

ÚNOR

FUN

1994

with Commodore

časopis uživatelů Commodore 64/128

9. číslo

BASIC

Tipy a triky

Geos - gateway

**Jehličky, inkoust
nebo laser**

Spolehlivý dataset

Programujeme

DEMO

OBSAH

Úvodní slovo	1
BASIC	2
Assembler na C64	5
Tipy a triky	8
Představujeme:	
Geos – gateway	11
Jehličky, inkoust nebo laser	12
Spolehlivý dataset	14
Comotronic News	16
C64 a video - pokračování	18
Programujeme DEMO	21

CO BUDE V PŘÍŠTÍM ČÍSLE?

Assembler

Jednoduchý A/D převodník

Tipy a triky

Program pro opsání–Universal Basic

C64 a video

Comotronic News ...

Adresa redakce: Dolnomlýnská 2, 787 01 Šumperk * Autoři čísla: Jaroslav Vančura a Jiří Kouřil
Podávání novinových zásilek povoleno Oblastní správou pošt v Ostravě pod č.j. 2882/92–P1 ze dne
14. 12. 1992

Podávanie novinových zásielok povolené SP š.p. ZsRP Bratislava č.j. 613–PD–1993 zo dňa 1. apríla 1993
FUN with Commodore – časopis Comotronic klubu pro uživatele počítačů Commodore 64/128.
1 / 94 –9. číslo

Fotosazba REPROtisk J. Kotinský * Tisk: Vegaprint Šumperk

ÚVODNÍ SLOVO

Vážení čtenáři,

záměr napsat, sestavit, vydat a předložit vám za rok 1993 deset čísel časopisu FUN se ukázal být nad naše síly. Rada potíží vedla k tomu, že byla vydána jen polovina zamýšlených čísel, zatímco na druhou polovinu se můžete těšit v letošním roce.

Nedočkavce, kteří by chtěli zbylá čísla dodat během dvou měsíců a a už nyní posílají další předplatné chci požádat o trpělivost. Prosim neposílejte v předstihu žádné peníze. Dnes ještě nedokáži říci, jakým způsobem bude další předplatné organizováno. V každém případě vás o všech změnách budeme informovat dopředu.

Vraťme se však zpět do reality, k časopisu. Velká časová prodleva mezi posledně vydanými čísly pravděpodobně způsobila menší zájem o čtenářské příspěvky, dotazy a inzerci. Proto uvedené rubriky budete dnes hledat marně.

Najdete však ostatní „stálé“ rubriky Basic, Assembler, Tipy a triky. Dále se dozvíte jakým způsobem se programuje Demo a doufám, že pomůžeme i těm, kteří se rozmýšlejí nad koupi tiskárny a nejsou se schopni v široké nabídce typů a výrobců zorientovat. Pro začínající, možná i některé pokročilejší commodoristy je určen stovební návod jednoduchého přípravku pro indikaci správného záznamu na dataset. Nechybí ani další zajímavosti.

Konec konců, posuďte sami! Výsledky soudů a vaše názory bychom ovšem rádi znali také. napište nám je.

*redaktor 9.čísla FUN
Jaroslav Vančura
Šumperk, únor 1994*

Tentokrát chceme ukázat, jak se programuje ve hrách přehled nejlepších dosažených výsledků. Dozvíte se přitom něco o řetězcových proměnných a způsobu přístupu k datovým souborům.

Seznam nejlepších výsledků, anglicky „High Score“, obsahuje nejlepší výsledky, které byly dosaženy při hře. Principiálně se rozlišují tři druhy výsledků:

- celkově (absolutně) nejlepší výsledky
- osobní výsledky
- nejlepší výsledky dne

Celkové výsledky se zaznamenávají na disketu vždy na konci herního kola. Na začátku nové hry se tento výsledek aktivuje, popř. znovu dotáhne. Osobní výsledky se od celkových liší jen v tom, že každý hráč je uveden pouze jedenkrát, přirozeně se svými nejlepšími výsledky. Poslední, výsledky dne, představují nejlépe dosahované výsledky od okamžiku zapnutí počítače, do jeho vypnutí. Výsledky se ovšem neukládají na disketu. Tento seznam vydrží tedy pouze jednu „sezónu“.

V našem případě se budeme zabývat prvými dvěma typy seznamů. Pro realizaci výsledků dne se jednoduše vynechají z programu rutiny pro přístup na disketu (čtení, zápis). Nejprve si prohlédněme celkové výsledky.

