POKE 42,26
85 REM Gépi kód betöltése
90 FOR A=0 TO 96
100 READ B
110 POKE 6639+A,B
120 NEXT A
130 DATA
    243,62,80,211,2,33,255,127,6,240,14,64,35,
    203,30,13,32,250,55,63,5,32,243,251,201
140 DATA
    243,62,80,211,2,33,0,192,6,255,14,64,43,203,
    22,13,32,250,55,63,5,32,243,251,201
150 DATA
    243,62,80,211,2,33,64,128,17,0,128,1,192,63,
    237,176,251,201
160 DATA
    243,62,80,211,2,33,191,191,17,255,191,1,
    192,63,237,184,251,201
170 DATA 243,62,5,211,100,62,8,211,101,251,201
180 SET DELAY 1:SET RATE 1
190 EXT 4
195 REM Billentyűzetfigyelés
200 A$=INKEY$
210 IF A$=CHR$(4) THEN EXT 0
220 IF A$=CHR$(19) THEN EXT 1
230 IF A$=CHR$(5) THEN EXT 2
240 IF A$=CHR$(24) THEN EXT 3
250 IF A$="v" THEN SET DELAY 30:SET RATE
    3: END
260 GOTO 200
500 PLOT
    300,300;300,700;700,700,700,300;300,300:PL-
    OT,
    304,304,PAINT
510 RETURN
```

ZX-Spectrum

A Beta basic és alkalmazása I.

A cikksorozat azt mutatja be, hogy a Beta basic 1.8 felhasználói program segítségével hogyan írhat magának bárki olyan hasznos programot, amely segítségére lehet a nyelvtanulásban vagy feljegyzések tárolásában, és amelyet a Beta basic nélkül csak gépi kódban tudna megírni. A program egy szótár. A beírt — tetszőleges hosszúságú, de max. 255 karakteres — szavak hosszától függően kb. 1000-1200 szót képes gyorsan visszakereshető módon tárolni.

A megtanult szavak egyenként kitörölhetők, helyet szabadítva fel az új ismeretlen szavak számára. A mindenkori szóállomány a programból magnóra menthető és onnan bármikor visszatölthető.

A Beta basic

A Beta basic az angliai Betasoft Ltd. fejlesztésének eredménye. Több változata terjedt el. Én most csak az 1.8 verzióval foglalkozom, mivel programot ezen írtam.

A Beta basic 1.8 34 új utasításszót és 19 új függvényt tesz hozzá a gép BASIC-jéhez. A kulcsszavak grafikus üzemmódban 1 billentyű lenyomására, a függvények az FN // segítségével érhetők el. Shift nélküli kulcsszó az EDIT, ami akkor íródik ki, ha Enter után nyomjuk meg a Ø-t. (A program írói úgy okoskodtak, hogy Ø-val kezdődő sor nem létezik.)

A Beta betöltése: LOAD "" ENTER. Ekkor betöltődik a háromsoros BASIC-betöltő és a 9,5 kbájt terjedelmű gépi kódú rész. Ha a betöltés sikeres volt, megjelenik a "Beta basic 1.8 c Betasoft 1984" felirat.

A BASIC-betöltő Ø-dik sora tartalmazza az új függvényeket; ez a programban marad, és LIST Ø parancsra láthatóvá válik. Az első sor a Beta esetleges másolását segíti, a második tölti be a gépi kódot, meghívja, és kitörli az első sort és saját magát.

A Betában írt BASIC-programok magnóra mentésével viszik magukkal a Ø-dik sort is, így visszatöltésnél használható a LOAD. Ha egy már meglévő Spectrum BASIC-ben írt programot akarunk átalakítani Betára, akkor ezt MERGE utasítással töltsük be, különben elveszítjük a függvényeket.

A Beta basicban írt programok betöltése előtt be kell küldenünk a Betát is, különben listázásnál az új kulcsszavak helyén csak magányos grafikus karaktereket találunk, és futtatásnál a program "C Nonsense in BASIC" üzenettel leáll.

Az 1.8-as verzió csak 48 k-s gépen futtatható. A gépi kódú rész az 55801-es címnél kezdődik és 65367-ig tart. Tehát az UDG terület használható. Ilyenkor azonban ki kell kapcsolnunk a Betát KEYWORDS Ø utasítással ("G" kurzor, 8). Ekkor minden grafikus karaktert használhatunk, kivéve a 8. billentyűn lévőt, amelyre a visszatéréshez van szükség, hogy kiadhassuk a KEYWORDS 1 utasítást.

Mivel a Beta basic leírása kb. 35 gépelt oldal, csak a szótárprogramban lévő utasításokat és függvényeket magyarázom el. Előbb azonban még egy ötlet. Ha már van egy Beta basicben írt programunk, amit sűrűn szeretnénk használni, de fárasztónak tartjuk, hogy a Betát előtte mindig külön szalagról be kell tölteni, megtehetjük a következőket. Betöltjük a gépbe a Betát és a programot. Programunkat kiegészíthetjük az alábbi sorokkal:

1 CLEAR 558ØØ : LOAD "beta" CODE : RANDOMIZE USR 58419.
2 CLS : DELETE 1 TO 2

Ezután fogunk egy kazettát, és kimentjük rá programunkat:

SAVE "program" LINE 1

Most kiadjuk a RANDOMIZE USR 599Ø4 parancsot, és közvetlenül a program után kimentjük kazettánkra a Beta gépi kódját is:

SAVE "beta" CODE 558Ø1,9567

Így ha legközelebb használni akarjuk programunkat, nincs más dolgunk, csak a kazettát betenni a magnóba és kiadni a LOAD "" parancsot. Ekkor betöltődik a program, Ø-dik sora viszi magával az új függvényeket, és autostarttal elindul. Első sora beállítja a RAMTOP-ot 558ØØ-ra, betölti a Beta basicet és aktivizálja (RAND USR 58419).

A második sor letörli a copyright feliratot, majd kitörli a programból az első sort és saját magát (DELETE 1 TO 2). Ezután már zavartalanul futhat a program, de csak abban az esetben, ha a futás közben semmilyen hibaüzenet vagy hibakezelés nem fordul elő. Mert ha igen, akkor a Beta végre szóhoz juthat, és boldogan követelheti a copyright üzenet után neki kijáró ENTER-t. Ilyenkor a program nem hibaüzenettel, hanem villogó "K" kurzorral áll le, és egy RUN parancs után már normálisan fut tovább. Ezért ha programunkban számítunk valamilyen hibaüzenetre, jobban teszszük, ha a lista elején később kitörlődő sorok valamelyikébe beírunk egy STOP utasítást.

A RAND USR 599Ø4 lényegében teljesen kikapcsolja a Beta basicet. Csak annyi köze lesz a géphez, hogy benne van a memóriában. A Beta akkor is kimenthető, ha nem adjuk ki ezt a parancsot, de VERIFY esetén még hibátlan felvételnél is "R Tape loading error" üzenetet kapunk. Gondolom, a Betasoft emberei ezzel is az illegális programmásolók életét akarták keseríteni.

Utasítások

DELETE ("G" 7)

Alakja DELETE sorszám TO sorszám.

MINI PLUS-Comp BASIC fordító

Talán mire megjelennek ezek a sorok, már nem is számít újdonságnak a hír: elkészült a PLUS-Comp, a Plus/4-es BASIC fordítóprogram kazettás verziója. (A lemezes változatot lapunk ez évi 4. számában ismertettük.)

Tudomásom szerint ez az első olyan Commodore BASIC fordítóprogram, amely nem mágneslemezzel, hanem kazettával működik. Bízom benne, hogy nem az utolsó, de erről majd később...

Miben különbözik a MINI a PLUS—Comp-tól? Elsősorban a lefordítható BASIC program nagyságában:

	Lemezes	Kazettás
A BASIC sorok max. száma	1600	310
A vezérlésátadó utasítások száma	2000	400
A programméret	kb. 25 kbájt kb. 12 kbájt	15 kbájt (nem graf.) 7 kbájt (grafika)

A másik különbség a munkaterület helye. A PLUS—Comp ugyanis mágneslemezen tárolja a munkafájlokat, a MINI esetében pedig a memóriában készül a futtatható P-kódú program. Kazettára csak a hibafájl íródik, ha ezt a fordítás előtt kérte.

A fordítóprogram, hasonlóan a „nagyobb testvérhez", elsősorban a CBM 2.0-ás BASIC verziót támogatja, de direktívák segítségével a 3.5-ös verzió is beépíthető a P-kódú programba.

Mivel a fordító felülről lefelé (top-down), balról jobbra (left-right), visszalépés nélküli algoritmussal dolgozik, ha az elemzett BASIC kifejezés legelső tagja nem BASIC 3.5-ös utasítás, direktívával meg kell jelölni, hogy a fordító helyesen építse be a P-kódú programba. Az így megjelölt sor nem tartalmazhat vezérlésátadást és FOR... NEXT ciklust (FOR, NEXT, STEP).

A direktívában álló kifejezés vagy sor szintaktikus elemzése csak futás közben folyik. Így fordításkor ezekben a részekben még maradhat hiba. Általánosan igaz, hogy csak alaposan tesztelt programokat érdemes lefordítani.

A lefordítandó BASIC programban a fordító nem használhatja az alábbi utasításokat:

CONT, LIST = a P-kódból származó korlátozás
ELSE
DO UNTIL/WHILE EXIT LOOP
RESTORE n
TRAP/TRON/TROFF/RESUME

A fejlett programozási technikáról való lemondás az ára a mintegy 4—12-szeres sebességnövekedésnek. A legnagyobb sebességet egész típusú FOR ciklusba ágyazott GOSUB-oknál mértem: ez kb. 25-szörös volt a BASIC-hez képest.

A fordító ezek alapján Commodore 64-es programok Plus/4-re átírt változatainak fordítására kiválóan alkalmas. A direktívakezelés elsajátítása kezdő programozónak több időbe telhet, mint azoknak, akik már dolgoztak fordítóprogramokkal. Ezt a kellemetlenséget egy készülő előfordító fogja megszüntetni, amely a lefordítandó BASIC programot automatikusan fordítható állapotba alakítja.

Tervezés alatt áll egy 15 kbájtnál nagyobb programok fordítására alkalmas Plus/4-es fordító, amely a munkafájlokat kazettára írva hozná létre a P-kódú programot. Természetesen ez több időbe telik, mint a memóriában való fordítás. Ha igény lesz rá, TURBO MPC néven kerül majd forgalomba.

Egy bonyolult, sok feltételt tartalmazó program futási ideje BASIC-ben 5 óra volt, ami a lefordított program esetében majd 30 percre csökkent. Ekkora különbséget már mindenképpen meg lehet érezni!

Mivel a MINI fordító készítésénél sok olyan problémát meg kellett oldani, amely fellépne Commodore 64 és C-1530 esetében is, ha megfelelő igény lesz rá, a C64-es verzió elkészítése is szóba jöhet. A leendő fordító érdekessége, hogy a lefordított program a BASIC munkaterületből kevesebb helyet foglal, mint BASIC-ben, függetlenül a BASIC program nagyságától — ellentétben a közismert C64-es BASIC fordítókkal, ahol a szükséges RUN—TIME modul egysoros program esetén is a program elé generálódik, minimálisan 17 blokknyi helyet foglalva a BASIC munkaterületből.

BÁRÓ CSABA

A ROM-ban tárolt aritmetikai rutinok használata

TVC

A TV—Computeren az assembler hiánya miatt csak rövid gépi kódú programok írására vállalkozhatunk. A sokat használt ciklusmagok gépi kódban való megírásával azonban csökkenthető a futási idő, és olyan programok is futtathatók lesznek, melyek tisztán BASIC-ben megírva igen lassúak. A ROM adatmozgató és aritmetikai rutinjaival gépi kódú programunkat szubrutinhívások sorozatává rövidíthetjük. Ezen a módon tetszőlegesen bővíthető a beépített függvénykészlet is.

Amint a gép leírásából kiderül, a TV—Computer az aritmetikai műveleteket a BASIC veremben végzi, és ez a helye a beépített függvények közötti paraméterátadásnak is. A verem utolsó elemére az IY regiszter mutat. A programírás során — különösen ciklusoknál — mindig figyelni kell ennek helyes állítására.

A lebegőpontos számokat a gép háromféle számábrázolási formátum szerint kezeli, amit az ábrákon szemléltetünk. Az ábrázolás bájtonként történik. Egy mantisszabájt két BCD számjegyet tartalmaz.

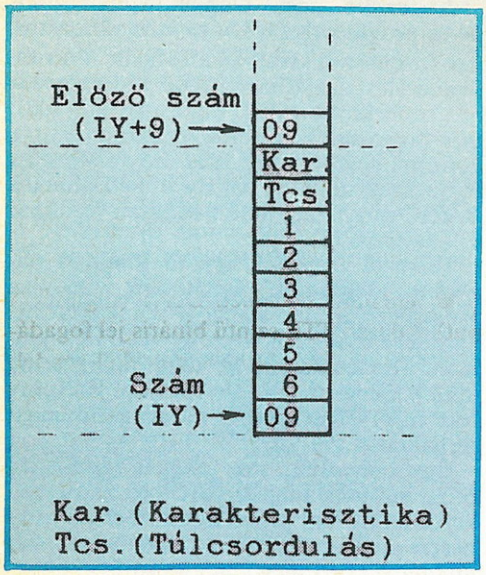
Az *1. ábrán* a verem látható. A számok magukban foglalják a túlcsordulásbájtot, amellyel együtt a verembe elhelyezett számok 14 decimális számjegy pontosságúak (egy bájt = két decimális számjegy).

Az ún. regiszter formátumú számoknál nincs túlcsordulásbájt *(2. ábra)*. A gép két beépített aritmetikai regiszterét (X és Y) a beépített függvények használják; a processzor HL és DE regiszterpárjainak megfelelő állításával több ilyen regiszter kialakítható a gép memóriájában.

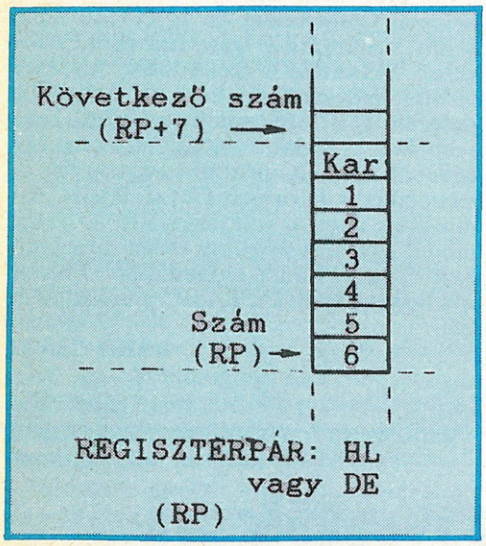
A *3. ábra* az X és Y belső aritmetikai regiszterek felépítését, a *4. ábra* a szimbólumtáblázatban (változótáblázatban) levő számok formátumát mutatja be. Ez utóbbiak már csak 10 decimális számjegy pontosságúak, ami egyben a megjelenítés pontossága is. A különböző formátumú számok közötti konverziót a ROM adatmozgató rutinjai végzik, a szubrutinhívások során erre nem kell ügyelni.

A szubrutinok vagy a szokásos módon, CALL utasítással, vagy táblázatból való kijelöléssel hívhatók. Ilyenkor a szubrutin táblázatbeli sorszámát kell megadni az alábbiak szerint:

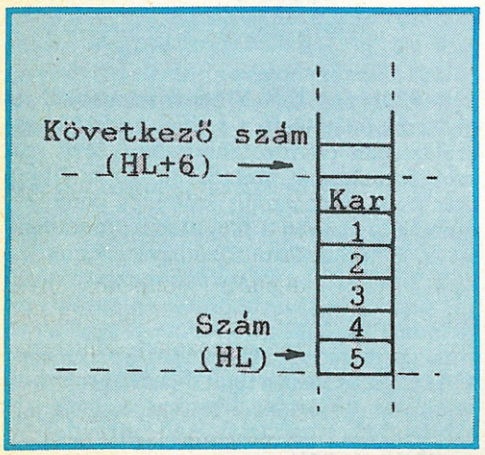
RST 18H	A táblázat aktivizálása
sorszám 1	Az 1-es sorszámú szubrutin végrehajtása
sorszám 2	A 2-es sorszámú szubrutin végrehajtása



1. ábra. Az aritmetikai verem



2. ábra. Egy tetszőlegesen kialakítható aritmetikai regiszter



3. ábra. Szimbólumtáblázat-elem

.
.
.

sorszám n Az n-edik sorszámú szubrutin végrehajtása után a szubrutinlista végét az jelzi, hogy a sorszám legnagyobb helyiértékű (128-as súlyú) bitje 1-est tartalmaz. A processzor az ezután következő kódot már utasításként értelmezi, és végrehajtja.

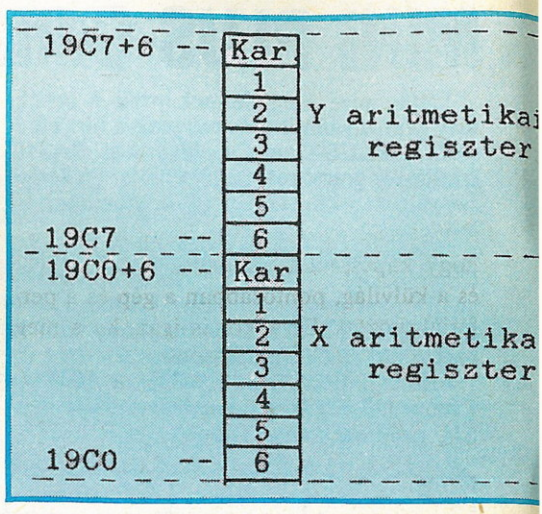
A hívható adatmozgató szubrutinokat a *táblázat* tartalmazza.

A szubrutin hívása	Hová	Honnan	Megjegyzés
CALL EAA2H	Verem (IY)	Regiszter (HL)	IY=IY−9
CALL EAC6G	Regiszter (DE)	Verem (IY)	IY=IY+9
CALL EAD5H	Regiszter (DE)	Verem (IY)	IY marad
CALL FA63H	Verem (IY)	Szimbólumtábla (HL)	IY=IY−9
CALL FB28H	Szimbólumtábla (HL)	Verem (IY)	IY=IY+9
RST 18H 06H	Verem (IY)	X regiszter	IY=IY−9
RST 18H 07H	Verem (IY)	Y regiszter	IY=IY−9
RST 18H 08H	X regiszter	Verem (IY)	IY marad
RST 18H 09H	Y regiszter	Verem (IY)	IY marad
RST 18H 0AH	X regiszter	Verem (IY)	IY=IY+9
RST 18H 0BH	Y regiszter	Verem (IY)	IY=IY+9

Példaként az alábbi programrészlet megcseréli az X és Y aritmetikai regiszterek tartalmát:

RST 18H
06H X regiszter tartalma a verembe (IY=IY−9)
07H Y regiszter tartalma a verembe (IY=IY−18)
0AH Veremből az X regiszterbe (IY=IY−9)
8BH Veremből az Y regiszterbe (IY=eredeti érték)

A 8BH hatásában megegyezik 0BH-val, de a szubrutinhívások végét jelzi.



4. ábra. A beépített X és Y aritmetikai regiszterek

A táblázatban nem szereplő további adatmozgató rutin, az
RST 18H
0CH
hatására a veremben levő szám még egyszer rákerül a veremre, önmaga fölé, az IY pedig a másolt számra mutat.

Végül az utolsó adatmozgató rutin a ROM-ban tárolt, megadott sorszámú konstanst a verembe teszi. A sorszámok 0—39-ig terjedhetnek; a hozzájuk tartozó konstansok értéke az *1. listán* levő programmal határozható meg.

A következő programrészlet a veremben egymás fölé helyezi a 4-es, 23-as és a 16-os decimális sorszámú konstansokat:

RST 18H
05H A szubrutin sorszáma
04H A 4-es sorszámú konstans a verembe kerül (IY=IY−9)
05H A szubrutin sorszáma
17H A 23-as sorszámú konstans a verembe kerül (IY=IY−18)
85H A szubrutin sorszáma befejezéskor
10H A 16-os sorszámú konstans a verembe kerül (IY=IY−27)
... A processzor ezt a kódot már utasításként kezeli

A *2. lista* az USR függvény felépítését mutatja.

BOTTYÁN ZOLTÁN

1. lista

```
 10 LOMEM 8000 : C=0 : D=0
 20 FOR A=0 TO 7
 30 READ B:POKE 7000+A,B:NEXT
 40 C=VARPTR(B)
 50 PRINT"Sorszám,Konstans"
 60 FOR A=0 TO 39
 70 POKE 7003,A: D=USR(7000,C)
 80 PRINT A,B
 90 IF INKEY$="" THEN 90
100 NEXT
110 DATA 229,223,133,39,225,
         195, 40,251
```

2. lista

```
        1B58H    PUSH HL
                 RST 18H
                 85H
POKE 7003,A→     ...
                 POP HL
                 JP FB28H
```

Átadták az 5000. Videoton iskola-számítógépet

Köpeczi Béla művelődési miniszter átadja a Sámsonkerti Általános Iskola igazgatónőjének az 5000. Videoton iskolaszámítógépet

Sötétkék ruhás, fehér inges, piros nyakkendős gyerekek sorakoznak a Sámsonkerti Általános Iskola udvarán. Ünnep ez a nap, hiszen a kis tanyasi iskolába ellátogatott Köpeczi Béla művelődési miniszter, hogy átadja az ötezredik Videoton iskola-számítógépet.

Az iskola a harmincas években épült. Ma négy tanteremben oktatnak. A településen, Sámsonkerten 350 család lakik, az iskolában 72 gyerek tanul. A tanyaközpont Debrecentől 14, Hajdúsámsontól 6 kilométerre terül el. Az oktatási feltételek bizony nem ideálisak. Osztatlan rendszerben tanítanak, vagyis az I—III., II—IV. és az V—VI., VII—VIII. osztályokat összevonták. Az osztatlan torna gondjai mellett dicséretes nevezetessége is van a sámsonkerti iskolának: a hattagú tantestület 1977-ben Állami díjat kapott! Valamennyien nők, s Debrecenből járnak a tanyaközpontba. Lelkesedésük, pedagógiai gyakorlatuk, szakmai tudásuk garancia lehet arra, hogy a sámsonkerti gyerekek mindazt megtanulják, amit ma az általános iskolás diákoknak tudniuk kell.

Dr. Andó Lászlóné igazgatónővel és Hramura Gábornéval, a tagiskola igazgatójával beszélgetve mindjárt a számítógépes szaktantermükbe invitálnak. A kis szobában úttörőjelvények díszelegnek a falon — egyben ez a helyiség az úttörőszoba is —, az asztalon viszont ott áll a számítógép és a televízió.

— Ősszel kaptunk egy Videoton gépet, nyáron dr. Szegedi Sándorné — a matematika-fizika szakos tanárunk — elvégezte a megyei pedagógiai intézet szervezte tanfolyamot, így ebben a tanévben már megismerhették a gyerekeink a számítógépet. Amikor a tanárnőnek ideje volt, foglalkozott az érdeklődő gyerekekkel, akik nagyon figyeltek, és indulásként néhány dolgot meg is tanultak. Sőt, télen Zánkán több tanulónk részt vett a számítógépes táborban. A felsősöket küldtük, hiszen náluk tapasztalható elsősorban a számítástechnika iránt érdeklődés.

Jócsák István hetedikes bizonyult eddig a legügyesebbnek.

— Délutánonként főleg játszottunk, amikor a tanárnő ráért. A gépet tudjuk kezelni, sőt Homlok Tibivel és Bordás Attilával már megpróbáltuk programozni is. Sajnos kevés a kész programunk.

— Örömünkre — mondja Andó Lászlóné — most egyszerre nem egy, hanem mindjárt két Videoton gépet kapunk, ugyanis a megyei tanácstól is érkezett számítógép. Ez lehetővé teszi számunkra, hogy a közeljövőben szakkört szervezzünk.

Gépkocsik sora tűnik elő a homokos dűlőúton. Megérkezik a miniszter, a megyei és a Debrecen Városi Tanács művelődési osztályvezetőinek kíséretében, és eljött Párizs György minisztériumi főtanácsos, a Tudományszervezési és Informatikai Intézet igazgatója, valamint Gantner János, a Videoton Számítástechnikai Gyárának igazgatója is. Az igazgatónő az egyik tanterembe kíséri vendégeit.

A köszöntő szavak után Köpeczi Béla művelődési miniszter mond beszédet.

— Örömmel és szeretettel köszöntöm a Sámsonkerti Általános Iskola pedagógusait és diákjait az ötezredik Videoton gyártmányú iskolaszámítógép átadása alkalmából. Büszkén állíthatjuk, hogy a magyar közoktatás lépést tartva a legkorszerűbb oktatási rendszerekkel — a szerény lehetőségekhez mérten — bár nagy anyagi áldozatokkal, de követni tudta a jobb gazdasági helyzetben lévő országok számítástechnikai okatásának fejlődési ütemét. Ezt bizonyítja, hogy az iskolák 1983-ban kapták az első számítógépeket, és ma, négy év után már csaknem huszonötezer gép működik: ezek közül mintegy tízezren az általános iskolások tanulhatnak.

Ma már kétségtelen, hogy a legkülönfélébb szakmák, tudományok, de a mindennapok sem nélkülözhetik a számítástechnikai ismereteket. És az is nyilvánvaló mindannyiunk számára, hogy ezeket a korszerű, a fejlődés, a haladás szempontjából nélkülözhetetlen ismereteket csak akkor szerezhetjük meg, ha ehhez a megfelelő eszközök léteznek. Sok tudományban lehet haladni, ha adott az érdeklődés, a szorgalom, a tehetség, de egy ilyen, technikához kötött tudást csak akkor szerezhetünk meg, ha az itt sem nélkülözhető érdeklődés, szorgalom, tehetség mellé a szükséges gépek és felszerelések is megvannak.

Szeretném jelképnek tekinteni, hogy ennek az ünnepélyes pillanatnak egy kis település a helyszíne. Jelkép abból a szempontból, hogy nincs olyan kicsiny helység az országban, ahová ne kellene eljuttatni a modern technikát nevelésünk, oktatásunk korszerűsítésére. Magyarországnak jó alapműveltségű, felkészült szakmai tudású, önmagukat becsülő és népüket szerető hazafiakra van szüksége, akik tudnak alkalmazkodni az új körülményekhez, alakítják is azokat. E cél megvalósítását kívánva adom át a Sámsonkerti Általános Iskolának az ötezredik Videoton gyártmányú iskolaszámítógépet.

Dr. Szekeres Antal, a megyei tanács művelődési osztályának vezetője ezt követően jelenti be, hogy a megyei tanács is ad egy Videoton számítógépet, a Videoton gyár pedig egy színes televíziót ajándékoz a sámsonkerti iskolásoknak. S hogy az öröm még teljesebb legyen, Párizs György programokat hozott magával, és azokat átadja az iskola igazgatónőjének.

Mit jelent ez a nap e kis település iskolájának életében? Elsősorban azt, hogy a jövőben lehetőség nyílik arra, hogy a gyerekek szervezetten, szakköri formában tanuljanak meg a számítógépekkel bánni, és most már ne csak játsszanak azokkal, hanem fontosságának megfelelően alkalmazzák a technikát. Hramura Gáborné arról álmodozik, hogy kollégái hamarosan a tanítási órákon is használják majd a számítógépeket.

Pillanatokon belül a szaktanterembe kerül a két gép. Hajdú-Bihar megyében ezután már 1500 számítógépet tarthatnak nyilván az iskolák. És ez nem kis szám. A számítógépes program a megyében megfelelő ütemben folytatódik . . .

FILEP TIBOR

A ROM-ban tárolt aritmetikai rutinok használata

TVC

Előző számunkban bemutattuk a TV-Computer ROM-jának adatmozgató szubrutinjait. A gép az adatmozgató rutinokhoz hasonlóan a négy alapműveletet is a veremben elhelyezett számok között végzi el a következő módon.

A veremben egymás fölé kell tenni a két operandust (OP1, OP2), amelyek között elvégzi a műveletet. Az eredmény az elsőnek betett (OP1) szám helyén keletkezik, az OP2 pedig elvész (definiálatlan).

A szubrutinok hívása az előzőekben már ismertetett módon, az RST 18H utasítást követően, a szubrutin sorszámának megadásával történhet. Ebben az esetben is igaz, hogy ha a sorszám legnagyobb helyiértékű (128-as súlyú) bitje 1, akkor ez az utolsó végrehajtandó szubrutin sorszáma; az ezt követő kódot a processzor már utasításként értelmezi és végrehajtja.

Az *1. táblázat* a szubrutinokat mutatja be. IY a veremmutató, a verem pedig a memóriában fentről lefelé terjed. Az utolsó sorban a 04-es sorszámú szubrutin a veremben levő szám előjelét változtatja ellenkezőjére.

A gép beépített BASIC függvénykészlete hívható a felhasználói gépi kódú programokból is, bár itt meg kell jegyezni, hogy e függvények használatával a gépi kódú programozás legfőbb előnye, a futási idő jelentős csökkenése nem érvényesül.

Ez a probléma a beépített függvények kikerülésével oldható meg úgy, hogy a nem használt függvényt a felhasználó által megírt szubrutinnal pótoljuk. Természetesen ehhez gyorsabb algoritmusra van szükség.

A gépi kódú felhasználói függvények a ROM ismertetett adatmozgató és aritmetikai szubrutinjainak alkalmazásával készülhetnek, de ha a futási idő nem elsődleges, akkor minden további nélkül hívhatják a *2. táblázatban* szereplő függvényeket. Hívás előtt az argumentumot a verembe kell helyezni, az eredmény ugyanitt keletkezik.

A *3. táblázatban* összefoglalásképpen megadjuk az RST 18H utáni sorszámokhoz tartozó szubrutinok memóriabeli címét, így azok CALL-lal gyorsabban hívhatók. Ekkor a program hossza nő. A „!"-et csak az RST 18H-val célszerű hívni.

A függvények egy numerikus értéket szolgáltatnak eredményül, és attól függően, hogy egy vagy több bemenő paraméterük van, felépítésükben lehet némi eltérés.

A függvényeket megvalósító gépi kódú rutinok szerkezete:

— az első két bájt az előző BASIC válto-

Szubrutin hivása	OP1 helye	OP2 helye	Művelet	Eredmény helye
RST 18H 00H	IY	IY-9	+	IY
RST 18H 01H	IY	IY-9	/	IY
RST 18H 02H	IY	IY-9	*	IY
RST 18H 03H	IY	IY-9	-	IY
RST 18H 04H	IY	nincs	CHS(X)	IY

1. táblázat

Függvény hivása	A függvény BASIC megfelelője
CALL EAECH	ABS(X)
CALL EBA7H	COS(X)
CALL EBB8H	EXP(X)
CALL ECAFH	INT(X)
CALL ED21H	LOG(X)
CALL EDE3H	PI
CALL EE3EH	SGN(X)
CALL EE5EH	SIN(X)
CALL EECAH	SQR(X)
CALL EFFBH	TAN(X)

2. táblázat

3. táblázat

RST 18H utáni sorszám	CALL-lal hivható cím	Funkció	Megjegyzés
00H	F493H	OP1=OP1+OP2	IY=IY+9
01H	F5FBH	OP1=OP1/OP2	IY=IY+9
02H	F512H	OP1=OP1*OP2	IY=IY+9
03H	F48EH	OP1=OP1-OP2	IY=IY+9
04H	F726H	OP1=CHS(OP1)	IY=IY
05H	EA82H	(IY)=N. KONST.	IY=IY-9
06H	EA9FH	(IY)=X reg.	IY=IY-9
07H	EA9AH	(IY)=Y reg.	IY=IY-9
08H	EAD2H	X reg.=(IY)	IY=IY
09H	EACDH	Y reg.=(IY)	IY=IY
0AH	EAC3H	X reg.=(IY)	IY=IY+9
0BH	EABEH	Y reg.=(IY)	IY=IY+9
0CH	FA92H	(IY-9)=(IY)	IY=IY-9
0DH	FB28H	(HL)=(IY)	IY=IY+9
0EH	EA68H	Lásd gépi kódú felhasználói függvények.	

```
7000 (1B58H) ..                    ←PEEK(5924)
             ..                    ←PEEK(5925)
             02                    függvény nevének a hossza
             53                    'S'
             48                    ,H,
             0A                    tipusbájt
             CD,68,EA (CALL EA68)  változó a verembe (IY)=A
             CD,B8,EB (CALL EBB8)  (IY)=EXP(A)
             DF       (RST 18)
             08                    Xreg.=EXP(A)
             05
             01                    (IY-9)=1
             06                    (IY-18)=EXP(A)
             01                    (IY-9)=1/EXP(A)
             03                    (IY)=EXP(A)-1/EXP(A)
             05
             01                    (IY-9)=1
             0C                    (IY-18)=1
             00                    (IY-9)=2
             81                    (IY)=(EXP(A)-1/EXP(A))/2
             C9       (RET)
```

1. lista

```
7000 (1B58H) ..
             ..
             05
             50                    'P'
             42                     R
             4F                     O
             42                     B
             41                    ,A,
             0A
                 LD   A, 96 }
                 CALL  FD54 }
          ......  CALL  F0A7
N-db változó     LD   A, A4 }
esetén N-1-szer  CALL  FD54 }
alkalmazni ......CALL  F0A7 }
                 LD   A, 95 }
                 CALL  FD54 }
                 RST  18
             0C
             02
             0A
             0C
             02
             06
             80
                 RET
                                PROBA ( X , Y )
```

2. lista

zóra vagy függvényre mutató cím, ezt a függvény memóriabeli elhelyezése után POKE utasításokkal kell beállítani,

— a függvény nevének hossza (1 bájt),

— a függvény nevét alkotó ASCII karakterek,

— típusbájt, amely jelen esetben 0AH (10),

— a függvény argumentumát a verembe elhelyező rutin, amely a következő lehet: egy bemenő paraméter esetén CALL EA68H *(1. lista);* több bemenő paraméter esetén a *2. lista* szerinti programrészletet kell alkalmazni, N darab változó esetén N−1-szer,

— maga a függvényt megvalósító programszegmens,

— Return.

Az így megalkotott felhasználói függvényt a BASIC beépített függvényeihez hasonlóan a nevével és a numerikus paraméter(ek) megadásával lehet hívni.

A függvényeket a LOMEM utasítással védett területen célszerű elhelyezni. Egy függvény esetén a gyári leírásban szereplő CHAIN rendszerváltozó értékét kell POKE utasításokkal elhelyezni a függvény első két bájtjában, az első bájt címét (kezdőcím) pedig be kell tenni a CHAIN változóba. A CHAIN az 5924, 5925-ös decimális címen van.

Több függvény fejlécét úgy kell összeláncolni, hogy az mindig az előző függvény kezdőcímét tartalmazza. A CHAIN értékét az első függvény fejlécébe kell bevinni, a CHAIN pedig az utolsó függvény kezdőcímét tartalmazza.

Az 1. listán látható függvénynek egy bemenő paramétere van, és azt az EA68H címen lévő szubrutin teszi a verembe. A program az (EXP(X)−EXP(−X))/2 függvényt valósítja meg, és SH(X) alakban hívható. A hozzá tartozó BASIC programrészletet a *3. lista* tartalmazza.

A 2. lista a két bemenő paraméteres függvény felépítését mutatja (PROBA(X,Y)), amelyben nem az EA68H szubrutin teszi a változókat a verembe, hanem a megjelölt programrészlet, amelyet annyiszor kell alkalmazni, ahány vesszővel vannak elválasztva a változók. A nyitó zárójel kódja a hexadecimális 96, a vesszőé az A4, a bezáró zárójel kódja pedig a 95. A programrészlet az SQR(X^2+Y^2) függvényt számolja ki (komplex szám abszolút értéke). Ez a függvény a *4. lista* BASIC programjával hozható létre.

BOTTYÁN ZOLTÁN

3. lista

```
10 LOMEM 8000
20 FOR I=0 TO 24
30 READ A: POKE 7000+I,A:NEXT
40 POKE 7000,PEEK(5924):POKE 7001,PEEK(5925)
50 POKE 5924,88: POKE 5925,27
60 INPUT PROMPT"X=?":A
70 PRINT SH(A):GOTO 60
80 DATA 0,0,2,83,72,10,205,104,234,
       205,184,235,223,8,5,1,6,1,3,
       5,1,12,0,129,201
```

4. lista

```
10 LOMEM 8000
20 FOR I=0 TO 41
30 READ A: POKE 7000+I,A: NEXT I
40 POKE 7000,PEEK(5924):  POKE 7001,PEEK(5925)
50 POKE 5924,88 :POKE 5925,27
60 INPUT PROMPT"A , B =?":EGY,KETTO
70 PRINT PROBA(EGY,KETTO) :GOTO 60
80 DATA 0,0,5,80,82,79,66,65,10,
       62,150,205,84,253,205,167,240,
       62,164,205,84,253,205,167,240,
       62,149,205,84,253,223,12,2,10,
       12,2,6,128,205,202,238,201
```

TVC

Univerzális perifériakártya

Ahogy a Magazin 1987/12. számában olvastuk, már több mint 5000 Videoton számítógépet adtak el az országban. Sajnos a gépekkel kapcsolatos hardver és szoftver ismeretek „eladását" nem szorgalmazza a Videoton, ezért az ilyen információk sokakat érdekelhetnek. Ezúttal csupán egy kis részt tárgyalunk: azt, hogy hogyan lehet a gép négy csatlakozósávjába helyezhető perifériakártya illesztését elvégezni. A gép kezelési útmutatója tartalmazza a négy 2×43 pólusú csatlakozó jeleinek kiosztását, de sajnos magyarázatok nélkül.

A perifériák illesztéséhez egyrészt az adatvezetékekre, másrész a címkiválasztó hálózatra van szükség. A TVC-ben ezt igen ötletesen úgy oldották meg, hogy a címet tartalmazó nyolc vezetékből a felső négyet belsőleg dekódolták, és ebből négyet, az 1XH, 2XH, 3XH, 4XH címeken megszólalókat (H hexadecimális alakot jelöl) kivezették a csatlakozósávra (A39 pontok, jelük: SLOT 0—3). A gépet a billentyűzet felől nézve az első 1XH kártyacím, a SLOT 0 a jobb oldali legszélső kártya, és utána sorban következik a többi.

Az alsó négy címvezeték 16-féle kombinációjával (X=0H ... FH) tehát összesen 16 perifériacímet hozhatunk létre egy kártyahelyen belül.

Az elmondottak illusztrálására a 8 digitális be- és 8 digitális kimenetet tartalmazó perifériakártya elvi kapcsolási rajzát közöljük az *1. ábrán.*

A rajzon szereplő téglalappal bekeretezett kivezetések jelnevei megegyeznek a csatlakozókiosztásnál leírtakkal (lásd a kezelési útmutatót), így az adott kártyakivezetési pont onnan azonosítható.

Az U1 jelű VAGY kapukból kialakított hálózat az U2 áramkörrel együtt végzi a teljes címdekódolást. Az NBRD és NBWR jelek felhasználásával — az U2 felső négy kimenete — a bemeneti kapuk kiválasztását végezhetjük el, az alsó négy segítségével a kimeneti tárolókba írhatunk. Az így dekódolt perifériacímek rendre K0H—K3H, ahol K=1,2,3,4, attól függően, hogy melyik csatlakozóba helyezzük a kártyát.

Az összehuzalozott kártyát az 1. csatlakozóba helyezve, a következő BASIC programmal ellenőrizhetjük a KI1—KI8 kimenetek működését:

```
10 FOR I=0 TO 255
20 OUT 16,1
30 NEXT I
40 GOTO 10
```

A program futásakor a kimeneteket oszcilloszkóppal nézve, mindig felezett frekvenciájú, szimmetrikus négyszögjeleket látunk.

A BE1—BE8 bemenetekre kapcsolt logikai jeleknek megfelelő decimális érték folyamatosan az alábbi programmal jeleníthető meg:

```
10 A=IN(16)
20 PRINT A
30 GOTO 10
```

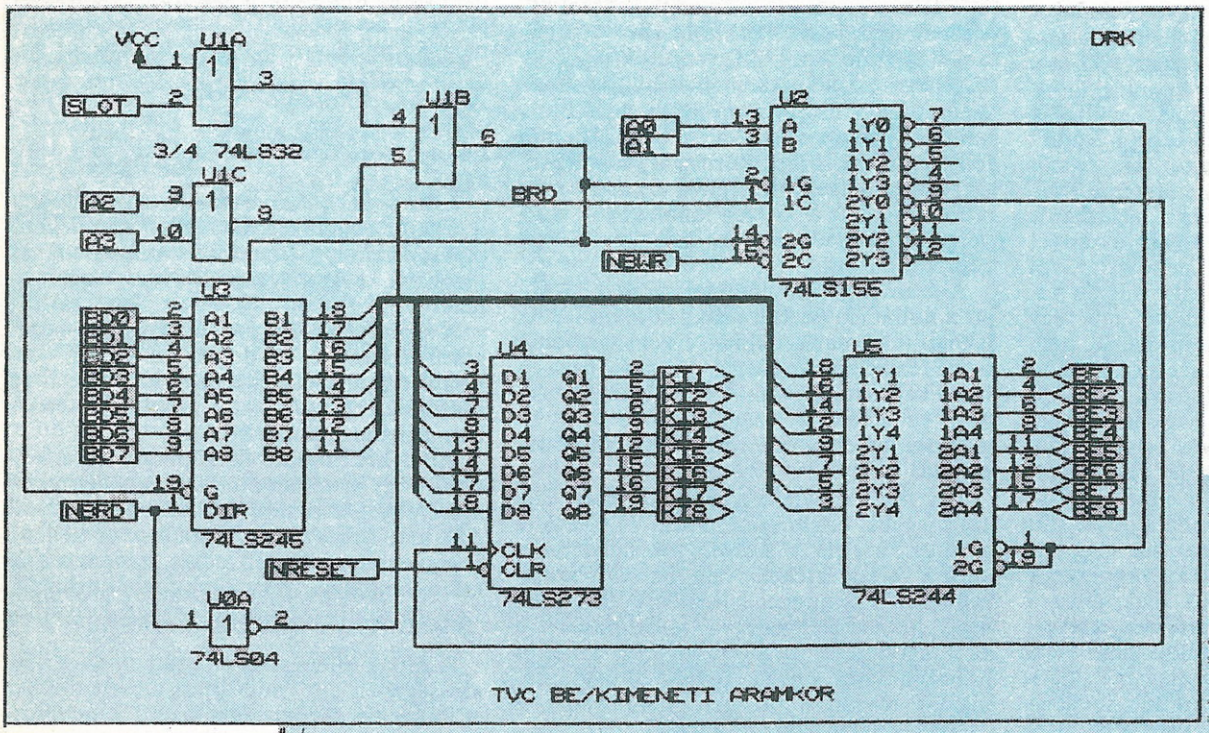
Az U4 és U5 jelű áramkörökből maximálisan 4-4 darab használható (azaz 32 be- és kimenet), amelyek vezérlésére értelemszerűen az U2 többi kimenetét használhatjuk fel.

Mivel minden bizonnyal nehézséget okozhat ennek vagy az ehhez hasonló kártyának az elkészítése, ezért a *2. ábrán* egy „csupalyuk" univerzális kártya NYÁK-rajzát is közöljük, amin az ilyen — és a későbbi számokban ismertetésre kerülő — kapcsolások könnyen összehuzalozhatók.

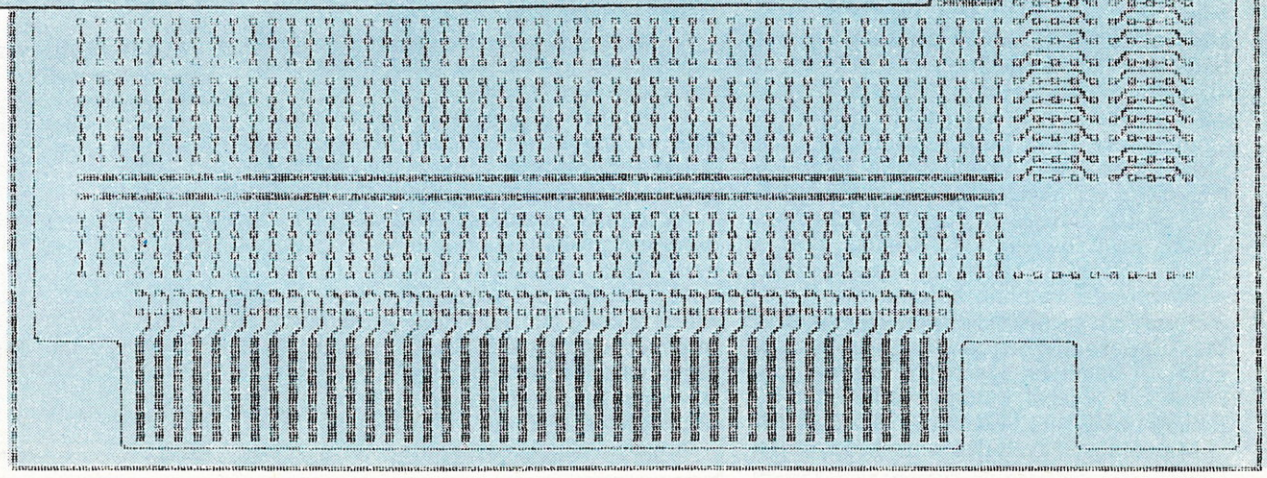
Célszerű az integrált áramköröket foglalatba helyezni és úgy behuzalozni, mert egy tokcsere sokkal egyszerűbben elvégezhető.

A szerkesztőségen keresztül ez az univerzális kártya 740 forintos áron megrendelhető.

DR. KÓNYA LÁSZLÓ



1. ábra. TVC perifériakártya kapcsolási rajza



2. ábra. TVC univerzális kártya NYÁK-terve

Spectrum INPUT rutin TVC-re

A ZX81 és a ZX-Spectrum BASIC-jének egyedülálló tulajdonsága, hogy az INPUT-ra adott válasz tetszőlegesen bonyolult kifejezés is lehet. A válasz beadása után az interpreter kiértékeli a kifejezést, és ezután folytatódik a program futása. Ez a módszer nagyon jó többek között függvénykiértékelő és -ábrázoló program készítéséhez. Az alábbi program ezt valósítja meg TV-Computeren.

Nézzük a működési elvét *(1. lista)*. A TV-BASIC a függvényeket nem tokenizálja, hanem ASCII kód formában helyezi el, a BASIC sorban tokenizálódnak viszont a műveleti jelek. A program az INPUT-ba beadott kifejezéseket átalakítja úgy, hogy a kisbetűket nagybetűkre (17-es sor), a műveleti jeleket megfelelő tokenjükre cseréli (5—16. sor), majd elhelyezi egy BASIC sorban (ez a 2. sor). Amennyiben a beadott kifejezés szintaktikailag hibás, hibajelzést kapunk (*** Not understood).

A program használata: az első két sorát különös gonddal gépeljük be, különben futáskor felülírja önmagát! Saját programunkat a 20. sortól kezdve írhatjuk. Az INPUT rutin meghívása:

— ha numerikus kifejezést akarunk bekérni: VT=0: GOSUB 3.

Az eredmény az A változóba kerül.

— Ha sztring kifejezést akarunk bekérni: VT=1: GOSUBA 3.

A kifejezésben az idézőjel helyett aposztrófot kell használni. Az eredmény az AS változóba kerül.

— GOSUB2 utasítással a kifejezés bármikor újra kiértékelhető.

```
   1 DIMS$*254,A$*254:GOTO20
   2 A=.........................................................................
.........................................................................
.........................................................................
.........................................
   3 INPUTPROMPT"KIFEJEZES:":S$:X=6667:IFS$=""THENPRINT"HIBAS!!":GOTO3
   4 IFVT=1THENPOKE6665,ORD("$"):ELSEPOKE6665,ORD(" ")
   5 FORI=TOLEN(S$)
   6 IFS$(I)="("THENS$(I)=CHR$(150)
   7 IFS$(I)=")"THENS$(I)=CHR$(149)
   8 IFS$(I)="="THENS$(I)=CHR$(154)
   9 IFS$(I)="+"THENS$(I)=CHR$(152)
  10 IFS$(I)="-"THENS$(I)=CHR$(162)
  11 IFS$(I)="*"THENS$(I)=CHR$(168)
  12 IFS$(I)="/"THENS$(I)=CHR$(161)
  13 IFS$(I)="↑"THENS$(I)=CHR$(159)
  14 IFS$(I)="&"THENS$(I)=CHR$(151)
  15 IFS$(I)=":"THENS$(I)=CHR$(253)
  16 IFS$(I)="'"THENS$(I)=CHR$(34)
  17 IFS$(I)>"Z"ANDS$(I)<="z"THENS$(I)=CHR$(ORD(S$(I))-32)
  18 NEXTI:FORI=1TOLEN(S$):POKEX,ORD(S$(I)):X=X+1:NEXTI:POKEX,253:POKEX
+1,217:POKEX+2,253:GOTO2
  20 REM *** PROGRAM ELEJE ***
```

1. lista

```
30 REM *** SPECTRUM INPUT DEMO 2.  ***
40 REM ***     FUGGVENYTABLAZAT     ***
50 VT=0:GOSUB3:FORX=0TO100STEP5
60 GOSUB2:PRINT"X=";X,"Y=";Y
70 NEXTX:GOTO20
```

2. lista

```
30 REM *** SPECTRUM INPUT DEMO 1.  ***
40 VT=0:PRINT"NUMERIKUS ";:GOSUB3:PRINTA
50 VT=1:PRINT"STRING ";:GOSUB3:PRINTA$
60 GOTO20
```

3. lista

A *2. és 3. listán* látható program jól szemlélteti a rutin használatát. A második listán látható programmal egy függvénytáblázatot készíthetünk. A 3. program numerikus és sztring típusú kifejezés használatát mutatja be.

NAGY ISTVÁN

Memóriaterület mentése C64-en

Az alábbi — szubrutinként is alkalmazható — program igen hasznos lehet olyankor, amikor a gép memóriájából adatokat vagy gépi kódú programokat kívánunk adathordozóra menteni. Természetesen ezt nemcsak lemezre menthetjük, hanem bármilyen más perifériára is. Ilyenkor azonban a 3. sort módosítanunk kell. A kezdő és a végcím beállítására az 1. sorban nyílik lehetőségünk. Mivel a létrejött fájl PRG formátumú, a visszatöltése is igen egyszerű.

SURÁNYI PÉTER

```
1 INPUT"KEZDET";KEZ:INPUT"VEG";VEG
2 LO=KEZ AND 255 : HI=INT(KEZ/256)
3 OPEN 1,8,1,"PROGRAMNEV"
4 PRINT#1,CHR$(LO);CHR$(HI);
5 FOR X=KEZ TO VEG : A=PEEK(X)
6 PRINT#1,CHR$(A);:NEXT
7 CLOSE 1
```

TVC GÉPI KÓD BEÍRÁSA – GYORSAN!

Gyakran okoz bosszúságot, ha a nehezen beírt rutin az első futtatásnál elszáll. Szintén nem tartozik az örömök közé a memóriacímek kétbájtos alakra történő átírása sem.

Az alábbi program meggyorsítja az újraindítást, az átszámítást pedig feleslegessé teszi. Egymás után több rutin is beírható, kezdőcímeiket a program kiírja a képernyőre. Lehetőség van javításra, adatátírásra. A betöltéskor pedig ellenőrizhető a gépelés helyessége.

A program használata

A kódokat decimális számokkal kell beírni. A gép jelzi, hogy melyik billentyűkkel kell a javítást, illetve a befejezést végrehajtani. A kódok bevitele előtt lehetőség van a LOMEM rendszerváltozó átírására. Amennyiben nem kell megváltoztatni, „A BASIC kezdőcíme:" kérdésre csak a RETURN gombot kell megnyomni. Ezután kezdődhet a kódok bevitele. Itt is érvényes, hogy ha az adott memória tartalma megfelelő, elegendő a RETURN gombot megnyomni.

Javítás

A „j" betű beírása után a RETURN gombbal lehet a kurzort a megfelelő címig felfelé léptetni. Ha az adott cím már nem látható, a program csökkenő sorrendben belépteti a címeket a képernyő tetején. A kívánt cím elérésekor a jó adat beírása után a RETURN gomb nyomogatásával vissza lehet térni a javítás előtti címre. A javítás közben átlépett adatok változatlanok maradnak. A program szöveget is elfogad, de azt 0-nak tekinti. A 256 és 65 535 közötti számokat automatikusan felbontja alsó és felső bájtra, majd úgy is írja ki a képernyőre. 0-nál kisebb, illetve 65 535-nél nagyobb szám beírása esetén a program az adatot nem fogadja el. Ettől az egytől eltekintve a program téves beírás ellen nincs védve. A beírás befejezése után megjelennek a rutinok kezdőcímei.

Figyelem! A 201-es adatot is RET utasításnak érzékeli, ezért az azt követő címet is kiírja mint kezdőcímet.

RUN paranccsal és a RETURN gomb nyomogatásával lehet végig ellenőrizni a beírás helyességét. NEW parancs után csak a BASIC program törlődik.

GULYÁS KIS PÉTER

```
10 CLS: OUT 101,12
50 PRINT AT 3,0:"Javítás a 'j' és a
   RETURN"
55 PRINT AT 6,0:"Betöltés befejezése
   a 'v' és a RETURN billentyűkkel
   történik."
60 PRINT AT 9,0:"Rutin hívása:
   x=USR (cím)"
65 PRINT AT 12,0:"Nyomj meg egy
   gombot!"
70 GET
75 CLS
100 INPUT PROMPT "A Basic
    kezdőcíme:": A$: ! Helyfoglalás a
    gépi kódnak
130 A=VAL (A$)
135 IF A$="" THEN 170
140 POKE
    5920,(A/256—INT(A/256))*256:
    POKE 5921,INT(A/256)
160 LOMEM A
170 POKE 5895,255:DIM Z(20)
200 CLS:PRINT"cím,"e.tart.","új tart"
220 B=6639
240 PRINT B,PEEK(B);" ",
260 INPUT PROMPT"":C$
265 IF C$="v" THEN 600
267 IF C$="j" THEN B=B-1:PRINT
    CHR$
    (5);CHR$(5);CHR$(25);CHR$(5):
    GOTO 240
270 C=VAL (C$)
272 IF C<0 OR C>65535 THEN PRINT
    CHR$
    (5);:GOTO 240
274 IF C>255 THEN 500
275 IF C$="" THEN
    C=PEEK(B):PRINT CHR$
    (5);CHR$(9);CHR$(9);CHR$(8);C
280 POKE B,C
300 B=B+1
310 GOTO 240
500 POKE B, (C/256—INT(C/256))*256
510 PRINT CHR$(5),,CHR$(8);
    PEEK(B);
      "     "
520 PRINT B+1, PEEK(B+1),
530 B=B+1
540 POKE B, INT(C/256)
550 PRINT CHR$(8);PEEK(B)
560 GOTO 300
600 I=1:Z(I)=6639:K=201:CLS:PRINT
    "Rutin kezdőcímek:":PRINT
605 PRINT Z(I)
610 FOR R=6639 TO B-2
620 IF K=PEEK(R)
    THEN I=I+1:Z(I)=R+1:
    PRINT Z(I)
640 NEXT R
650 END
```

„RADÍROZÁS" RÁKÉRDEZÉSSEL

Az alábbi rövid, gyors, gépi kódú rutin (1. lista) a felesleges BASIC programrészek törlését oldja meg ZX-Spectrumon. A mellékelt kis program (2. lista) a blokktörlő rutint helyezi el a memóriában (GOTO 9998), majd megkérdezi a törlendő blokk kezdősorának számát és a blokk utáni első sor számát.

POLGÁR GYULA