Celkově nejlepší

- | | |
|---------------|--------------|
| 1. Karel 678 | 7. Karel 377 |
| 2. Karel 567 | 8. Petr 50 |
| 3. Karel 503 | 9. Katka 33 |
| 4. Monika 445 | 10. Pavel 9 |
| 5. Katka 440 | |
| 6. Niki 390 | |

Osobní výsledky

1. Karel 678
2. Monika 445
3. Katka 440
4. Petr 50
5. Pavel 9

Karel je jednoznačně nejlepším hráčem, pravděpodobně tráví při hrách každou volnou chvíli. Ve výsledkové tabulce obsazuje prvá tři místa. Také Monika si nevede špatně a v tabulce nejlepších výsledků je zastoupena dvakrát. V tabulce osobních výsledků je naproti tomu každý zastoupen jen jedenkrát, přirozeně se svými nejlepšími výsledky. Těmito dvěma poznámkami je v podstatě stanoveno třídění.

TŘÍDĚNÍ

Celkové nejlepší výsledky. Každý výsledek se porovná s prvním záznamem v tabulce. Je-li větší, zůstane na první pozici a ostatní výsledky se automaticky posunou o jedno místo dozadu. Postup se při další hře opakuje. Pokud je výsledek stejný nebo menší, posouvá se seznam. Je-li výsledek horší než poslední místo, může se objevit povzbudivá věta typu „Snad to příště bude lepší...“. Jméno nešťastného hráče zůstane přitom na konci.

U osobních výsledků se nejprve vyzkouší, zda je tento hráč už na seznamu. Pokud ne, jeho výsledek se výše uvedeným postupem zatřídí. Při dalším výsledku od stejného hráče se nejprve vyzkouší, zda hráč svůj výsledek překonal. V záporném případě se může napsat „Byl jste už lepší!“. Když se však hráč zlepšil, vymaže se starý výsledek a nový se zatřídí podle pořadí.

Z těchto podmínek, také však z obou tabulek je zřejmé, že se musíme zabývat problémem umístění jmen, míst a výsledků. Vždyť jen sám Karel zabírá čtyři místa se čtyřmi různými výsledky. Situace vypadá tím hůře, než potom ve skutečnosti je.

První vzorový program (listing 1) realizuje tabulku celkových výsledků s 15 příspěvky. Výsledková tabulka se ukládá na disketu jako pole. V řádku 210 vyhradzujeme pro 15 jmen a 15 výsledků dvě proměnné: NAS(I) pro jména od místa I a PU(I) pro dosažené body na místě I (I od 1 do 15). V tomto řádku se navíc vymazává obrazovka (kód 147). Správné zadefinování výsledkové tabulky na začátku programu zařídí řádek 220. Neobsadí se ani jedno jméno, body se přidělí místům 1 až 15 sestupně od 30 do 2, ve dvojkových odstupech.

POUŽITÍ STARÉ TABULKY

Položme otázku, zda se má naladovat stará tabulka uložená na disketě. Právě tento moment je zvláštností tabulky celkových výsledků. Pokud uživatel zadá <N>, pokračuje se v hlavní části programu, řádek 310, zobrazením nejlepších výsledků uložených v paměti. V opačném případě <J> se otevře sekvenční datový soubor pro čtení. Smyčka v řádku 280 přečte všech 15 výsledků.

Od řádku 310 se na obrazovce objeví „High Score“. Řádek 330 zadá před jméno hráče pořadí. Z funkce RIGHTS se vidí, že pořadí se zadává dvěma znaky. Skóre se zadává v řádku 350. Nyní začíná zadávání jmen a výsledků dosažených hráčem. Pokud se ob-

jeví text „Stačí“, skočí program z řádku 440 na řádek 810. V něm se podle přání uživatele před ukončením programu uloží výsledky ještě na disketu. Starý datový soubor se v řádku 860 vymaže.

Místo hry simulujeme dosažený počet bodů příkazem INPUT. Známým způsobem písmeno po písmeni se zadá jméno. V řádku 450 zadá uživatel dosažený výsledek. Aby nebyl přebrán starý výsledek posledního hráče, nastavíme proměnnou PU předtím na nulu. Řádky v rozmezí 500 až 590 chybí. Později se sem zabuduje část programu, která z celkových výsledků vytvoří osobní. Proto program pokračuje řádkem 610. Seznam se kompletně prohlédá. Najde se místo, na kterém je zanesen první příspěvek, který je menší než aktuální výsledek. Pokud neexistuje skóre nižší než aktuální stav, výsledek se vůbec nedostane do tabulky a v řádku 640 se objeví hlášení.