```
LD      HL,0        az első törlendő sor száma
CALL    #196E
PUSH    HL
EX      DE,HL
PUSH    DE
LD      HL,0        az első megmaradó sor száma
CALL    #196E
POP     DE
CALL    #19DD
POP     HL
CALL    #19E8
RET
```

1. lista

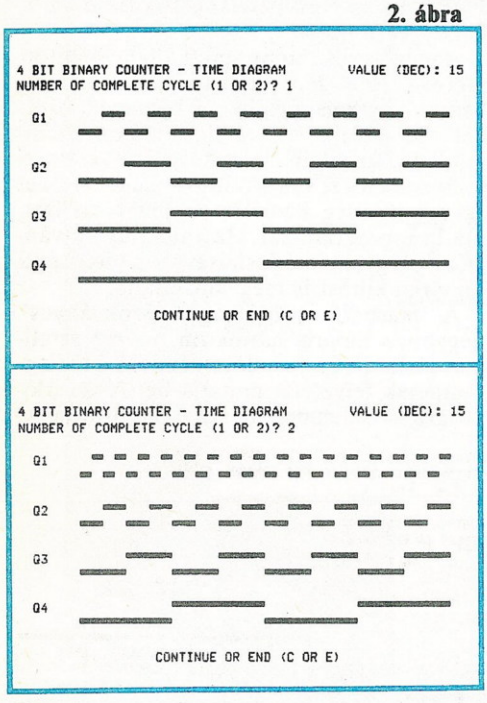
```
60000 33 0 0......... ld hl,0
60003 205 110 25....call 6510
60006 229 ........... push hl
60007 235 ........... ex de,hl
60008 213 ........... push de
60009 33 0 0......... ld hl, 0
60012 205 110 25....call 6510
60015 209 ........... pop de
60016 205 221 25.....call 6621
60019 225 ........... pop hl
60020 205 232 25.. call 6632
60023 201 ........... ret
```

2. lista

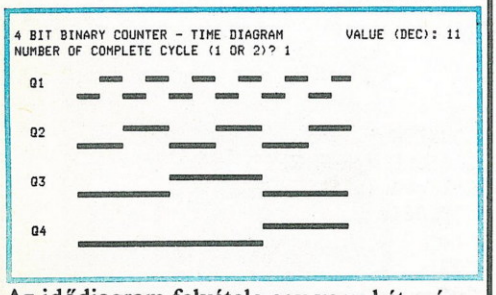
```
9995 INPUT "Az elso torlendo sor :";s: POKE 60001,s-256*INT (s/
256):POKE 60002, INT (s/256)
9996 INPUT "Az elso megmarado sor:";s: POKE 60010,s-256*INT (s/
256): POKE 60011,INT (s/256)
9997 RANDOMIZE USR 60000: STOP
9998 CLEAR 59999: FOR i=60000 TO 60023: READ a: POKE i, a: NEXT i
9999 DATA 33,0,0,205,110,25,229,235,213,33,0,0,205,110,25,
209,205,221,25,225,205,232,25,201
```

kezdődik (lásd a *2. lista* 1ØØ-as sorát!). A törlő jelet — ez L szint — a USER PORT A_1-es, azaz a bővítő Q_2-es kimenete adja a számlálónak. A folyamatos számlálást a program 11Ø-es sora látja el az órajelkiadó és kimenetlekérdező szubrutinok váltogatott hívásával. Az órajeleket a USER PORT A_0, azaz a bővítő Q_1 kimenetéről kapja a számláló, a négy kimenet állapotát a bővítő $D_9 \ldots D_{12}$, azaz a USER PORT $B_0 \ldots B_3$ bemenetein fogadja a számítógép.

2. ábra



3. ábra



Az idődiagram felvétele egy vagy két számlálási ciklusra lehetséges; ezt a program elején, az I1 változóba bevitt érték határozza meg.

A *2. ábrán* két teljes idődiagramot láthatunk. A különbség csupán a teljes számlálási ciklusok számában van. A *3. ábra* a mérés közben megszakított állapotnak megfelelő screent mutat be.

Második példánk — a számláló idődiagramjának felvétele — egyben arra is utal, hogy — természetesen megfelelő programmal — akár 12 sugaras oszcilloszkópot helyettesítő eszközhöz is juthatunk (sajnos ez a „műszer" egy kicsit lassú).

A két program látszólagos bonyolultsága senkit ne riasszon el a munkától. Ezek csupán azért ilyenek, hogy a képernyőn szemléltetendő eredmények áttekinthetők, könnyen kiértékelhetők legyenek. Emiatt a programokat a feltétlenül szükségesnél bővebb lére eresztettük.

NAGY IMRE

A TVC 64 k+

A TVC 32 k és TVC 64 k után a Videoton új TV—Computer-változat gyártását kezdte meg, melynek neve TVC 64 k+.

Az előző két TVC-hez képest a hardver által nyújtott lehetőségek két irányba bővültek:

— Használható rajta a VT—DOS operációs rendszer (a 64 k-s változatnál ehhez kisebb hardvermódosítás szükséges).

— A videomemória 64 kbájt méretű (szemben az előző változatok 16 kbájtos videomemóriájával), így a gép négy kép egyidejű tárolását teszi lehetővé (4×16 kbájt). A videomemória azon része, amelyet képelőállításra nem használnak, adatokat tárolhat, illetve gépi kódú programok is futtathatók benne.

A TVC 64 k+ változat új, bővített BASIC operációs rendszerrel rendelkezik, melynek verziószáma: BASIC 2.0. Az új BASIC alatt is futtathatók a korábbi 1.2 változat BASIC programjai. A BASIC programokban két ok miatt lehet szükség módosításokra:

— Az új utasításokkal azonos nevű változót használunk.

— A PLOT utasításba beépített kerekítés miatt a megrajzolt pontok, vonalak a képernyőn egy raszterponttal elcsúsznak.

Néhány új parancs, utasítás:

Parancsok:	AUTO (automatikus sorszámozás)
	RENUMBER (átsorszámozás)
Utasítások:	ON EXCEPTION GOTO (hibakezelő BASIC-szubrutin definiálása)
	EXCEPTION (szándékos hiba-előidézés)
	PLOT (relatív plot)
	PLOT RECTANGLE (téglalap és négyzet rajzolása)
	PLOT POLYGON (szabályos sokszög rajzolása)
	PLOT ELLIPSE (kör és ellipszis rajzolása)
Függvény:	HEX$(x) (hexadecimális átírás)

A TVC-ben bizonyos ROM-rutinokat nyílttá tettek, és restart vektorokon keresztül biztosították ezek meghívását gépi programok számára. Így hívható a hardverkezelést végző operációs rendszer és a BASIC aritmetikai rutincsomag. A 2.0 változatban ezek ugyanúgy működnek, sőt további rutinok is hívhatóvá válnak. A bővítések miatt azonban a ROM-címek megváltoztak. Ezért azok a programok, amelyek közvetlen ROM-hivatkozásokat tartalmaztak, futtatás előtt módosításra szorulnak. A rendszerváltozók számára fenntartott területen újabb változókat helyeztek el, ezért az itteni „üres" helyeket használó programokon is változtatni kell.

Várhatóan néhány, a régebbi típusokra íródott gyári és NOVOTRADE-program módosítások nélkül nem fog futni a 64 k+-on.

A VT—DOS operációs rendszer a Videoton TV—Computer új, fejlett operációs rendszere. A 64 k+-on futtatható, a 64 k-s gép hardvermódosítással, a 32 k-s pedig memóriabővítő használatával alkalmassá tehető a VT—DOS futtatására.

Maga az operációs rendszer egy floppycsatoló kártya, egy dugaszolható programmodul és egy rendszerlemez formájában jelenik meg. A floppycsatoló kártya tartalmazza a lemez fizikai kezelését és a fájlszervezést szolgáló programrészeket is, így a programmodul nélkül alkalmazva a gép a BASIC-ből használhatja a floppy perifériát.

A szabványos UPM funkciókat használó felhasználói programok futtathatók a VT—DOS-ban is. Jelentős előnye a VT—DOS-nak, hogy operátori parancsai szinte teljesen megegyeznek az IBM PC és a vele kompatibilis gépek parancsaival.

A VT—DOS is ugyanolyan lemezszervezést használ, mint az IBM PC-vel kompatibilis számítógépek. Ez azt jelenti, hogy e gépekkel a VT—DOS adatszinten kompatibilis. A lemez kompatibilitásához természetesen szükséges, hogy a TVC-n ne 720 kbájtra, hanem 360 kbájtra megformált lemezt használjunk, hiszen az előzőt a PC nem is ismeri. (Célszerű a lemezt 360 kbájtos meghajtóban formázni.)

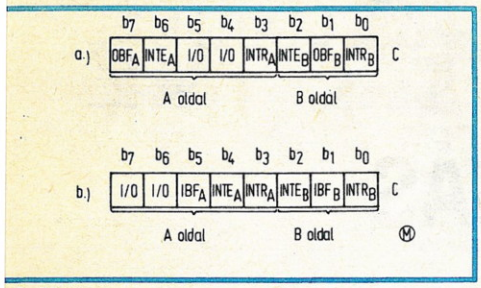
A VT—DOS jóval hatékonyabb környezetet biztosít a felhasználói programok számára, mint az UPM (de azért programkompatibilis vele).

— Egyszerű lehetőséget ad a véletlenül törölt fájlok visszaállítására.

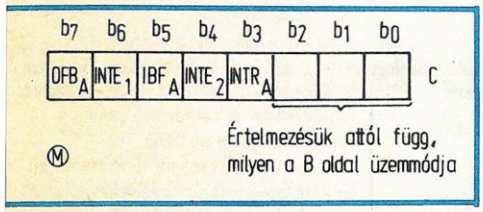
— Több mint 40 parancsot tartalmaz.

— Igen sok floppymeghajtó kezelésére alkalmas.

VÁMOS SÁNDOR



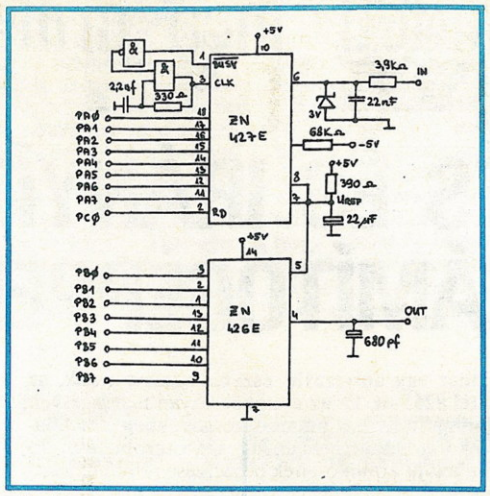
4. ábra



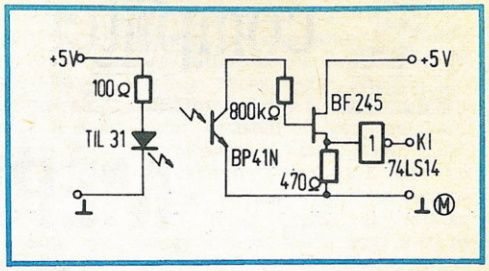
5. ábra

port, a $FD42 a C port, a $FD43 a CR parancsregisztere. A C Plus/4-nél a $FD40—$FDD0-ig terjedő terület nincs semmilyen tárterületre sem kikódolva. Ide tehát külső egységeket lehet elhelyezni. A 74LS123 a már említett R/$\overline{W}$ jelből állít elő megfelelő $\overline{R}$ és $\overline{W}$ jeleket a 8255 számára. A bemeneten zárójelben jelölt betűk és számok a C Plus/4 bővítő csatlakozójának jelzései.

A 8255 programozásával kapcsolatban: BASIC-ből POKE utasítással lehet az áramkörbe írni, PEEK utasítással olvasni, és természetesen gépi kódból a megfelelő regiszterek programozásával lehet az IC-t vezérelni. A programozásnál ügyeljünk, hogy először az üzemmódot írjuk be, majd a megfelelő port címzésével adatot írhatunk vagy olvashatunk.

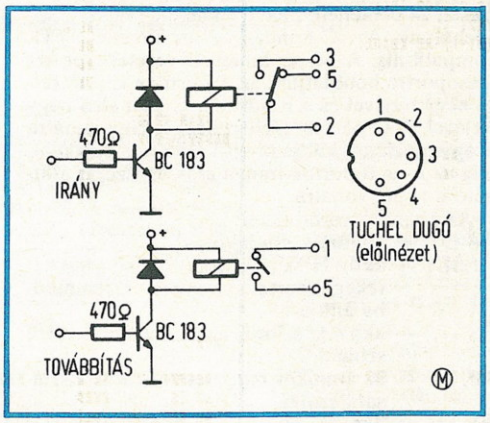


7. ábra



8. ábra

9. ábra



Néhány alkalmazási ötlet

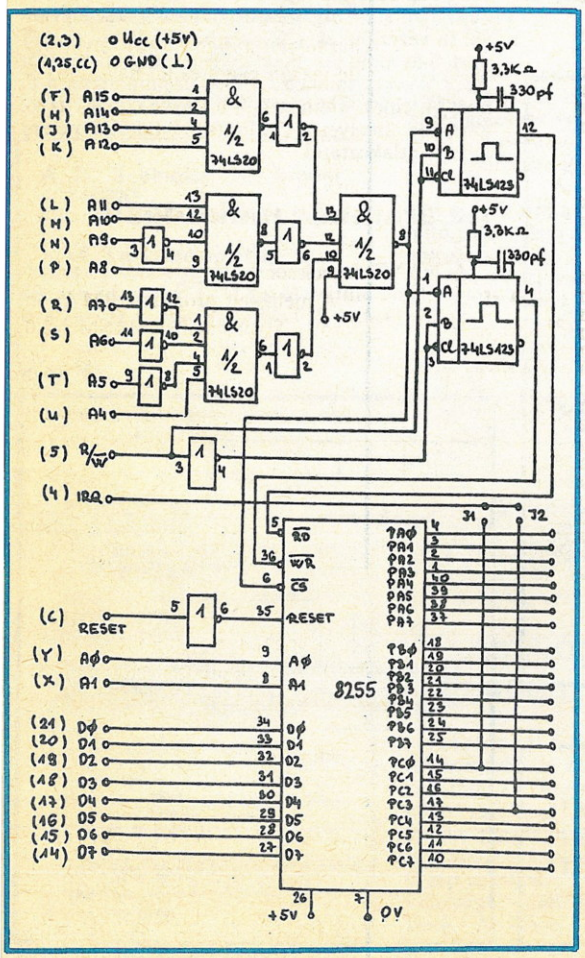
A *7. ábrán* egy analóg ki- és bemenet áramköri megvalósítását láthatjuk. Ehhez megfelelő osztó és erősítő áramköröket kapcsolva, mérőműszereket, motorvezérlőket és szabályozóegységeket készíthetünk. A *8. ábrán* egy egyszerű infrakapu látható. Körülbelül 20 cm távolságig működik. Több fénykapu felhasználásával pontos időmérésre is alkalmazható (az oktatásban például fizikai méréseknél jól használható).

A *9. ábra* egy egyszerű ASPECTOMAT diavetítő-vezérlőt ábrázol. Ezzel az áramkörrel programozhatóan vezérelhetünk diavetítőt (felhasználása például az oktatásban: grafikus, bonyolult ábrák helyettesítése diaképekkel). A 8255 felhasználásával kiterjeszthetjük gépünk tárkapacitását is. Készíthetünk különféle EPROM-okhoz égetőt (égető impulzusokat előállító áramkörrel lehet kiegészíteni a kapcsolást).

Az építésnél, felhasználásnál mindenképpen ügyeljünk arra, hogy csak kikapcsolt állapotban csatlakoztassunk bármit is a géphez. A gép tápegysége a közölt kapcsolásokat még kiszolgálhatja, de nagyobb bővítésekhez külön tápegységet kell készíteni.

NIEBERL JÓZSEF

6. ábra



NEM EMBER-, KÉPERNYŐMENTÉS!

Ezzel a programmal a TV-Computer teljes képernyőjét kimenthetjük, eltárolhatjuk, majd alkalmas helyen visszatölthetjük. Menteni az EXT 0-val, visszahívni az EXT 1 utasítással lehet.

A 4010-es sorban foglalunk helyet a tárolandó képnek. A 4020-as sorban állítjuk be a mentő, töltő gépi rutinok címeit (6639, 6665).

Ha nincs szükség a teljes képernyő mentésére, a DATA listán az alábbi címeket kell módosítani: 0,128 a videomemória kezdete, 35,26 a tárolás címe, 0,60 a mentett bájtok száma.

PÉNZES ÁKOS

```
4000 REM *** KÉPERNYŐ MENTŐ RUTIN ***
4010 LOMEM 22051: POKE 5920,35: POKE 5921,86
4020 POKE 33,239: POKE 34,25: POKE 35,9: POKE 36,26
4030 FOR I=6639 TO 6690: READ A: POKE I,A: NEXT: END
4040 DATA 62,80,50,3,0,211,2
4050 DATA 33,0,128,17,35,26,1,0,60,237,176
4060 DATA 62,112,50,3,0,211,2,201
4070 DATA 62,80,50,3,0,211,2
4080 DATA 33,35,26,17,0,128,1,0,60,237,176
4090 DATA 62,112,50,3,0,211,2,201
```

Azt hiszem, nem tévedek nagyot: a programírás alkalmasint egyik legfőbb erénye a tömörség és a rövidség. Természetesen hosszú program is lehet nagyon jó, ha a dolog is úgy kívánja. Ha viszont nyújtja valaki, mint a rétestésztát!... TVC-re írt grafikai trükkjeimet, hatásaimat éppen ezt elkerülendő készítettem. Igyekeztem programtyűimet minél rövidebbre fogni, hogy mindegyik szinte egy lélegzetre beírható legyen. A miniprogramoktól persze senki se várjon csodát, de arra alkalmasak, hogy bővítsék a gép egyelőre még vékonyka „szótárát".

Az első egy érdekes, színes — VIDEOTON — feliratot ír ki a képernyőre *(1. lista)*. Ez csak egy fajta, de próbálgatással sok szép betűtípust el lehet érni, így karakterek tervezgetése nélkül is látványossá tehetjük programunkat.

Ha közvetlenül a videoportra küldjük a sztringünket, (PRINT 0:"krixkrax"), akkor az az aktuális rajzolási pozíciótól kezdve, az aktuális színben, az aktuális keresztezési mód szerint fog megjelenni.

A 2896-os bájt első két bitje állítja be a karakter-felülírás módját. (1234) 2 esetén csak a tintaszínű pontokat írja a gép.

Ilyen módszerrel nagyon egyszerűen térbeli betűket is lehet rajzolni, úgy hogy a kiírt karaktersort még egyszer kiírjuk, függőlegesen és vízszintesen négy-négy egységgel eltolva, más színnel. (A 2896-ra ne felejtsünk 2-őt írni!) Ezt is nehezebb volt elmondani, mint megcsinálni...

A második is egy egyszerű BASIC program. Az általam futófénynek nevezett effektust mutatja be: a *2. lista* beírásával fogaskerekeket látunk forogni.

A fogaskerekeknek a „fogait" sorra a következő színnel rajzoljuk. (1231231...) A futófény, vagyis itt a forgás úgy jön létre, hogy a palettán az összes színt háttérszínűre tesszük. Mégpedig úgy, hogy kivesszük az egyest, a kettest majd a hármast. A három szín egyértelmű forgásirányt ad meg. (2-nél hol erre, hol arra látjuk forogni.) A mozgás három fázisát bizonyos késleltetéssel váltogatjuk, és már forog is a fogaskerék, a körhinta, fut a futófény, vagy rugdoshatja a szódás a lovát. Csupán a mozgás három fázisát kell külön-külön színnel megrajzolni.

Ezúttal végül az LDIR utasítás BASIC-ből való felhasználását mutatom be *(3. lista)*. Ezt a példa arra használja, hogy a képernyőt (csak 16200 bájt) pillanatok alatt elmentse, majd visszatöltse. Az LDIR blokkmozgatást végez. Tehát például áthelyezhetek vele a 6639-es címről a 6700-ra tíz bájtot. A rutin betöltése után az LDIR-t az EXT-tel érjük el BASIC-ből. EXT 0, forráscím, célcím, áthelyezendő bájtok száma.