MICHAL UDĚLAL 400

Vezměme teoretický příklad, kdy Michal udělal výsledek 400 bodů. Není tedy tak dobrý jako Katka se 440, ale je lepší než Niki s 390. Měl by tedy zaujímat 6 místo. Proměnná I (řádek 620) obsahuje v tomto případě hodnotu 6. Stará místa 6 až 10 (v programu 15) se posunou o jedno místo dolů. Položka, která je na samém spodu (15 místo) vypadne ze seznamu pryč.

Udělejme tedy místo pro Michalův výsledek. Začneme dole a postupujme pomalu nahoru. V řádku 730 je příspěvek z dřívější pozice K-1 posunut o jedno místo dozadu (K). Posune se jak název NAS(K), tak počet bodů PU(K). V řádku 740 se nastaví nový výsledek. Opět název a body. Tím se seznam nejlepších výsledků dostane do nového stavu a po odpovídajícím hlášení může být opět zobrazen.

Program je prakticky kompletní. Snad jen ještě několik poznámek. Jak vidno, každý nový příspěvek se zanesení do seznamu již zaříděný. Třídění celého seznamu se v žádném případě nekoná. Lehce se dá také změnit celkový počet záznamů (momentálně 15). K tomu nahraďte všude v programu (v řádcích 210,220,280,320,610,710,880) konstantu 15 jinou hodnotou.

OSOBNÍ VÝSLEDKY

Přejdeme ke druhé variantě, osobním výsledkům. Zde je zastoupen každý hráč jen jedenkrát, přirozeně se svým nejlepší výsledkem. Výsledná tabulka se lehce získá z předchozí. Pomocná rutina se připojí do listingu 1. V paměti jsou totiž pořád ještě celkově nejlepší dosažené výsledky.

```
1000 PRINT „OSOBNÍ VÝSLEDKY“
```

```
1010 FOR I=1 TO 15
```

```
1100 NEXT I
```

```
1110 END
```

Prosím nezapomeňte I za NEXT. První místo z celkových výsledků se přepíše I do osobních výsledků. Test pro první místo se tedy dá ušetřit:

```
1020 IF I=1 THEN 1060
```

Od řádku 1060 zadejme skóre pro první zápis. Okopírujme programové řádky 330 až 350.

```
1060 AS=RIGHT$(STR$(I),2)
```

```
1070 PRINT „MISTO“ AS „:“NAS(I);
```

```
1080 PRINT LEFT$( „ (21x tečka)“, 23-LEN(NAS(I)));
```

```
1090 PRINT RIGHT$( „ ( 5x SPACE)“+STR$(PU(I)),6)
```

V řádku 1100 se pokračuje příkazem NEXT. Pokud program odstartujete pomocí RUN a zadáte několik „Highscores“, potom se dá běh celkových výsledků stiskem <RUN/STOP> přerušit a skokem GOTO 1000 zobrazit nový seznam. Protože chybí ještě zkouška na opakování jmen, vytisknou se už známé celkové výsledky. Přezkoušejme tedy, zda při I > 1 bylo již toto jméno vytištěno. Pokud se tedy NAS(I) vynoří na místě menšího I, přeskočíme tisk a pokračujeme příkazem NEXT dále.

```
1030 FOR K=1 TO (I-1)
```

```
1040 IF NAS(K)=NAS(I) THEN 1100
```

```
1050 NEXT K
```

Aby byly obě smyčky dobře provedeny je nezbytné, aby v řádcích 1050 a 1100 byly za příkazy NEXT správné proměnné (K), resp. (I).

Vyzkoušejte teď vystoupit z hlavního programu a potom nechat vytisknout tabulku příkazem GOTO 1000. Uvidíte, že každé jméno bude vytištěno jen jedenkrát. Seznam má vadu na kráse v tom, že místa jsou číslována po staru. Problém řeší nová proměnná N, která se při každém průběhu zvýší o 1 a vytiskne se namísto I.

```
1005 N=0
```

```
1060 N=N+1: AS=RIGHT$(STR$(N),2)
```

Nyní se dají z celkových výsledků udělat osobní. Nejprve opět vymažeme řádky 1000 až 1100. Pro přetvoření hlavního programu na osobní výsledky je potřeba dále k hlavnímu programu (listing 1) přidat řádky listingu 2. Samotný listing 2 ovšem nefunguje. Aby nedošlo ke kolizi dat na disketě, označme nový seznam jiným jménem „NEJLEPSI/O“ pro „osobní“. V řádcích 270, 860 a 870 je nový název uveden.

Uvažme, co se stane. Potom, co bylo zadáno jméno a počet bodů, zapíše se výsledek do tabulky jen za podmínky, že se tento hráč sám zlepšil anebo dosud vůbec nehrál. Nejprve se tedy vyzkouší, zda „starý známý“ je na seznamu. Pokud ne, jde se dále na řádek 610.

Najde-li se záznam uvedeného jména, odskočí program na řádek 540. Zde vyzkoušíme, představuje-li

uvedený bodový stav dosud nejlepší výsledek hráče. Je-li menší (tedy horší) než nový výsledek pokračujeme v řádku 565 dále. V opačném případě se hráč nezlepšil a seznam nesmí být měněn. (Hlášení „Už jste byl lepší“).

Od řádku 565 se ze seznamu vymaže starý záznam tohoto hráče. Hráč např. byl předtím na 5. místě. Dále se okopíruje jméno a počet bodů hráče ze 6. místa na 5. Potom se I zvýší o 1, tím se provede přesun záznamů dat ze 7 místa na 6, z 8 na 7 atd. až 15 místo na 14. I má nyní hodnotu 15. Program v 570 opustí řádek, v řádku 590 nechá pro jistotu prázdné místo. Hráč byl teď ze seznamu se svým starým, horším výsledkem vyškrtnut a s novým zápisem se může pokračovat od řádku 610.

Listing 1.

```

100 REM * BASIC-Corner C 64
110 REM * HIGHSCORE-LISTE
120 REM * NIKOLAUS HAUSLER
130 REM * ZWENGAUERWEG 18
140 REM * MNICHOV
150 :
160 REM — VERZE 1: NEJLEPSI VYSLEDKY
200 :
210 DIM NAS(15),PU(15): PRINT CHR$(147): REM
    VYMAZAT OBRAZOVKU
220 FOR I=1 TO 15: NAS(I)=,(SPACE)" : PU(I)=32-
    I*2: NEXT
230 PRINT: PRINT,, SEZNAM Z DISKETY ? [J/N]
240 GET AS: IF AS="N,, THEN 310
250 IF AS <> "J,, THEN 240
260 PRINT: PRINT " SEZNAM SE NATAHUJE
270 OPEN 1,8,2,,NEJLEPSI,S,R
280 FOR I=1 TO 15: INPUT #1, NAS(I), PU(I): NEXT
290 CLOSE 1
300 :
310 PRINT: PRINT "NEJLEPSI VYSLEDKY:
320 PRINT: FOR I=1 TO 15
330 AS=RIGHT$(STR$(I),2): PRINT,,MISTO
    "AS,":NAS(I);
340 PRINT LEFT$(,".....",23-
    LEN(NAS(I)));
350 PRINT RIGHT$(,"6SPACE"+STR$(PU(I)),6)
360 NEXT
400 :
410 PRINT
420 PRINT „ZADEJTE -STOP- UKONCITE PRO-
    GRAM“

```

```

430 INPUT "VASE JMENO,,;NAS
440 IF NAS="STOP,, THEN 810
450 INPUT "VYSLEDEK (BODU),,; PU
600 :
610 FOR I=15
620 IF PU > PU(I) THEN 710
630 NEXT
640 PRINT: PRINT" VYSLEDEK ZEL NESTACI NA
    NEJLEPSI !660 GOTO 310
700 :
710 K=15
720 IF K=I THEN 740
730 PU(K)=PU(K-1):NAS(K)=NAS(K-1): K=K-1: GO-
    TO 720
740 PU(K)=PU: NAS(K)=NAS
750 PRINT: PRINT „JSTE MEZI NEJLEPSIMI NA "I,,
    MISTE !760 GOTO 310
800 :
810 PRINT: PRINT" PROCES SE NA VASE PRANI
    PRERUSI
820 PRINT: PRINT,, ULOZIT SEZNAM NEJLEPSICH
    NA DISK [J/N]
830 GET AS: IF AS= "N,, THEN END
840 IF AS<> "J,, THEN 830
850 PRINT: PRINT" SEZNAM SE UKLADA
860 OPEN 15,8,15,,S:NEJLEPSI":CLOSE 15
870 OPEN 3,8,2,,NEJLEPSI,S,W
880 FOR I=1 TO 15: PRINT#3, NAS(I):
    PRINT#3,PU(I): NEXT