```
ORG 6639: LOAD 6639
DI                      ;megszakítás tiltás
LD A,BO:OUT (2),A       ;video belapozás
LDIR                    ;blokkmozgatás
EI:RET                  ;megszakításengedélyezés, BASIC
END
```

BORDÁS BENCE

1. lista

```
10 REM **** 1. ****
20 GRAPHICS 4
30 POKE 2896,2
40 A$="VIDEOTON"
50 SET MODE 3
60 SET INK 0:PLOT 330,810:SET INK 3:PRINT #0:A$
70 SET INK 0:PLOT 338,818:SET INK 2:PRINT #0:A$
80 GET
```

2. lista

```
10 REM **** 2. ****
20 GRAPHICS 4
30 FOR I=0 TO 2*PI STEP PI/18
40 PLOT 512,480;512+200*SIN(I),480+200*COS(I)
45 PLOT 312,680;312+200*COS(I),680+200*SIN(I)
50 J=J+1:IF J>3 THEN J=1
60 SET INK J
70 NEXT I
80 SET PALETTE 0,85,0,0
85 FOR J=0 TO 30:NEXTJ
90 SET PALETTE 0,0,85,0
95 FOR J=0 TO 30:NEXTJ
100 SET PALETTE 0,0,0,85
105 FOR J=0 TO 30:NEXTJ
110 GOTO 80
```

3. lista

```
10 REM **** 3. ****
20 GRAPHICS 4
30 LOMEM 24700
40 POKE 33,239:POKE 34,25
50 READ A:IF A=555 THEN 100
60 POKE 6639+B,A:B=B+1:GOTO50
70 DATA 243,62,80,211,2,237,176,251,201,555
100 REM DEMO
110 LIST
120 PRINT:PRINT"GOMBNYOMASRA ELTESZI...":GET
125 EXT 0,32767,6700,16200
130 CLS:PRINT"GOMBNYOMASRA VISSZA!"
140 GET
150 EXT 0,6700,32767,16200
```

ÖRÖKÉLET

Úgy látszik, Tuba Imre kifogyhatatlan az örökélet-kódokból. Mostani gyűjteménye C16-os és C Plus/4-re bővített C16-os gépen működik.

3-D QUASARS	7198,173
AIRWOLF 2	7962,165
ARENA	4794,0
AURIGA	8113,189
BOMB JACK	5196,173
BOOTY	9013,165
BURGER HOUSE	8046,173
CAVE FIGHTER	4922,172
idő	4309,234
	4310,234
	4311,234
CRAZY GOLF	4174,201
	6890,0
Ezután E gombbal váltható a pálya.	
CREATURE CRAWL	10676,173
DIAGON	9606,0
DIAMOND MINE 1	5938,173
	7370,0
	8251,0
	9224,0
DIAMOND MINE 2	4206,173
GHOSTS 'N' GOBLINS	4365,173
	4393,165
GUNLAW	4627,173
HEKTIK	6468,173
	7363,173
JETBRIX	8864,173
time	16055,169
	16056,28
	16057,234
	8369,29
KIKSTART 1-3.	10403,165
idő	9033,189
	9036,189
	9082,189
	9085,189
	11938,12
	11946,32
KILLAPEDE	10496,173
KING OF KINGS	9725,173
	9747,173
KNOCK OUT	4630,173
L 'UMO	8960,173
LASER 900	7362,173
LASER ZONE	5426,100
elektro	5671,101
MAIL TRAIL	6613,173
days	6159,189

A HIÁNY HIÁNYT SZÜL

EGY GÉP

1988. június. Az ország különböző pontjain 12 ezer TVC van, ennek nagy része iskolákban. A gép olcsó és közkedvelt, a programok száma viszont kevés, és akad más kisebb-nagyobb hiba, kifogásolnivaló is. Mi hát mindezek oka, mi a valóság, s mi igaz a szóbeszédből? Elsősorban a tulajdonosok levelei nyomán indultunk útnak az illetékes cégekhez. Aztán az eredetileg egyszerűnek tűnő „nyomozás" érdekes fordulatokat vett.

Először Budapesten, a Videoton Számítástechnikai Kereskedelmi és Szolgáltató Leányvállalatánál jártunk, ahol Gergely János és Kiss Pál fogadott minket szívélyesen.

— Elterjedtek olyan pletykák, melyek szerint a TVC egy régebbi Enterprise gép licence. Tényleg így van?

— Ebben csupán annyi az igazság, hogy mindkettőt ugyanaz az angol cég tervezte.

— Pontosan mióta gyártják a TVC-t, s milyen a kereslet iránta?

— A sorozatgyártást 1985-ben kezdtük el, azóta majdnem 3000 darab 32 k és 9000 darab 64 k gép talált gazdára, illetve van kereskedelmi forgalomban. A vásárlók elsősorban az iskolák, hiszen iskolaszámítógépnek is szántuk, de bőven akadnak magánszemélyek is, akik főként a gép tudásához viszonyított olcsó ára miatt vesznek ilyen típusú számítógépet.

— Hogyan lehetett azt elérni, hogy a gép ára ne emelkedjen túlzottan?

— Igaz, hogy az árra rá kellett tenni az általános forgalmi adót, viszont máshol a gyártás folyamán csökkenteni tudtuk a kiadásokat, s ez eredményezte a csekély mértékű áremelést.

— Ez mindenesetre örvendetes, ám a géppel szemben problémák is felmerültek. A beépített botkormány labilis, hamar tönkremegy, ezenkívül a reset gomb sem tökéletes.

— Ezekről mi is tudunk, de sajnos, megváltoztatásukkal kapcsolatban még nem született döntés. Erről talán a székesfehérvári fejlesztők többet tudnak mondani.

— Ugyanez a helyzet a kitámasztó lábakkal is?

— Nem, az újabb gépek már stabil lábakkal készülnek.

— És az a billyentyű, amelyik backlash helyett aposztrófot ír?

(Erre mindketten nyomogatni kezdik az előttük levő gép billentyűit, próbálják kinyomtatni újra és újra, de a backlash csak nem akar a helyére kerülni.)

— Erre azt hiszem, nem tudok válaszolni, ez még számomra is újdonság.

— Hallottunk panaszokat a lemezegységgel kapcsolatban is. Várható-e új lemezegység megjelenése?

— Igen, már el is készült néhány egy- és kétmeghajtós egység. A régi UPM kompatibilis volt. Az új VT-DOS kompatibilis, de képes lesz elolvasni az elődje által készített lemezeket is. Ennek nagy előnye, hogy egy cartridge segítségével használhatóvá válik a CP/M operációs rendszer a 64K+ és a 32 k-s gép kibővített változatán. (Erről az újdonságról legutóbbi lapszámunkban közöltünk részletesebb információt.)

— És mit tudnak tenni azok a felhasználók, akiknek 64 k-s régi gépük van?

— Természetesen át lehet majd alakítani ezeket a gépeket is. Valószínűleg a szervizek is elfogadják a javaslatunkat, s vállalják majd az átalakítást, hiszen csupán egy IC beültetéséről van szó.

— Mely szervizek foglalkoznak a javítással?

— A TVC-vel több szervizlánc is foglalkozik. Az iskolai gépeket a Professional Országos Számítógép Szerviz, a magántulajdonúakat az Agroindustria Innovációs Vállalat javítja.

(Kaptunk egy listát, amelyen további nyolc cég neve és címe olvasható. Ezzel, úgy érezzük, a szervizelés gondja tényleg megoldottnak tekinthető. Minket azonban nemcsak ez érdekelt.)

— Egy gép megítélésében egyik legfontosabb szempont a szoftver. Milyen a programellátottság?

— Ez igen összetett kérdés. A tulajdonosok mindig több és több programot szeretnének, ami miatt a kínálat lemaradozik. Meg kell azonban mondani, hogy sajnos ennek más oka is van. Természetesen a Videoton is készített jó néhány programot, de született egy szerződés a Novotrade-dal, amely szerint ők a 2,6 millió forint értékű gépek, valamint a 400 ezer forint reklámköltség fejében vállalják hardverkiegészítők, évi száz program és négy könyv elkészítését. Most 1988 júniusa van, harminc programmal és két könyvvel.

— Mi volt erre az önök reakciója?

— Sajnos, nem tehettünk semmit, mivel a szerződés szankciókat nem tartalmaz.

— És ez miért van így?

— A Novotrade a magyar piacon monopolhelyzetben van. De találtak ürügyet is. Azzal védekeznek, hogy a gépeket és dokumentációkat két hónapos késéssel, 1987 februárjában szállítottuk le részükre. Azóta eltelt másfél év, s programokat, könyveket még ma is alig lehet látni. Hivatkozásuk alapja most a kazettahiány.

— De hát sorra látni a Novotrade-os kazettákat és a hozzájuk kapcsolódó hirdetéseket, úgyhogy kazettahiányról ilyen értelemben aligha beszélhetünk.

— Sajnos, nemcsak a kazetták számával, hanem a minőségükkel is baj van. A szalagok könnyen nyúlnak, hamar használhatatlanná válnak.

— Hol lehet ezeket egyáltalán beszerezni?

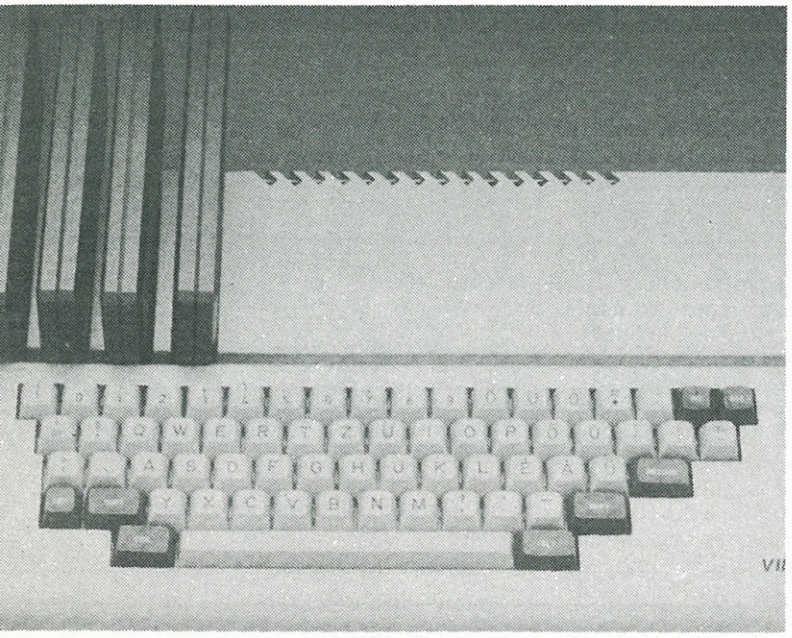
— Elvileg a Centrum áruházak foglalkoznak árusításukkal, ám ők előtérbe helyezik az Enterprise gépek és programok forgalmazását. A TVC csak ezután jön. Ez mindenképpen késlelteti a TVC 64 K+ megjelenését is, ami üzletpolitikai okok miatt majd csak akkor kerülhet a pultokra, ha az elődje elfogyott.

Haszontalan reklám

Utunk ezek után Székesfehérvárra vezetett, ahol kissé körülményesen, a bürokrácia által kiszabott utakat megjárva ugyan, de bejutottunk a gyárba. Itt Cseh Tibornak, aki a Számítástechnikai Gyár fejlesztési osztályán dolgozik, az előbbiekhez hasonló kérdéseket tettünk fel. Ő ugyan bizonyos dolgokról csak magánemberként mondott véleményt, de volt némi tanulság a válaszaiban.

— Olvasóink panaszolták a reset gomb és a beépített botkormány gyenge minőségét, s a billentyűzet egy apró hibáját. Lehetne ezeken valamit javítani?

SZÁZ BAJT CSINÁL?



— Természetesen ezeket házilag bárki kicserélheti, de mi a gyártáskor arra törekszünk, hogy a gép minél olcsóbb legyen, hiszen magasabb áron valószínűleg kisebb lenne a kereslet. A botkormány nyomókeresztje egyébként most már erősebb anyagból készül, de ha mégis eltörne, a garancia erre is vonatkozik. A reset gomb kicserélése többek között azért sincs betervezve, mert a gép dobozát előállító szerszámot kellene módosítani, ami pedig ugyancsak drága. A billentyűzet szépséghibája a festéshez szükséges maszk pontatlanságából ered, de a gépkönyv végén található egy helyesen kinyomtatott kiosztás, ami garantáltan pontos.

— Szerintünk a gépnek nagyobb reklámra volna szüksége. Mi erről a véleménye?

— A Videotonnak nem érdeke a gép reklámozása, ezen nekünk semmi hasznunk nincs; a TVC-t ugyanis körülbelül ugyanolyan áron állítjuk elő, mint amennyiért adjuk. Mi állami nagyvállalat vagyunk, fentről kapjuk az utasítást és néha a támogatást. Ezek után nem hinném, hogy a vállalat érdekei nagyobb propagandát kívánnának meg.

— És mi a helyzet a TVC-klubokkal?

— A Videoton a klubjainak csak gépeket ad. Én személy szerint csupán két TVC-s klubról tudok, az egyik itt Székesfehérváron, a Videoton Oktatási Központban, a másik Budapesten, a Marcibányi téri művelődési házban található. (A budapesti klubot időközben kitették innen, jelenleg a Bem rakparton találtak otthont. — A szerzők.)

Az ördög sohasem alszik?!

Természetesen kíváncsiak voltunk a Novotrade hozzáállására is. Csak kisebb kutatással tudtuk meg a TVC-ügyben jártas emberek nevét, de sajnos, a szerződés ügyében illetékes nem került elő. Szerencsére azonban más jellegű információkat bőven kaptunk az Octasoft-ban. Első kérdésünk a kazettahiánnyal volt kapcsolatos.

— Sajnos, az utóbbi időben tényleg nagyon nagy kazettagondokkal küzdöttünk, de remélem, hogy ez már a múlté. A szocialista importban semmiféle szankcióra nincs lehetőségünk, aminek következménye, hogy a megrendelt kazetták csak néha-néha érkeznek meg. Ráadásul egy másik Enterprise-os megrendelésnél már kötbért is fizettünk, így annak a sorozatnak az előállítása fontosabb volt, ennélfogva az élvezett előnyt. Az áruházakban pedig azért szembetűnőbb az Enterprise programok magasabb száma, mert azokat a Centrum már korábban felvásárolta. A TVC programokra viszont jóval kisebb a kereslet.

— De a minőséggel is vannak problémák.

— A kazetta ma már nem a legmodernebb adathordozó, ennek ellenére még szükség van rá, s ha használjuk, legalább a magnetofon lejátszófejének pontos beállítására ügyeljünk. Ugyanazt a programot az egyik magnón be lehet tölteni, a másikon nem. Egyébként már mi is szerettünk volna venni egy modernebb másológépet, de ez a lehetőség anyagi okok miatt kútba esett, mert az oktatóprogram nem a legkifizetődőbb.

— Egyáltalán hány program kapható a Novotrade forgalmazásában?

— Jelenleg körülbelül negyven. De mutatok egy megrendelőt, ez ha jól tudom, teljes.

— Ez elképesztő, ebből a 39 féle programból összesen csak 77 darabra van megrendelés?

— Ettől a cégtől igen, de a többi sem különb. Szeretnénk nagyobb tételben is eladni iskoláknak, hogy az árakat lejjebb szoríthassuk. Állami támogatás inkább csak a gépekre van, s a programok vásárlására kevés pénz jut.

— Pedig ha nincs program, a gépet sem veszik annyian.

— Ez már csak egy ilyen ördögi kör. Mi olcsón behoztuk volna továbbra is a C Plus/4-et, saját valutakeretből, de a belföldi ipar tért hódított. Mire kezdtünk belejönni a Plus/4-be, stílust kellett váltanunk. De ha csak ezen múlott volna! Jártam én már olyan számítástechnika-oktatási tanácskozásokon is, ahol... De ezt hagyjuk.

Gép — kazetta — könyv. Három egység, amely nélkülözhetetlen. Vajon mi lehet a legsanyarúbb sorsúval, a harmadikkal, a könyvvel? Erről a Novotrade-nél telefonon érdeklődtünk. Kérdéseinkre Szemere Gabriella adott választ.

— Szeretnénk tudni, hogy az eddig megjelent kiadványokon kívül (TVC BASIC, Operációs rendszer) számíthatunk-e újabbakra?

— Igen, most jelenik meg egy hardverről szóló kiadvány, és további két könyv is szerepel terveinkben, a gépi kódú programozásról és a TVC ROM-ról szólóak, melyeket az év végéig szeretnénk megjelentetni.

Mindenkinek kára...

Az eddigiek mindenképpen azt sugallják, hogy a Novotrade és a Videoton kapcsolatai nem nevezhetők rózsásnak. Erre konkrét példa, hogy a gépre eddig a hivatalosan forgalomba került programokról sehol sem tudtak összefoglaló kiadványt adni. A két cég szerződésében szó van egyébként hardverkiegészítőkről is, holott a Novotrade-esek szerint cégük nem is foglalkozik ilyesmivel, továbbá a szerződés nem köti ki ezek számát sem, tehát akár semmibe is vehető.

Szomorú, de végül a boltokban tapasztaltak szerint sem a legjobb a programkínálat. Sehol sem haladta meg a hatnyolc darabot.

Ez hát a keserű igazság, s mindennek kárát a felhasználó vagy az a diák látja, akinek az iskolában ilyen gépparkon kell tanulnia. De mi van akkor, ha nem csak kell, hanem meg is szeretné tanulni szegény diák a számítástechnika fortélyait, vagy a számítógép segítségével szeretné elsajátítani más tantárgyak ismeretanyagait? Hiszen azért van a gép, hogy vele tanuljunk, vagy legalábbis azért lenne. Vagy netán valami porszem került az iskolaszámítógép-program gondosan megolajozott fogaskerekei közé?

BÁRTFAI BARNABÁS—
VÁMOS SÁNDOR

ÚJ ASSEMBLER A TVC-RE

Ha új, akkor bizonyára régebbi is volt. Most, hogy éppen nem egy oktatóprogramot elemzek — aminek az az oka, hogy a szerkesztőségben a kezembe nyomtak egy programot, nézzem meg —, töprengek, hogyan is fogjak hozzá. Kérdezősködés után kiderült, hogy VIDAS néven már a Videoton is kínálja assemblerét a gépéhez. Kézenfekvő tehát, hogy összehasonlítsuk az újat azzal, ami van.

A program készítője kisvállalkozó, aki termékét a sok mikrogépes vállalkozást igen dinamikusan támogató Magiszter Könyvesbolt útján terjeszti.

A program leírása bizony elég vékonyka, talán azért is, hogy elférjen a kazettatartóban. A VIDAS leírása sem több. Megtaláltam viszont egy olyan Z80 assembler leírását, amit korábban Spectrumon használtam. Ez a SPECTRE—MAC—MON, mely fullscreen editorral integrált mikroassembler. A leírása majdnem harminc oldal. A szövegek körülbelül arányosak a mondanivalóval. A VIDAS leírása csak azért olyan hosszú, mert tartalmaz egy mnemonik táblázatot, egyébként az összehasonlítottak közül a legkevesebb funkció található benne. A programot lényegében a végleges címre helyezi, ami a lehetőségeit korlátozza. Az „új" termék ennél többet tud. A LOAD funkcióval utasíthatjuk, hogy a végleges cím hol legyen.

A VIDAS ténylegesen csak assembler, igaz, annak elég gyors. Az „új" termék viszont, a SPECTRE—MAC—MON-hoz hasonlóan, komplett szoftverfejlesztő rendszer. Ezzel a tulajdonságával hasonlít a Borland cég turbo szoftvereinek filozófiájához, ezért elég korszerű programnak ítélhető. Ha a tárigényeket hasonlítjuk össze, az eltérés szintén szembeötlő. A VIDAS 32 k-s gépen is működik, míg az „új" termék csak 64 k-s gépen tud futni. A szolgáltatásaihoz tehát igen sok tárat elrabol. Ezt nyugodtan felróhatjuk neki, hiszen a több funkciót ismerő SPECTRE—MAC—MON mindössze 16 k-t igénnyel, pedig nem a Spectrum BASIC editorára hivatkozik, mint akár a VIDAS, akár a TV-Computer Assembler. A nyelvi elemek tekintetében a VIDAS-ban a gépi kód mnemonikjain kívül csak elemi utasításokat használhatunk (ORG, EQU, DB, END). Az „új" termék a már említett LOAD áthelyező utasításon kívül csak néhány újabb adatmezőtípust definiáló utasítást ismer. Ezzel szemben a SPECTRE—MAC—MON ismeri a feltételes fordítás lehetőségét — beleértve a ciklusokat is —, és a szokásos assembler direktívákon túl, mint neve is jelzi, makroprocesszor-lehetőségeket is tartalmaz.

A SPECTRE—MAC—MON-nal akár újradefiniálhatjuk a kódgenerátor-fokozatot is. Amikor használtam, éppen alkalmam nyílt ennek a lehetőségnek a kipróbálására. A programbelövéseknél azt vettem észre, hogy a programomban az egyik ADC utasítás helyett ADD van. Nem így a forráskódban. Nos, meglepődtem egy kicsit, mert hosszú pályafutásom alatt assemblert még nem csíptem rajta hibás működésen. A leírás és a programhoz mellékelt forrásnyelvű makrokönyvtár elemzésével sikerült kiderítenem, hogy az ADC neve nem volt felsorolva az egyik tulajdonságlistán. Ennek kijavításával a programot újrageneráltam, és lássunk csodát, a programhibám megszűnt. A „szerszámom" alkalmas volt arra, hogy még önmagából is kiköszörülhető legyen a csorba.

Mellesleg a forrás- és lefordított program mentő és ellenőrző-betöltő kódrészei egy BASIC modulban voltak. Ez lehetővé tette, hogy a SPECTRE—MAC—MON-t meghajtó helyett át lehessen helyezni 3,5″-os hajlékonylemezre. Ebben a konfigurációban egy Spectrum 128 k-s gépen a meghajtóval elképzelhetetlen fejlesztési sebességet lehetett elérni.

Az „új" termék végül is tartalmaz assembler, disassembler, programlistázási, nyomtatási, monitor (adatbeírási, tárlistázási, tármásolási, tármozgatási), programbelövési (regiszterállapot, töréspontkezelés, lépésenkénti végrehajtás, elindítás) és egyéb funkciókat (munkamező határai, számrendszeralapszám-váltás, kimentés-betöltés a forrás- és tárgykódra).

A használat közben gyorsan visszatérhetünk a BASIC editorba javítás és programbevitel céljából. A program igen jól használható fejlesztő eszköz, még középiskolai iskolagép-környezetben is többet nyújt annál, ami igazán kihasználható. Kevésbé használható profi fejlesztéseknél. Ilyen célra a SPECTRE—MAC—MON-féle megoldás lehet csak igazán jó, aminél szintaktikus vagy címhivatkozási hibánál az assembly leállásakor a kurzor a hibás ponton villog. Ezenkívül legalább a feltételes fordítás az, ami egy profi fejlesztésnél nélkülözhetetlen. Hogy ne is elemezzük a keresztfejlesztést lehetővé tevő makroprocesszor szükségességét. A SPECTRE—MAC—MON talán működne a TVC testvérgépén, az Enterprise-on, Spectrum emulátorral. Ki kellene próbálni, ha még nem tette meg valaki.

Az „új" termék tehát elég korszerű elveken alapuló, jó program, amely iskolagép-környezetben feltétlenül, profi célokra kevésbé, de még így is javasolható.

ZSADÁNYI PÁL

ÖSSZEFOGLALÓ ADATOK

Forgalmazó:	Magiszter Könyvesbolt Bp. V., Városház u. 1.
Terméknév:	TV-Computer Assembler
Szerző:	Kós Géza
Géptípus:	TVC
Hordozó:	kazetta
Dokumentáció:	12 oldalas, a kazettatartóban
Ár:	472 Ft (ÁFA-val)

MINŐSÍTŐ ADATOK

Kezelhetőség:	jó
Teljesség:	jó
Dokumentáltság:	jó
Ár/teljesítmény:	kiváló
Összbenyomás:	jó

gyázni kell arra, hogy a 256. sorozat megtalálása után fejezzük be a tömörítést. (Ezt a lépést egy adott sorozat harmadik, negyedik stb. megtalálása után már nem kell elvégezni.)

c) A sorozatok kicserélése két bájtra: 1 — jelzőbájt, 2 — a sorozat sorszáma. Természetesen az egyik — mondjuk a 0. — „sorozatnak" magának a jelzőbájtnak kell lennie. A tömörítés megkezdése előtt a jelzőbájt minden előfordulása után be kell szúrni egy $00 bájtot (0. sorozat). Újra jelentkezik tehát az a probléma, amiről a bájtos tömörítésnél már szó volt.

d) Vissza az a) ponthoz.

Ezeket a lépéseket többször egymás után, többféle sorozathosszal ajánlott végrehajtani.

Mint látható, a keresés során iszonyúan sok összehasonlítást kell végezni, emiatt a tömörítés akár órákig is eltarthat. Egy lényegesen gyorsabb keresőalgoritmus alapjainak leírása található a Tudomány (Scientific American) 1986-os szoftver-különszámának 31—32. oldalán, Niklaus Wirth Adatszerkezetek és algoritmusok című cikkében.

A szétpakolónak a jelzőbájtot és a táblázat helyét kell ismernie. Ezek alapján nagyon könnyű dolga van: olvassa a tömörített fájlt, ha jelzőbájtot talál, a megfelelő sorozatot belefejti a szövegbe. Vigyázat! A továbbolvasást az éppen befejezett sorozat elején kell kezdeni, kivéve, ha a 0. sorozatot (jelzőbájt) fejtettük ki.

Deierl András—Molnár Imre

ÖRÖKÉLET

ATLANTIS	13660,209
	17128,145
BRIDGEHEAD	8282,165
	8444,165
bomba	11446,165
CUTHBERT IN THE TOMBS	7499,173
idő	7148,173
DEATH RACE	8860,169
extended play	8707,48
ENIGMA	8933,234
energy	13306,165
FINDERS KEEPERS	28497,173
ICICLE WORKS II	21765,0
	24423,0
time	23354,0
JACK ATTACK	8635,165
	8640,234
	10460,234
	10473,165
LEAPER	8076,234
	11562,234
MONTY ON THE RUN	11572,173
PURPLE TURTLES	7077,173
time	8087,172
ROBO KNIGHT	15966,234
time	12171,0
TREASURE ISLAND	6029,173

Miért is nincsen körülöttünk örök nyár! — ábrándoznak a napimádók. S miért nem lehetünk örökké gyermekek és ifjak, miért nem élhetünk örökké! — sóhajtoznak, akik álomképükből száműzik és megtagadják a létező világot. Nem is gondolnak rá: milyen unalmas is volna az az örökkévalóság! — A játékok kedvelőinek persze nem illúzió az örökélet. Sőt örülnek, ha minél több van belőle. Szerencsére Tuba Imre kódolt megint. C Plus/4-re és 64 kbájtra bővített C16-ra ismét készített egy csokorra való örökéletkódot.

Az ilyen terülj, terülj asztalkámra pedig igazán senki sem mondhatja, hogy unalmas!

Képcsere — bitenként

Játék- és rajzprogramokban gyakran szükség lehet két kép gyors és látványos cseréjére. Az alábbi rutinnal ez megvalósítható. A program memóriatakarékos, mivel nem használ külön munkaterületet.

BASIC-ből az EXT 0,x,y,z utasítással hívható, ahol

x — a cserélendő kép(részlet) kezdőcíme,

y — a háttér kép(részlet) kezdőcíme,

z — a cserélendő kép(részlet)-ek hossza.

Például az EXT 0,32767, 16384, 15360 utasítás hatására az egész képernyőt felcseréli a 16384-es címen eltárolt képpel.

Futtatás előtt nem árt kimenteni a programot.

Kókai László