```

Listing 2.

```

160 REM —VERZE 2: OSOBNI VYSLEDKY
270 OPEN 1,8,2 "NEJLEPSI/P,S,R
500 :
510 FOR I=1 TO 15
520 IF NAS(I)=NAS THEN 540
530 NEXT: GOTO 610
540 IF PU(I) < PU THEN 565
550 PRINT: PRINT ,, BYL JSTE UZ LEPSI:
560 PRINT " UZ JSTE NAHRAL „PU(I)“ BODU,,; GO-
    TO 310
565 PRINT: PRINT "TIM JSTE PREKONAL SVUJ DO-
    SAVADNI
567 PRINT ,,OSOBNI REKORD "PU(I),, BODU.
570 IF I=15 THEN 590
580 PU(I)=PU(I+1): NAS(I+1): I=I+1: GOTO 570
590 PU(I)=PU(I)=0: NAS(I)=(SPACE)
860 OPEN 15,8,15 "S:NEJLEPSI/P,,: CLOSE 15
870 OPEN 3,8,2,"NEJLEPSI/P,S,W

```

ASSEMBLER NA C64

(8. pokračování)

PŘÍKAZY PRO STACK

Takový způsob programování se nazývá strukturované programování. To je základní idea vyšších programovacích jazyků jako je C nebo Pascal, nikoliv však Basic. V konečném efektu vede strukturované programování k tomu, že Váš program vypadá asi takto:

```
Podprogram 1
Podprogram 2
Podprogram 3
      .
      .
      .
      Část podprogramu
      .
      .
      .
Podprogram X
-----
JSR podprogram A
JSR podprogram B
      .
      .
      Hlavní program
      .
      .
      .
```

Pokuste se tak programovat, pokud je to z hlediska délky běhu programu realizovatelné.

Na závěr ještě příkazové kódy příkazu JSR a RTS a přenos parametrů příkazu JSR:

JSR \$Adresa

JSR: \$20
RTS: \$60

6.2.14 PŘÍKAZY PRO PŘERUŠENÍ BRK A RTI

Ačkoliv o možnostech přerušení procesoru 6510 budeme ještě pojednávat na jiném místě pokud možno přístupnou formou, musíme na tomto místě pro úplnost tyto příkazy představit.

Jedná se o příkazy BRK s příkazovým kódem \$00 a RTI s příkazovým kódem \$40. BRK znamená BReak (přeruš) a RTI ReTurn from Interrupt (návrát z přerušeni). V případě příkazu BRK se jedná o příkaz k vymazání softwarového přerušení. Procesor v takovém příkladě „přistane“ v MONITORU, pokud je v počítači nainstalován. V opačném případě počítač „zamrzne“ a je nutno provést alespoň softwarový RESET. Příkaz RTI je velice podobný příkazu BRK.

Jediný rozdíl je v tom, že u příkazu RTI je vytažen obsah stavového registru ze Stacku a proto slouží tento příkaz k výstupu z podprogramu. Z tohoto důvodu vás nesmí ani napadnout užít tento příkaz místo BRK.

6.2.15 ŽÁDNÁ OPERACE – NOP

U všech procesorů řady 65XX a kompatibilních existuje ještě jeden příkaz, který neslouží vůbec k něčemu. Je to příkaz NOP s příkazovým kódem \$EA. Ačkoliv nemá žádnou funkci, trvá jeho zpracování procesoru nějaký čas a proto je používán ve smyčkách ke zpomalování programu. Dále je tento příkaz užitečný v případě, kdy potřebujeme z nějakého programu odstranit některé příkazy, aniž by bylo nutné celý program přesunovat. V tom případě se napíše do paměťových míst, která před vymazáním byla obsazena různými příkazy příkazový kód \$EA. Pro tento účel se hodí příkaz „F“ monitoru strojového kódu.

7. PRAKTICKÉ VYUŽITÍ STROJOVÉHO JAZYKA

7.1 VÝPIS JEDNOHO ZNAKU NA OBRAZOVKU

Potom, co jsme si v předchozích nezáživných kapitolách poměrně podrobně popsali jednotlivé instrukce procesoru 65XX, dostáváme se konečně k tolik očekávané kapitole, kde se budeme zabývat praktickým využitím strojového jazyka. Využití v tomto případě neznamena tvorbu složitých a optimálních programů. Spíše půjde o přiblížení různých technik a způsobů adresování. Díky tomu je možné, že kromě řešení, které budeme popisovat existuje i jiné, s lepším algoritmem řešení. To je však dáno tím, že my budeme používat postupy názornější, umožňující lépe pochopit navržené řešení.

Takže začneme prvními úkoly, které je možno ve strojáku řešit různými způsoby. Půjde v první řadě o výpis znaku na obrazovku. Protože v strojovém jazyce není žádný příkaz PRINT, uvedeme nejdřív druhou možnost, jak v Basicu zobrazit znak na obrazovce. Dá se k tomu použít příkaz POKE, pomocí kterého můžeme uložit kód znaku do obrazové paměti. Obrazová paměť je ta část RAM počítače, kte-

rou VIC využívá k ukládání obrazových informací. Pokud je například v první paměťové buňce obrazové paměti uložena hodnota 65, znamená to pro VIC, že má v levém horním rohu obrazovky zobrazit znak „A“. K tomu je nutno dodat, že obrazovka je obsluhována po řádcích. To znamená, že první řádek obrazovky má svůj obsah uložen v paměťových buňkách video-RAM číslovaných 0 – 39, druhý řádek 40 – 79, třetí 80 – 119, atd. Obrazová paměť je u počítače C64 uložena v adresovém prostoru \$0400 – \$0800, to je dekadicky 1024–2047.

S pomocí příkazu POKE bude k zobrazení znaku „A“ v levém horním rohu vhodný třeba následující postup:

1. Vymažeme obrazovku pomocí kombinace kláves SHIFT + CLR/HOME. Poté je vhodné přemístit kurzor z rohu poněkud níže několikanásobným stiskem klávesy <RETURN>.
2. Pak zapište příkaz POKE 1024,65 a stiskněte <RETURN>. Pokud chcete zobrazit jiný znak vyberte si jeho kód z tabulky POKE–kódů v originálním manuálu.

Jak uvidíte, objeví se v levém horním rohu znak „A“, pokud na adresu 1024 uložíte kód 65. Pokusíme se teď převést tento postup do strojového jazyka.

Je to již dost dávno, co jsme si objasňovali, že pro práci s pamětí máme k dispozici jen 3 příkazy: STA, STX a STY. Jeden z těchto příkazů proto musíme v každém případě použít. Vezmeme si proto například příkaz STA. Abychom mohli pomocí příkazu STA uložit na adresu 1024 hodnotu 65, musíme mít tuto hodnotu uloženu v akumulátoru (STA znamená Store Accumulator – ulož akumulátor). Praktické řešení při splnění této podmínky vypadá následovně:

Nejdříve zapišeme do akumulátoru hodnotu 65 s využitím příkazu LDA, poté tuto hodnotu přepíšeme z akumulátoru na určenou adresu paměti příkazem STA a nakonec strojový program ukončíme příkazem BRK, což správně zapsáno vypadá v assembleru takto:

MONITOR

```
PC SR AC XR YR SP
; FF00 00 00 FF 00 F8
```

A 0801 LDA #\$41 ; uložení hodnoty 65 (hexadecimálně \$41 do akumulátoru)

A 0803 STA \$0400 ; přepsání obsahu akumulátoru do video-RAM

A 0806 BRK ; přerušení, zpět do monitoru

(Ten text za středníkem do počítače nezapisujte, jsou to jen naše vysvětlivky!)

Vymažte nyní obrazovku kombinací SHIFT + CLR/HOME a stiskněte 5x <RETURN>. Nyní zapište (v monitoru) příkaz

G 0801 <RETURN>.

Jak vidíte, v levém horním rohu se objevilo písmeno „A“. To znamená, že již dokážeme na obrazovku zapisovat jednotlivé znaky. Jak to ale uděláme, když budeme chtít na obrazovku zapsat písmeno „A“ třeba 100x?

Můžeme pochopitelně postupovat následujícím postupem:

```
A 0801 LDA #$41
A 0803 STA $0400
A 0806 STA $0401
A 0809 STA $0402
A 080C STA $0403
```

```
STA $05FF
BRK
```

Tato možnost je však hrozně nepraktická a pomalá. K tomuto účelu bude vhodnější naprogramovat smyčku. V tom případě je více možností. Začneme tou nejjednodušší:

Protože chceme naplnit danou hodnotou jen 100 paměťových míst, nabízí se jako řešení indexované adresování. Nejdříve si připomeneme, co se indexovaným adresováním rozumí:

Při indexovaném adresování se zadá adresa, ke které se postupně přidává hodnota obsažená v registru X nebo Y. Pokud se tedy potřebuje od jisté adresy adresovat ještě 99 následujících adres, tak se ve smyčce přidává do registru X nebo Y hodnota 1. Napišeme si pro tento účel program.

Nejdříve musíme nastavit obsah vybraného registru (ne akumulátoru, protože do toho si uložíme hodnotu znaku „A“) na „0“. Použijeme například registr „Y“. Odpovídající příkaz bude ve strojovém jazyku.