```
  1 !*************************************
  2 !*    Bitenkenti kepcsere TVC-re     *
  3 !*     (C)1988 BUDDHA Softhouse      *
  4 !*            Kokai Laszlo           *
  5 !*************************************
 10 KE=3735:I=0
 20 READ A:IF A=-1 THEN GOTO 50
 30 CH=CH+A:POKEKE+I,A:I=I+1
 40 GOTO 20
 50 IF CH=5562 THEN 60
 55 PRINT"Hiba a DATA sorokban!!":END
 60 PRINT"Kesz!!!":SOUND
 70 POKE 33,3735 AND255:POKE34,3735/256
 80 PRINT"EXT 0,eloter,hatter,hossz"
1000 DATA243,62,80,211,2,34,200,14
1010 DATA237,83,202,14,237,67,204,14
1020 DATA6,8,197,42,200,14,237,91
1030 DATA202,14,237,75,204,14,26,23
1040 DATA203,22,26,23,18,35,19,11
1050 DATA120,177,32,242,193,16,227,251
1060 DATA201,0,128,0,64,0,60,-1
```


A HTE Mikroszámítógépes Programnyelvek és Operációs Rendszerek, és a MATE Irányításelméleti szakosztályai közös rendezésében

A SIMULA 67 szimulációs bázisnyelv alkalmazási lehetőségei IBM PC-n

címmel előadást tartunk. Előadó: Gáspár András, a magyar S67 fordító szerzője. Azoknak, akik két lemezt is hoznak magukkal, a szerző díjmentesen ráviszi demo-anyagát. SIMULA nyelvdefiníciós könyvek is kaphatók a helyszínen 40 forintos áron.

Minden érdeklődőt szeretettel várunk!

Időpont: 1989. február 9., csütörtök 14 óra

Hely: Budapest V., Kossuth tér 6—8. MTESZ-székház, III. em. 333.

„KORMÁNYVÁLTÁS"

A C Plus/4-es gépek billentyűzetének és botkormányának adatvonalai közös TED regiszterhez csatlakoznak. Ez az oka annak, hogy a programok, amelyeket csak botkormánnyal lehet irányítani, könnyen módosíthatók úgy, hogy billentyűzetről kezelhetők legyenek. Ehhez csak annyit kell tenni, hogy a megfelelő TED regiszter figyelését a $DB70 címen kezdődő billentyűzetvizsgáló rutinra kell bízni. Az alábbiakban ismertetem néhány program módosítását:

A program neve	Cím	Módosítás
BOOTY	$1175	00 20 70 DB
HARVEY HEAD	$1129	00
	$115C	20 70 DB
BONGO	$2C9D	00 20 70 DB
FURY	$2712	00 20 70 DB
RUNNER	$29E1	00 20 70 DB
INVADERS	$1067	00 20 70 DB
BMX RACER	$2030	00 20 70 DB
AUTOBAHN	$383A	00
	$3866	20 70 DB
STELLAR WARS	$108A	00 20 70 DB
TREASURE ISLAND	$7447	00 20 70 DB
	$1610	00 20 70 DB
STARFORCE NOVA	$2387	00 20 70 DB
	$1534	00 20 70 DB
ROBO KNIGHT	$281A	00 20 70 DB
MATRIX	$13E0	A9 00 20 70 DB
AIRWOLF 16	$2F85	00 20 70 DB
AUTO ZONE	$1C3C	00 20 70 DB

A módosítás tehát mindössze annyiból áll, hogy a megfelelő program betöltése után belépünk monitorba és kiadjuk a > parancsot a fenti táblázatban szereplő operandusokkal. Például a BOOTY programnál a parancs:

> 1175 00 20 70 DB.

Ezután irányítására a billentyűzetet használhatjuk. A botkormány egy-egy iránya több billentyűvel is helyettesíthető:

Fel: <1> <9> <7> <5> <3>

Le: <CLR/HOME> <•> <P> <I> <Y> <R> <W> <RETURN>

Jobbra: <2> <0> <8> <6> <4> <HELP>

Balra: <CTRL> <;> <L> <J> <G> <D> <A> <£>

Tűzgomb

PORT1 esetén: <SHIFT> <@> <X> <X> <V> <N> <,> </> <RUN/STOP>

PORT2 esetén: <E> <F3> <T> <U> <0> <—> <+> <Q>

Egy kicsit bonyolultabb a megoldás akkor, ha egy program eredetileg botkormányt és billentyűzetet is használ. Ilyen például a MERCENARY, aminek az átírásához az alábbi gépi kódú program szükséges:

Cím	Módosítás
$07F8	80
$B27F	A9 BF 20 40 FF 20 4A FF
$B287	A9 DF 20 67 FF
$FF40	8D 30 FD 8D 08 FF AD 08
$FF48	FF 60 A2 FF 8E 90 FF 4A
$FF50	B0 07 48 A9 04 20 60 FF
$FF58	68 4A 4A 4A B0 08 A9 08
$FF60	4D 90 FF 8D 90 FF 60 20
$FF68	40 FF 4A B0 07 48 A9 02
$FF70	20 60 FF 68 4A 4A 4A B0
$FF78	05 A9 01 20 60 FF A9 FD
$FF80	20 40 FF 0A B0 05 A9 40
$FF88	20 60 FF AD 90 FF 60 FF

Ilyenkor a botkormány helyett a kurzormozgató és a SHIFT billentyűk használhatók.

Devecz Ferenc

Görgetés TVC-re

A Mikroszámítógép Magazinban már jelent meg TVC-re scroll rutin, az 1986. decemberi számban. Ezt azonban túlságosan lassúnak találtam.

Ez a két rutin, amely jobbra, illetve balra görgetést valósít meg, már elég gyors ahhoz, hogy játékprogramokban is felhasználható legyen. A rutinok előnye az is, hogy mindhárom grafikus üzemmódban alkalmazhatók. A listán látható program a két rutinnal a memória bármely szabad területén működőképes. BASIC-ből az EXT 0, illetve az EXT 1 parancsokkal hívható.

Kókai László

```
  1 !*********************************
  2 !* Gyors scroll jobbra/balra TVC-re *
  3 !*      (C)1988 BUDDHA Softhouse    *
  4 !*            Kókai László          *
  5 !*********************************
 10 KE=3735:I=0
 20 READ A:IF A=-1 THEN GOTO 50
 30 CH=CH+A:POKEKE+I,A:I=I+1
 40 GOTO 20
 50 IF CH=6392 THEN 60
 55 PRINT"Hiba a DATA sorokban!!":END
 60 PRINT"Kesz!!!":SOUND
 70 POKE 33,3735 AND255:POKE34,3735/256
 80 POKE 35,3759 AND255:POKE36,3759/256
 90 PRINT"Jobbra:ext0":PRINT"Balra :ext1"
1000 DATA243,62,80,211,2,33,0,128
1010 DATA6,240,197,229,209,35,1,63
1020 DATA0,237,176,193,16,244,251,201
1030 DATA243,62,80,211,2,33,63,128
1040 DATA6,240,197,1,63,0,229,209
1050 DATA43,237,184,35,17,127,0,25
1060 DATA193,16,239,251,201,-1
```

Listánkat felhasználói, illetve játékprogramokból állítjuk össze. A legjobbakat, legérdekesebbeket a beküldött javaslatok alapján rangsoroljuk. Ehhez kérjük az olvasók közreműködését. C64-re, ZX-Spectrumra, Enterprise-ra, TVC-re, Atarira és IBM-re készült programsorokat várunk havonta.

Címünk:
Mikroszámítógép Magazin
Szerkesztősége
1371 Budapest, Pf. 433
A diákszerkesztőség

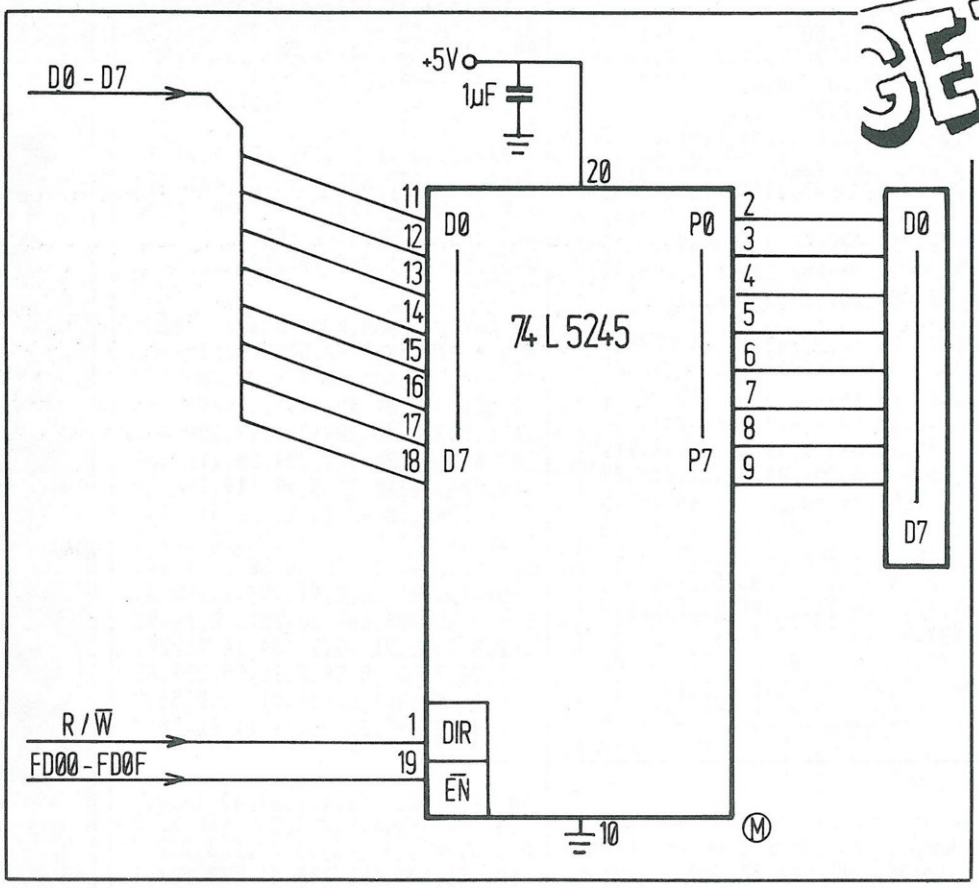
TOP - lista

BBS

	Felhasználói progamok	IBM	AMIGA	C-128	C-64	C+4 (16)	SPECTR.	ENTERP.	TVC
1	Giga Paint				•				
2	Geos 1.6 d.				•				
3	Rockmon v5.5				•				
4	Giga Cad +				•				
5	Windows	•			•		•	•	
6	Disk manager				•				
7	Printfox				•				
8	Printmaster	•			•				
9	Game maker				•		•		
10	Art studio 1.5e				•		•		

	Játék programok	IBM	AMIGA	C-128	C-64	C+4 (16)	SPECTR.	ENTERP.	TVC
1	Maniac M. II.				•				
2	Lord of Midn.						•		
3	Summer Ol.				•				
4	LLL	•							
5	Bombák								•
6	Super cobra					•			
7	Barbarian III.				•				
8	Monopoly II.	•							
9	Defender oto.		•		•				
10	Elite						•		

User port



1. ábra

Bizonyára sok C16-tulajdonos elgondolkozott már azon, hogy vajon miért nincs a gépén felhasználói csatlakozó. E kérdést én is megrágtam, de választ nem találva, magam kezdtem hozzá egy user port építéséhez. Az áramkört úgy terveztem, hogy a gép memóriabővítőjéről működjön.

Az *1. ábrán* látható IC kétirányú sínmeghajtó erősítő áramkör. Az egész IC-t a 19-es lábra adott engedélyező jel vezérli. Az írás-olvasás irányát az 1-es lábra adott jellel választhatjuk meg.

A user port kiválasztásához — a Plus/4-től eltérően — én azt a megoldást választottam, hogy a 7700—10 IC 15., FD00—FD0F címzésű, bekötetlen lábával engedélyezem a portot.

Ezután jön az építés neheze, mivel a 15-ös láb sehová sincs kivezetve. Ezt saját magunknak kell megoldanunk. A memóriabővítő *(2. ábra)* Z, AA, BB lábai bekötetlenek, ezért ide lehet vezetékkel a 15-ös lábat kivezetni. Ehhez a megoldáshoz egy csavarhúzó, forrasztópáka (nem pillanatforrasztó!) és mintegy 15-20 cm huzal szükséges.

A fedelet és a panelt borító hűtőlapot a csavarok kicsavarásával eltávolítjuk, majd a panel alsó felét borító fémlapot „operáljuk" le. Már csak a 7700—10 IC 15-ös lábát és a memóriabővítő bekötetlen kivezetését kell megtalálnunk, és máris forraszthatunk.

A user port panelt a memóriabővítő kivezetéseinek figyelembevételével alakítsuk ki.

Horváth Gábor

2. ábra

25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
⏚	Ø2	Ext. aud.	AEC	D0	D1	D2	D3	D4	D5	D6	D7	BA	MUX	CAS	CS0	CS1	C2H	C2L	C1H	R/W	IRQ	+5V		⏚
⏚				A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	Ø0	RAS	RES	C1L	⏚
Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
CC	BB	AA	Z	Y	X	W	V	U	T	S	R	P	N	M	L	K	J	H	F	E	D	C	B	A

Hivatalos jelölés

M

Hess, te csúnya!

Sokan szeretnék megvédeni programjukat, listázását megakadályozni. Ehhez nyújt hathatós segítséget a következő program, amely a CTLR+ESC használatát hatástalanítja:

TVC

```
1!.............................
..
  - 30 db pont -
  FOR A=6643 TO 6670: INPUT B:
  POKE A, B: NEXT A
```

A beírandó értékek:

```
243,33,70,11,54,195,35,54,
1,35,
54,26,251,201,197,213,229,
33,22,
11,54,0,225,209,193,241,251,
201
```

Indítás: X=USR 6643

A program felhasználja azt, hogy az operációs rendszer, az editor kezelése folyamán a 2851 dec címre ugrik. Ezen a címen egy, az editor számára nélkülözhetetlen gépi nyelvű szubrutin található. Azt, hogy az operációs rendszer megszakításkor melyik címre ugorjon, a 49 dec címen kezdődő két bájt határozza meg. Az ezt kihasználó, gépi nyelvű szubrutinoknak egy JP 2851 utasítással kell befejeződniük.

Szalontai Béla

jük a sebességet, mert földhöz csapódhatunk. Igyekezzünk a magasságot a rotorok állásszögével szabályozni!

A CÉLNÁL

Repülésünk közben többször megjelenik a (Target) célpont felirat. Az ehhez tartozó objektum távolságát és jellegét a (Space) megnyomásával olvashatjuk le az információs képernyőről. A kékkel írt, illetve U. S.-szel jelölt célokra ne lőjünk, hiszen azok saját állásaink. Az átkapcsolás után megjelenő célkereszt színe mutatja a találat valószínűségét, úgy, hogy minél világosabb, annál közelebb vagyunk a célhoz, tehát biztosabb a találat. A géppuska használatánál nem kell a célt a kereszt közepébe igazítani, mert a lövés automatikusan oda irányul. Rakéta használatánál azonban ez a mi dolgunk.

Repülésünket a térképen is nyomon követhetjük, ahol alapállapotban az első célponton helyezkedik el a célkereszt, amit azután a célra kell igazítanunk.

A célhoz közeledve elsősorban a megjelölt célpontra lőjünk. Sikeres találatnál rádióüzenetet kapunk. Addig keressük ezt a célt, amíg az üzenet meg nem jelenik, ugyanis csak akkor teljesítettük a feladatot!

A FEGYVERZET

A műszerfalon kijelezve látjuk fegyvereink jellegét és számát. Ezek balról jobbra haladva a következők: hőkövető rakéta, levegő-föld rakéta, levegő-levegő rakéta, radarzavaró fémcsíkok, gépfegyverlőszer, a kiválasztott bomba.

A VISSZATÉRÉS

A feladatok végrehajtása után vissza kell térnünk egy bázisra (U. S. Base). Ezekből három van. Repüljünk a leszállópálya fölé. Ereszkedjünk lassan, nulla sebességgel le, amíg rádióüzenetet nem kapunk. Ebben kérdezik azt a bizonyos jelszót.

Igyekezzünk nagyon alacsonyan repülni, mert a jelszó nélkül biztosan kilőnek, és lezuhanunk. A néhány méter magasból való zuhanásnál nem történik nagyobb baj.

Sikeres leszállás után tájékoztatást kapunk a sikeres feladatok számáról, magunk és gépünk állapotáról.

A BILLENTYŰZET ÉS A MŰSZERFAL

1,2	motorok ki/be kapcsolása
3	motorok rotorra kapcsolása
4,7	aktuális fegyver kiválasztása
\,CLR,DEL	jobbra, előre, balra tekintés
O	infrakereső ki/be
+	radar ki/be
F1, F3	a két rotor állásszögének növelése
F5,F7	a két rotor állásszögének csökkentése
SPACE	információs képernyő hívása
Z	térkép hívása
CRSR	forgás nulla sebesség esetén

A pilótafülke tetején az ablak fölött megfigyelhetünk 14 betűt. Ezek a gép bizonyos elemeinek állapotát jelzik. Meghibásodáskor sárgán, működésképtelen állapotban pedig pirosan világítanak. A jelzők sorrendben — a műszerfalon balról jobbra — a következők:

R fő rotor
O orroptikák
A orr, repüléselektronikai berendezések
G 30 mm-es gépágyú
F orr-üzemanyagtartály
W jobb oldali szárny
W bal oldali szárny
E jobb oldali motor
E bal oldali motor
F hátsó üzemanyagtartály
A hátsó repüléselektronikai berendezések
L radar álcélpontkilövő
L infra álcélpontkilövő
R forgatónyomaték-kiegyenlítő rotor

Tass Csaba

Mini mutatványok

Harminc sor, három program. Három apró grafikai mutatvány, amelyekhez hasonlókat a Szinte egy lélegzetre... című cikkben a tavaly novemberi számban már bemutattam. Tehát a mostaniak sem virtuóz hosszúak, inkább szórakoztató minik. Persze a mini is lehet tanulságos.

Az *1. listán* egy ötsoros BASIC programocska látható. Aki még nem látott TVC-t garázdálkodni, az el fog csodálkozni, hogy milyen gyors grafikát tittyent az öreg BASIC haló poraiban.

A SET MODE 3 hatását használja fel a program. Ha két vonalat egymásra írunk 3-as MODE-ban, akkor a felső letörli az alsót, és a korábbi hátteret látjuk alatta. Így lehet például célkeresztet mozgatni a táj fölött.

A következő program *(2. lista)* szerencsére nem jó semmire. Futtassuk le: egy mozgalmas grafikát láthatunk. Sokáig el lehet nézegetni. Kipróbálható különböző módokban (Graphics 2, 4, 16), más képernyőképekből indulva. A XOR utasítás okozza ennél is a kalamajkát, azaz gyakorlásra mégiscsak jó. A harmadik szintén gépi kódú program *(3. lista)*.A képernyő minden bájtját eltolja biteltoló — SLA — utasítással. Ezt nyolcszor végrehajtva „elporlasztja" a képernyőt, azaz látványosabb CLS-t hajt végre. Más biteltoló utasítással élve hasonlóan érdekes hatást érhetünk el.

Borbás Bence

```
10 **** 1. ****
20 GRAPHICS 4:SET MODE 3
30 X=RND(4)*200:Y=RND(4)*200
40 PLOT X,Y;X+200,Y;X+200,Y+200;X,Y+200;X,Y
50 GOTO 30
```

1. lista

```
10 REM **** 2. ****
20 LOMEM 7000
30 POKE 33,239:POKE 34,25
40 READ A:IF A=555 THEN 100
50 POKE 6639+B,A:B=B+1:GOTO 40
60 DATA 243,62,80,211,2,33,255,127,17
70 DATA 255,127,1,72,63,237,160,226
80 DATA 7,26,174,119,195,253,25,251
90 DATA 201,555
100 REM DEMO
110 GRAPHICS 4
120 LIST
130 EXT0:GOTO 130
```

2. lista

3. lista