LDY #\$00

Potom musíme do akumulátoru uložit hodnotu kódu znaku „A“, t.j. \$41 příkazem

LDA #\$41

Teď se dostaneme k programování smyčky. První věc, kterou ve smyčce musíme zařídit je, aby se obsah akumulátoru zapisoval od adresy \$0400 plus obsah registru Y. K tomu slouží příkaz

STA \$0400,Y ; zapsat obsah akumulátoru do paměťového místa \$0400+YR

Nyní musíme zvýšit obsah akumulátoru o 1 a testovat, zda již obsah registru Y není 100, t.j. zda se již na obrazovku nezapsalo 100 znaků „A“. Ke zvýšení obsahu slouží příkaz INY a k porovnání příkaz CPY. Proto doplníme náš krátký program o část:

INY ; zvýšit obsah registru
YCPY #\$64 ; srovnat jej s hodnotou 100 (hexadecimálně \$64)

Pokud nebyl obsah registru roven \$64, vymaže se nulový příznak, jinak se nastaví. Tuto smyčku musíme opakovat tak dlouho, až bude při srovnávání nulový příznak vymazán.

Pro skok při vymazaném nulovém příznaku existuje příkaz BNE. Protože v naší smyčce byl první příkaz indexovaný příkaz STA, musíme skočit zpět na něj. A jako každý pořádný program ve strojáku musí být i náš ukončen příkazem BRK, který vede ke skoku zpět do Monitoru. Celý program vypadá následovně:

A 3000 LDY #\$00 ; uložit do registru Y hodnotu 0

A 3002 LDA #\$41 ; uložit do akumulátoru hodnotu kódu znaku „A“ – \$41

A 3004 STA \$0400,Y ; uložit obsah akumulátoru na adresu 0400+YR

A 3007 INY ; zvýšit obsah akumulátoru o 1

A 3008 CPY #\$64 ; porovnat obsah akumulátoru s hodnotou \$64

A 300A BNE \$3004 ; pokud se nerovná, tak skok
A 300C BRK ; zpět do Monitoru

Vymažte nyní znovu obrazovku pomocí SHIFT a CLR/HOME, stiskněte 5x <RETURN>. Nyní můžete zapsat následující řádek a stisknout <RETURN>:

G 3000 <RETURN>

Téměř současně se stiskem klávesy <RETURN> se na obrazovce objeví 100x znak A. Pokud chceme náš program ještě trošku vylepšit, můžeme jej upravit takto:

Pokud naplníme registr Y na začátku programu nikoliv hodnotou 0, ale 99 a obsah registru budeme ve smyčce snižovat, ušetříme jeden příkaz – CPY. Musíme však místo příkazu BNE použít příkaz BMI. Po tomto vylepšení vypadá program takto:

A 3000 LDY #\$63 ; uložit do Y hodnotu 99

A 3002 LDA #\$41 ; uložit do akumulátoru 65

A 3004 STA \$0400,Y ; akumulátor uložit na adresu 0400+Y

A 3007 DEY ; registr Y snížit o 1

A 3008 BPL \$3004 ; pokud větší než 0, pak skok

A 300A BRK ; zpět do Monitoru

Nyní se podrobněji podíváme na funkci programu. Jak vidíte, uložili jsme nejdříve do registru Y dekadickou hodnotu 99. Příkazy, které potom následují, jsou shodné s těmi z prvního programu. Protože však náš druhý program má počítat od 99 k 0 včetně, musí být obsah registru Y snižován o jednu. Protože příkaz DEY ovlivňuje také negativní příznak, můžeme tento příznak využít k našim účelům. Rozsah hodnot, který se u příkazu DEY objevuje, musí ležet v rozsahu 99–0. U žádného z těchto čísel není nastaven 7. bit v binárním vyjádření. Díky tomu nebude nastaven ani nulový příznak, pokud čísla budou ležet v rozsahu 99 – 0. Pokud však dojde k podtečení, objeví se v registru Y hodnota 255. Protože však je tato hodnota záporná, nastaví se negativní příznak a při následujícím příkazu BPL nedojde ke skoku. Vidíte, že troška přemýšlení se vyplatí, zvláště, když jde o 99. násobný průběh smyčky, kdy se nám podařilo odstranit časově náročný porovnávací příkaz.

Až dosud jsme naplňovali jen malé části obrazovky, jak to ale vypadá, když má být vyplněna celá obrazovka? Přirozeně i tu existuje řada možností:

1. Využijeme nepřímé indexované adresování a proběhneme 3x smyčku 0 – 255 a jedenkrát smyčku 0 – 232, protože video-RAM má právě 1000 bajtů.
2. Pracujeme s vnořenou smyčkou a dělíme obrazovku například na 25 částí po 40. znacích (25 * 40 = 1000).