```
10 REM **** 3. ****
20 LOMEM 6800
30 READ A:IF A=555 THEN 100
40 POKE 6639+B,A:B=B+1:GOTO 30
50 DATA 243,62,80,211,2,203,38,237,160
55 DATA 234,244,25,251,201,555
100 REM DEMO
110 GRAPHICS 2:LIST
120 FOR I=0 TO 7
130 EXT 0,32767,32767,16200
140 NEXT I
150 END
```

PROGRAMTERMÉK

Ismerd meg anyanyelved!

Meggyőződésem szerint sokkal többen hiszik, hogy jól beszélik a magyar nyelvet, mint ahányan ezt tényleg elmondhatnák magukról. Már számtalanszor vettem részt vérforraló, de halott vitákban, amelyek arról folytak, hogy ez vagy az a kifejezetten magyar szó mit is jelent pontosan. A jövevényszavakról ne is beszéljünk. További gond, hogy néha nagy nevű vagy nagy buzgalmú aktívák „megszólalásig" szimpatikusak. Azután már nem — a katasztrofális beszédhibák miatt. Mintha az alapismereti tárgyak oktatásánál hiányoztak volna az iskolából. Persze az oktatást sem dicséri, hogy csak ennyit tudott ráragasztani az érintettekre.

A fenti „morgás" természetesen megint valami program kapcsán jutott eszembe. Méghozzá nem is egy, hanem két programról van szó. Mindkettő anyanyelvünk jobb megismerését célozza. A programok önmagukban elég rövid lélegzetűek, de jelentőségüket ez nem kisebbítheti. Szerintem a két program együtt kitenne egy nagyobbat, ha szembe kellene állítani őket azokkal, amiket eddig vizsgáltam.

Szójáték

A Szójáték célja a magyar szókincs mintegy 400 szavának felismertetése, olykor kicsit meghökkentő szómeghatározások alapján, három, egyre nehezített megfejtési móddal. A játékot legfeljebb négyen játszhatják. A kérdések száma fordulónként egytől nyolcig terjedhet.

Az első fordulóban megkapjuk a megfejtendő szó összes betűjét ábécérendben (ezt ugye egy profi számítástechnikus az anagrammák kanonikus alakjának nevezné), továbbá egy meghatározást. Az idő telik. Lejárta után megkapjuk a választ. Ha az idő lejárta előtt sikerül megfejteni a szót, akkor a program a versenykiértékeléshez feljegyzi a felhasznált időt. Rossz válasz esetén büntetőidőt számít fel. A kiértékelés pontos algoritmusát sajnos nem lehetett kifigyelni, a leírás pedig homályos.

A második fordulóban csak a meghatározást kapjuk meg, a betűszám és a szó számára pedig egy kipontozott maszk jelenik meg. Ha nem tudunk válaszolni, akkor segítséget kérhetünk. Minden segítségkérésre véletlenszerűen egy-egy betűt kapunk meg a megfejtésből, méghozzá a megfelelő szópozícióban. A segítségért persze büntetőidő jár. Ezért súlyozni kell, mi éri meg jobban, kicsit törni még a fejet, vagy kapásból segítséget kérni. Néhány elég nehéz szóval is összeakadtam.

A harmadik forduló már igazán nehéz. A feltételek elvileg azonosak a második fordulóbeliekkel, de a segítség ravaszul csak a kanonikus anagramma betűit adja meg véletlenszerűen. Azokkal aztán néha nem sokra megy az ember.

Rakéta indul!

A harmadik forduló után a program értékeli a játékosok eredményét, és a legkevesebb időt felhasználó játékos rakétája egy kis animációt beiktatva elindul. Amint a rakéta elhúz, megjelenik a versenyzőnkénti kiértékelés, de érthetetlenül lassan. A versenyzők között van mód a visszavágóra.

A 400 szavas szótár, amelyet a program tárol, nem látszik túl nagynak, és ráadásul a program jelenlegi változatában nincs is mód a cseréjére. A szerző azonban úgy tervezi, hogy további szótárakat fog mellékelni a programhoz, és így újabb változatokat fog megjelentetni. Ha ez tényleg bekövetkezik, akkor a szerző teljesítménye magasabbra értékelendő. A játék közben egyébként a 400 szóból a választás véletlenszerűen történik, és a program arra is vigyáz, hogy elég hosszú legyen az ismétlődés ciklusideje. Újraindításkor a szósorozat elvileg nem ismétlődik.

A program kezelése általában egyszerűnek mondható, bár van néhány apróság, ami boszszantott. Például a kérdésszám 1—8 beállításakor gond van, ha a szám körbepördül. A 8-as után valami érthetetlen okból nehéz elkapni az 1-est. Rendkívül kellemetlen, hogy a törlőbillentyű helyett a csillag gombot kell használni javításra, és ráadásul az egész szót újra kell kezdeni. Ha az ember tévedésből mégis a törlőgombbal javít, akkor igen pórul jár. A törlőgomb ugyanis működni látszik, de a választ a program rossznak értékeli. A kezelhetőséghez tartozó problémának ítélem, hogy a program meglehetősen lassan dolgozik.

A program dokumentációja elég rövid, szövege homályos, és ráadásul nem is elég magyaros, jóllehet a cél éppen a magyar nyelv istápolása. A programban elhelyezett ismertetés egy klasszissal jobb a dokumentációnál. Igazából a dokumentáció nem is nyújt többet nála.

A Szófejtő program kifejezett célja tehát az, hogy a játékosok egymással versengve észrevétlenül mélyítsék el a magyar nyelv kiválasztott szókincsének árnyalatait, és ezzel gyarapítsák intelligenciájukat. A program segítheti a szabadidő értelmes eltöltését akár az iskolában, akár otthon. A program az összértékelésben csak jó minősítést kapott, mint ahogy részleteiben az értékelő táblázat első oszlopából kitűnik. A szerző törekvését mégis nagyra becsülöm, már ami a témát illeti, mert a magyar nyelv ápolására minden befogadható eszközt szükségesnek tartok.

Pótold ki a szavakat!

A fenti program az olvasási nehézségekkel küszködő — dyslexiás — tanulók beszédhibáinak és helyesírási hibáinak korrekciójára szolgáló drill programként született. A szerző profi gyógypedagógus (de sajnos kicsit gyenge programozó, talán jobb lett volna neki némi segítséget nyújtani a programozáshoz). A program elsősorban alsó tagozatos gyerekek oktatására alkalmas. A kísérletek szerint leginkább az értelmileg fogyatékos tanulóknak van hozzá a legnagyobb türelmük. A probléma oka valahol a program sebességében keresendő...

A program szótára 600 szavas. Egy-egy menetben a tanulónak tíz szót kell megfejtenie. A program indulásakor vagy újraindításakor választani lehet 8 gyakorolható „problémakör" közül. A nyolc problémakör a következő: B-P-M, N-M-G, K-G, L-R, GY-NY-LY, L-R-N-J, R-M-N-B és B-D. Nem lennék meglepve, ha a program néhány felnőttnek is okozna egy kis meglepetést. Ahogy a zenében fontos a hallás, ugyanúgy a beszédben is fontos az úgynevezett belső hallás. Mintha belülről valaki súgná nekünk, hogyan is kell kimondani és leírni egy-egy szót. Jaj nekünk, ha a súgás rossz! A korrekció bizony igen keserves munkával jár. Nos, ebben a munkában igen sokat segíthet a kifogyhatatlan türelmű számítógép.

Kell egy ügyes tanár

A program kezelése végtelenül egyszerű, de még ez a végtelenül egyszerű kezelés sincs jól leírva sem az igen lakonikus dokumentációban, sem a program képernyőszövegeiben. A program szinte rá van utalva arra, hogy egy ügyes tanár, aki már kiismerte a működését, szükség esetén ki tudja segíteni a tanulót, hogy most éppen mit kell csinálni. Egyébként rajtakaptam a programot: két egymás utáni menetben előfordult a tízből egy ismétlődő szó. A 600-as szótár és a nem túl nagy tárfoglalás (kb. 13 kbájt) alapján ez nem látszik indokoltnak. Érezhetően nincs beiktatva ismétlődésszűrő. Ez a már említett hiányos számítástechnikai ismeretek számlájára írható.

A számítástechnikai ügyetlenségek ellenére a program jó koncepciójú, a témában alapos felkészültséget mutató taneszköznek minősíthető, már amennyire ezt egy fizikus meg tudja ítélni. Ezért a program összminősítése szintén jó, sőt részleteiben jobban is tetszett, mint a Szójáték. A részletes adatokat a minősítés második oszlopa mutatja.

A két program mind az iskolai munkát, mind az önképzést támogathatja, s véleményem szerint ott a helyük minden iskolában és művelődési házban, ahol a gépi feltételek adottak!

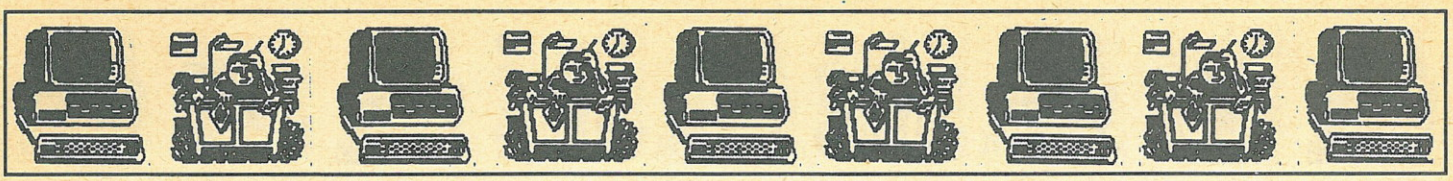
Zsadányi Pál

MINŐSÍTŐ ADATOK

	(1)	(2)
Kezelhetőség:	közepes	jó
Teljesség:	jó	kiváló
Dokumentáltság:	közepes	közepes
Használhatóság:	kiváló	kiváló
Ár/teljesítmény:	jó	jó
Összbenyomás:	jó	jó

ÖSSZEFOGLALÓ ADATOK

Forgalmazó:	Tudományszervezési és Informatikai Intézet, Kereskedelmi Iroda
Terméknév:	(1) Szójáték
	(2) Pótold ki!
Szerzők:	(1) Gombás Károly
	(2) Rezes Ferenc
Géptípus:	(1) TVC, (2) Plus/4 és TVC
Hordozó:	kazetta vagy floppy
Dokumentáció:	(1) öt oldal, (2) két(!) oldal
Ár:	(1) 300 forint (floppy)
	(2) 190 forint (kazetta)



goal (a program addig fut, amíg ez a cél nem teljesül, vagy meg nem állapítja, hogy sosem teljesülhet)

célkifejezés

A Turbo-Prolog 1.1-es verziójánál a célmeghatározás a program végén van. Ez a rész is elmaradhat, de léte vagy nemléte erősen befolyásolhatja a program működését. Ha van, akkor pontosan azt teszi, amire az ezt követő utasítások kényszerítik, és az első kielégítő eset után leáll (hacsak másra nem utasítjuk). Ha nincs, akkor a menüből választott RUN parancsra a DIALOG ablakban megjelenik a GOAL:_ kiírás, ami után az aktuális cél, más néven a kérdés begépelése következhet. Maga a program tehát csak kérdés megválaszolásáért hajlandó futni!

A Prolog program utolsó része a tények és szabályok felsorolása. Ennek a programrésznek a bevezető szava a **clauses.** A tények és szabályok tetszőleges sorrendben követhetik egymást, az azonban fontos, hogy az azonos nevűek egymást követően álljanak, az azon belüli sorrend pedig a végrehajtás sorrendjét fogja befolyásolni. Ennek az az oka, hogy számítógépeink a szabályokat nem egyszerre, hanem szép sorban egymás után dolgozzák fel. Ha majd olyan gépen fut a Prolog, ahol párhuzamos feldolgozás lesz, akkor a szabályok sorrendjére valószínűleg nem kell ilyen megkötést tenni.

A szabály olyan kifejezés, amely egy vagy több kisebb célállítás logikai kapcsolatából áll. Deklaratívan megfogalmazva: a szabály bal oldalán a következmény, jobb oldalán pedig az ahhoz szükséges előfeltételek állnak. Procedurálisan gondolkodva a bal oldalon az eljárásfejet értjük, a jobb oldalon pedig az eljárás törzsét találjuk.

A célkifejezés argumentumlistáiban szerepelhet egy vagy több változó, és az azonos nevű változónevek egy szabályon belül mindig ugyanazt a helyettesítési értéket jelentik. Logikai nyelvről lévén szó, a Prolog-terminológiában logikai változókról beszélnek, ami azonban messze nem azonos például a Pascal boolean típusú változójával.

Kis- és nagybetűk

Fontos szintaktikai szabály, hogy a változónevek **nagybetűvel** kezdődnek, míg a konstans szimbólumok és a szabályok egyaránt kisbetűvel. Az aláhúzásjel kisbetűnek számít a megnevezésekben, önállóan pedig egy olyan argumentum helyét jelölheti egy részcél kifejezésében, amelynek értéke a szabály megoldásában nem változik.

Egy paraméterezett eljárásban nem teszünk formai megkülönböztetést a bemenő és a kimenő paraméterek között. Ha egy változónak van már értéke (foglalt), akkor az a szabály értelmezésekor automatikusan bemenő paraméter lesz, és ha nincs értéke, akkor a szabály csak akkor teljesül, ha ebből a szabályból következően értéket vesz fel, tehát így a határozatlan változó kimenő paraméterré válik. A határozott változók egy szabályon belül nem kaphatnak új értéket, azaz ha egy változó bemenő paraméter, akkor az értéke megmarad a szabály kielégítése után is.

A szabály teljesülése feltételektől függ. A feltételek az if kulcsszó vagy a vele ekvivalens :— jel után állnak, és több szabály esetén logikai kapcsolatukat is meg kell jelölnünk. Ilyen logikai alapszavak és a velük ekvivalens írásjelek a következők:

and illetve , (vessző)
or illetve ; (pontosvessző)
not

Néhány Prolog-implementáció csak az egyik formát fogadja el, mások egy szabályon belül csak egyfajta jelölést engednek meg, a Turbo-Prolog viszont teljes szabadságot ad a program szerkesztőinek e kérdésben. Stilisztikai szempontból azonban jó, ha egységes jelölésrendszernél maradunk. Én például a kulcsszavas írásmódhoz igyekszem ragaszkodni, mert a szavak kifejezőbbek és főleg kevésbé téveszthetőek, mint az írásjelek. Természetesen a szabályok végén minden esetben ott kell lennie a szabályt lezáró pontnak.

A tényállítás konstans argumentumú, feltétel nélküli szabály, a szabály nevével és az argumentumlistával, a végén egy pont.

Makány György

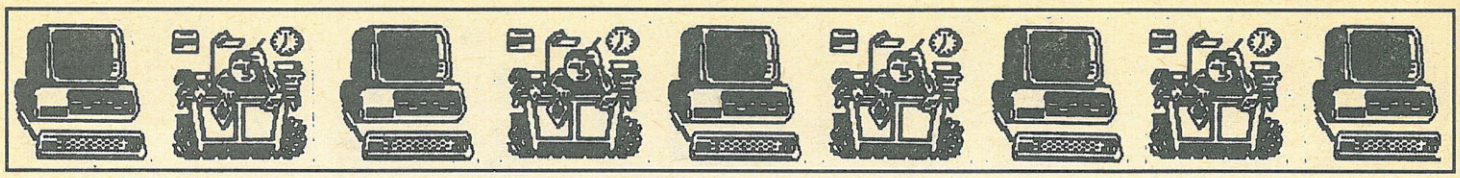
A REALNET és a TVCNET hálózatok

Eredetileg a REALNET a Commodore gépek soros buszának bővítésére készült. Minthogy az iskolai alkalmazásban is igen hasznosnak bizonyult, a fejlesztők célszerűnek látták a — kezdetben csakis hardveresen működő — eszköz továbbfejlesztését oktatóhálózattá. Ez némi hardveres változtatáson túl egy hálózati szoftver kifejlesztését is megkívánta.

A fejlesztés során messzemenően figyelembe vették az iskolai igényeket, lehetőségeket, sőt, még arra is volt gondjuk, hogy a Videoton TV-Computerre fejlesztendő — azóta már kifejlesztett — hálózat szoftverével a lehető legnagyobb mértékben egyezzen. Pedig az ilyen fokú kiszolgálástól, a kompatibilitás követelményétől az iskolákban már-már lemondtunk.

A REALNET hardver-üzemmódja

A REALNET bővítőre — a változatnak megfelelően — legfeljebb 6, illetve 16 soros busszal rendelkező Commodore vagy ilyen csatolóval ellátott más típusú számítógép kapcsolható. A csatlakoztatás a mellékelt kábelekkel történik. A hálózat üzembe helyezése szakértelmet, szerszámokat nem igényel, a hibás összekapcsolásokat az alkalmazott csatlakozók csaknem teljesen kizárják. A mellékelt dokumentáció az üzembe helyezéshez és a használathoz egyaránt elegendő. A hálózatra — az egyedi gépeknél megszokott módon — lehet a perifériákat csatlakoztatni. Ezek használata is a megszokott módon történhet, bár a rendszer néhány tulajdonsága nyilvánvalóan megváltozik.



A hardveres üzemmódhoz — a furcsaságok megértéséhez — érdemes néhány dolgot tudni, átgondolni.

1. A gépek kiszolgálása (a perifériákhoz fordulás lekérdezése) ciklikusan történik. A gépek sorrendjét a csatlakozás helye határozza meg.

2. Ha a soros vonal foglalt, akkor a (többi) gép várakozik a kiszolgálásra. Bár a megoldás néhány esetben túlságosan hosszadalmasnak tűnik, a felhasználó számára ez látszik kényelmesebbnek.

3. A gép és a periféria operációs rendszere között működik a hálózat, így a periféria hibaüzenete is csak a kiszolgálás során jön létre; ez persze egy kiszolgálásnak minősül.

4. Az átlagos várakozási idő a gépek számával nő, így — a soros vonalat gyakran használó programok esetén — a kiszolgált gépek száma és a várakozási idő között kompromisszumot kell kötni.

5. Lehetőség van arra is, hogy valamelyik gép helyére újabb REALNET bővítőt, ehhez pedig további gépeket kapcsoljunk. A gyártó erre az esetre nem garantálja a működést, de a tapasztalatok így is kedvezőek. Nyilvánvaló azonban, hogy két REALNET—6 összekapcsolása helyett célszerűbb egyetlen REALNET—16 alkalmazása, ez utóbbiból kettőt viszont a várakozási idők miatt nem érdemes összekapcsolni.

6. Természetesen perifériás egység(ek)et tartalmazó konfigurációk is csatlakoztathatók a hálózatra, ilyenkor azonban az egy-egy géppel elérhető perifériák egységszámainak különbözniük kell. A közös perifériákat minden gép használhatja, de egymás önálló perifériáit nem érhetik el a gépek.

Ebben az üzemmódban számos felhasználói program futtatható, a perifériák használata a szokásos módon történhet. Azt persze figyelembe kell venni, hogy a perifériákhoz több gép is hozzáférhet.

Szoftver-üzemmód

A hálózati szoftver mintegy „rásimul" a BASIC rendszerre. Az utasítások formalizmusa illeszkedik a BASIC-hez, általában parancs- és programmódban egyaránt kiadhatók. A BASIC program és a változók kiküldésén és bekérésén kívül módot ad tárterület és képernyő áttöltésére, valamint üzenet küldésére. Lehetőség van a rendszer állapotának lekérdezésére és átállítására is, sőt, a tanári géppel még a billentyűzetet is „el lehet venni" a tanulói gépektől. Mindemellett lehetőség van arra is, hogy a hálózatra kapcsolt gépek vegyesen működjenek hardver-, illetve szoftver-üzemmódban.

Meglepetések persze itt is adódhatnak, ám ezek a sajátságok is könnyen érthetők, ha a funkciókat kicsit elemezzük.

— Amíg kivitelnél van értelme több gép egyidejű feltöltésének, addig beolvasásnál ez értelmetlen.

— A vegyesen alkalmazott géptípusok esetén a tárterület mozgatása komplikációkat okozhat a különböző tárfelosztás miatt. Ilyen gond adódhat például a Plus/4 és a nem bővített C16 típusú gépek esetén.

— A már működő rendszerbe való belépést csak egy lekérdező ciklus után nyugtázza a tanári gép. Ez viszont nagyon elhúzódhat egy-egy adatmozgatás miatt.

— A tanulói gépek számának bekérése (itt a rendszer eléggé védtelen!) nem látszik elegánsnak, mégis ez tűnik kényelmesebbnek: a logikai számok ügyes kiosztása egyszerűbb kiszolgálást tesz lehetővé (például ciklusban).

Tapasztalatok

Az elektronika megbízhatóan működik, csak igen kevés meghibásodásról hallani. Sok esetben egy-egy perifériahibát vélnek hálózati hibának.

Néha előfordul, hogy a periféria „nem szólal meg". Ilyenkor általában segít, ha a bővítőt, illetve a perifériát kikapcsoljuk, majd ismét bekapcsoljuk. Ez a jelenség akkor tapasztalható, ha a hálózatra kikapcsolt gép is csatlakozik.

Bár a bővítő elektronikája nem mondható kényesnek, a csatlakozások szabályait érdemes betartani a gépek és a perifériák védelmében is. Néhány ellenkező értelmű tapasztalat alapján ne adjuk fel óvatosságunkat; azokat tekintsük a szerencse ajándékainak!

Egy-egy felhasználói program futtatása során előfordul, hogy például a lemezegység használata közben lemerevedik a rendszer. Ennek oka leggyakrabban az alacsony hálózati feszültségben vagy a hálózati zavarokban (feszültségingadozás, kapcsolási tranziensek) keresendő. A lemerevedés a REALNET ki- és bekapcsolásával rendszerint megszűnik, mindenféle adatvesztés nélkül.

A hálózati szoftver rendkívül megbízhatóan működik. BASIC programok futtatása esetén kimondottan kényelmes szolgáltatásokat nyújt; azzal azonban számolni kell, hogy az adatmozgatási műveletek egymást követően hajtódnak végre, és időbe telik, míg a tanári gép kezdeményezését a tanulói gép érzékeli, nyugtázza.

Vegyük figyelembe, hogy bármilyen gyors is, az átvitelnek látható jelei vannak a tanulói képernyőn. A tanári gép beavatkozásai tehát nem lehetnek „titkosak".

Ha a hálózatot szoftveres üzemmódban használjuk, de hardveres üzemmódú gépet is kiszolgál a rendszer, akkor a tanári gép és egy hardver-üzemmódú hálózatra csatlakozó gép egyidejűleg nem használhatja a soros vonalat. Erre fokozottan kell ügyelni.

A TVCNET felépítése és működése

A hálózat a gépek magnetofon-csatlakozóit használja, és csak szoftveres üzemmódban működik. A tanári géptől indulva sorosan (láncba) felfűzött gépeket speciális kábelek kötik össze, a tanári gépre csatlakoznak a perifériák, valamint a rendszer tartozékaként a hálózatvezérlő. A kábelek hibás csatlakoztatása esetén a hálózat nem működik: bizonyos funkciók nem használhatók.

A gyártó szállít elágazásos hálózatot is külön megrendelés — a hálózat szerkezetének és a kábelhosszak megadásának — alapján. A típusban szereplő szám a hálózatra kapcsolható tanulói gépek számának felső határa.

A hálózati szoftver a tanári gépbe töltődik be először a lemezről, és azonnal elindulva a tanulói gépekbe tölti a megfelelő programot. Ez is öninduló, és a gép számának bekérése után lehetővé teszi a kommunikációt a tanári géppel.

Különleges szolgáltatás a katalógus oldalankénti listázása, ami — a perifériahasználat engedélyezése után — tanulói gépen is kérhető.

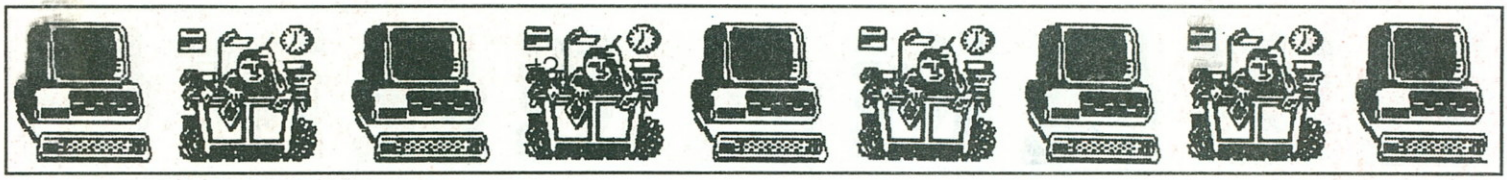
A hálózati utasítások a REALNET hálózatéhoz hasonlók; az eltéréseket a gép sajátosságaihoz való alkalmazkodás indokolja.

A friss fejlesztés miatt még aligha lehet tapasztalatokról számot adni, de annyi már kiderült, hogy a magnót és lemezegységet egyaránt használó programok esetén (könnyen érthető okok miatt) hiba keletkezhet. Ennek elhárítása a kritikus rész átírásával történhet. Az eddig tesztelt programokkal egyéb nehézségek nem adódtak.

Minthogy a hálózati szoftver és a gép rendszere együtt működik, szokatlan eredmények adódhatnak. Ezek egy részét a gyártó már ismeri és vizsgálja; ígéri, hogy a készülő újabb verziókban az ilyen észrevételeket figyelembe veszi. Egyúttal kéri, hogy a felhasználók hozzák tudomására tapasztalataikat, és közöljék kívánságaikat.

Összegezve: a hálózat egyszerű telepíthetősége, a kényelmes szoftver és a két hálózat hasonlósága kedvező előjel az iskolák számára.

Theisz György



A megoldás előtt gondoljuk végig a játék jellemzőit! Először is ez egy olyan feladat, amelyet jó taktikával — azaz a helyes módszerrel — felesleges próbálgatás nélkül is meg lehet oldani. Vezessük ugyanis vissza a feladatot az eggyel kevesebb számú korong esetére, és ha ezt „kivégeztük", már könnyen adódik a megoldás. Tehát ez a feladat tisztán rekurziós jellegű!

Hogy miben is áll a rekurzió, az alábbiakból kiderül. Van N korong a kiindulási rúdon, ezt kell a célrúdra átraknunk. Ha N—1 korongot átrakunk a szabad rúdra, és ha az utolsó korongot átrakjuk a célrúdra, akkor már csak N—1 korongot kell áthelyeznünk a szabad rúdról a célrúdra. A rekurziót tehát kétszer kell alkalmazni.

Ha csak egyetlen korongot kell áthelyezni, akkor nincs szükség rekurzív hívásra, mert egyszerűen ezt kell tennünk és kész.

Bármilyen meglepő, de ezeknek a mondatoknak Prolog nyelvre fordítása már elegendő a rendszernek ahhoz, hogy megoldja a problémát. Annyival azonban tanácsos a feladatot kiegészíteni, hogy miközben a program a korongokat rakosgatja, informáljon bennünket a képernyőn.

Most lássuk a Prolog indítását és futásának eredményét, majd elemezzük részletesen a programot!

```
Goal:hanoi(3)
Tegyél egy korongot a baloldali rúdról a jobboldali rúdra.
Tegyél egy korongot a baloldali rúdról a kozepso rúdra.
Tegyél egy korongot a jobboldali rúdról a kozepso rúdra.
Tegyél egy korongot a baloldali rúdról a jobboldali rúdra.
Tegyél egy korongot a kozepso rúdról a baloldali rúdra.
Tegyél egy korongot a kozepso rúdról a jobboldali rúdra.
Tegyél egy korongot a baloldali rúdról a jobboldali rúdra.True
```

A program listája az alábbi:

```
domains
    rud = jobboldali ; kozepso ; baloldali

predicates
    hanoi(integer)
    tedd(integer,rud,rud,rud)
    inform(rud,rud)

clauses
    hanoi(N) if tedd(N,baloldali,kozepso,jobboldali).

    tedd(1,A,_,C) if inform(A,C) and !.
    tedd(N,A,B,C) if N1=N-1          and
                     tedd(N1,A,C,B) and     /* 1.rekurzio */
                     inform(A,C)    and
                     tedd(N1,B,A,C).        /* 2.rekurzio */

    inform(Rud1,Rud2) if
            write("\nTegyél egy korongot a ",Rud1, " rúdról",
                                        " a ",Rud2, " rúdra").
```

Az első szembetűnő dolog a domains szekcióban van, ahol is a rud típus három különböző konstans értékét veheti fel, úgymint bal oldalit, középsőt és jobb oldalit. Amikor a „predicates" részben a rud típus szerepel, csak ez a három érték fordulhat elő.

A mérlegelt szabályok közül az első a hanoi(N). Ezt kell megadnunk a RUN parancs után — itt N a korongok száma. Funkciója csak annyi, hogy közvetítsen a továbbiak felé, így iteratív szabályként veendő. Mindenesetre magában foglalja, hogy a bal oldali rúdról a középső rúd felhasználásával a jobb oldali rúdra kívánjuk áthelyezni az N darab korongot.

Az inform (rud, rud) szabály olyan eljárás, amelyik kiír egy kerek kis mondatot, amelyik információt ad a teendőről. Az itt működő write beépített eljárás szabályai hasonlatosak a C nyelv hasonló nevű eljárásához: így például az "\n" soremelést idéz elő, és a kiírandókat vessző választja el. Az idézőjelbe tett sztring változatlan, hacsak nem a backslash ("\") mögött áll. A leglényegesebb formák a közölt programokból elsajátíthatók.

A feladat érdemi részéhez most értünk el. A tedd (integer, rud, rud, rud) szabályt úgy értelmezzük, hogy az első paraméter szerinti darabszámú korongot kell a második paraméterként megadott rúdról a negyedik paraméterben kijelölt rúdra helyezni, a harmadik paraméterben meghatározott rúd segítségével.

Ha a korongok száma 1, akkor a dolog egyszerű:

tedd(1,A,_,C) if inform (A,C) and !.

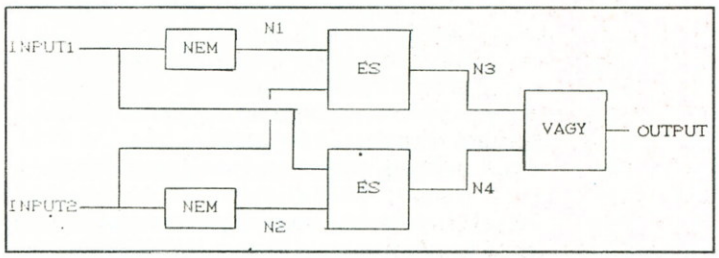
Szavakkal: tedd az A rúdról a C rúdra a korongot, és írd ki a tevékenységet.

A szabály rekurzív része így néz ki:

```
tedd(N,A,B,C) if N1=N-1            and
                 tedd(N1,A,C,B) and
                 inform(A,C)    and
                 tedd(N1,B,A,C)
```

A program első teendője, hogy egy koronggal kevesebbhez állítja a paramétert, hogy előkészítse a rekurziókat. Ha ezt nem tenné, nem vezetné vissza az algoritmust egyszerűbb esetekre, és soha nem érne véget a feladat. Utána az N1=N—1 korongot áthelyeztetjük az ideiglenes rúdra (ez tulajdonképpen ugyanennek az eljárásnak rekurzív hívása), majd ki kell írni az N-edik korong áthelyezését az A rúdról a C rúdra. Ezt teszi az inform(A,C). Miután ez is megvan, már csak az N1 korong ráhelyezése (rekurzió!) maradt hátra — a B-vel jelzett rúdról a C rúdra, a már szabad A rúd segítségével. A rekurziós hívások befejeztével a kimeneten a teendők elolvashatók; a feladatot kompletten megoldottuk.

Makány György



Meleg RESET TVC-n

A TV-Computeren a programok letiltásának leggyakoribb módszere, hogy a 2850-es (0B22H) címen lévő COLD_FLAG rendszerváltozóba 255-öt (FFH-t) írnak, s emellett úgy módosítják a megszakításrutin végét, hogy az a STOP_FLAG rendszerváltozóba 0-t tegyen. Ezek után a gép figyelmen kívül hagyja a CTRL+ESC-t, és a RESET gomb megnyomására "hideg" RESET hajtódik végre.

Az *1. lista* kis programja a megszakításrutin elejét úgy írja át, hogy a COLD_FLAG rendszerváltozóba 0 kerüljön.

Így már nem tilos a "meleg" RESET. A RESET gomb megnyomása után a BASIC program listázható, menthető.

A gépi kódú program BASIC-ből betölthető változata látható a *2. listán.*

Ocskó Tibor

1. lista

```
LD HL,1007H      ;A megszakitásrutin
LD (3E),HL       ;uj kezdozime: 1007H
RET
PUSH HL
LD HL,0B22H      ;COLD_RESET
LD (HL),0
POP HL
JP C412H
```

2. lista

```
10 FOR I=0 TO 16
20 READ A:POKE 4096+I,A
30 NEXT
40 PRINT USR(4096)
50 NEW
60 DATA 33,7,16,34,62,0,201,229
70 DATA 33,34,11,54,0,255,195,18,196
```