Jak z hlediska běhu programu, tak pro přehlednost je jistě druhá varianta lepší. Z toho důvodu si princip tohoto řešení přesně popíšeme:

Pro naplnění paměti o kapacitě přesně 1000 bajtů se rozdělí tento rozsah na 25 částí po 40 bajtech. Potom si navrhne program, který proběhne 25x smyčku od 0 do 39. Pro tento účel do sebe vnoříme dvě smyčky. Po každém průběhu vnější smyčky zvýšíme počáteční hodnotu vnitřní smyčky o 40. V BASICu to bude vypadat takto:

```
100 AD = 1024: REM startovní adresa obrazové paměti
110 FOR Y = 0 TO 24: REM vnější smyčka pro radky
120 : FOR X = 0 TO 39: REM vnitřní smyčka pro sloupce
130 : : POKE AD+X,65: REM kod A do paměti
140 : : NEXT X: REM X zvýšit o 1
150 : : AD = AD+40: REM AD zvýšit o 40
160 : : NEXT Y: REM vnější smyčku zvýšit o 1
```

Pochopitelně by bylo možno vnější smyčku programovat také s využitím příkazu STEP 40, ale přenos do strojového jazyka by pak vypadal složitěji. Pokud se nyní pokusíme program přepsat do strojáku, bude princip zcela stejný. K tomu účelu musíme nej-

dříve, obdobně jako v řádku 100 stanovit počáteční adresu obrazové paměti. Protože budeme pracovat s nepřímým indexovaným adresováním, bude nejlepší použít k tomuto účelu 2 adresy v nulté stránce. Nabízejí se adresy \$D8/\$D9, které nejsou využity operačním systémem. První 4 příkazy našeho strojového programu musí pak vypadat takto:

```
LDA #$00 ; LOW-bajt počáteční adresy
LDY #$0C ; HIGH-bajt počáteční adresy
STA $D8 ; uložit nejdříve LOW
STY $D9 ; a pak HIGH bajt
```

Nyní musíme ještě připravit indexový čítač pro řádky. Nejlépe se pro tento účel hodí registr X, protože registr Y budeme potřebovat při nepřímém indexovaném adresování, které s registrem X nepracuje. Pro přehlednost naprogramujeme obě smyčky se srovnávacím příkazem, tedy ne s programátorskými triky například v podobě BPL. Takže nejprve uložíme do registru X hodnotu \$00:

```
LDA #$00 ; uložit 0 do registru X
Potom musíme na nulu nastavit také registr Y.
```

(jk)

TIPY A TRIKY

Programátoři Basicu neznají nikdy dost triků. Proto jim přinášíme pár nových, které jim snad pomohou.

ZÁCHRANNÁ BRZDA

Při odlaďování programů v Basicu působí velmi rušivě, že listingy v mžiku přeběhnou přes obrazovku. Stisk klávesy CTRL je zabrzdí jen velmi málo. Nebylo by tedy na škodu moci si úplně zastavit proces listingu na předem určeném místě. Potom by ovšem mělo být možno v listingu pokračovat. Jistě uznáte, že stisk klávesy klávesy STOP problém neřeší. Krátkou utilitu, která s jistotou pomůže, opišete za necelé dvě minuty.

```
1 FOR I=1013 TO 1023: READ X: POKE I,X: NEXT
2 DATA 174,141,2,202,240,250,177,95,76,26,167
3 POKE 774,245: POKE 775,3
```

Po startu příkazem RUN se do paměti od adresy 1013 nainstaluje jen 11 bajtů dlouhá záchranná brzda. Klávesou SHIFT-LOCK se dá listing zastavit na libovolně dlouhou dobu. Stisk klávesy Commodore vrátí listing do normálního tempa, CTRL jej přibrzdí, na co už jste ostatně také zvyklí.

Pokaždé, když má počítač vytisknout písmeno, natáhne předtím obsah paměťového místa 653 (SHIFT FLAG) do x-registru a zkouší, zda se jedná o 1.

START S RESTORE

Mnoho komerčních programů se dá klávesou RESTORE znovu odstartovat. Chcete být vlastníky takových programů? Žádný problém. Pokud programujete ve strojovém kódu, napište jednoduše do vlastního programu dva příkazy POKE (ve strojovém kódu tuto funkci plní příkazy LDA # a STA)

POKE 770,L a POKE 771,H

Hodnoty L a H se přepočítají podle následující rovnice:

$H = \text{INT}(AD/256)$; $L = AD - H * 256$

AD (AD=L+256*H) je startovní adresa strojového programu (L=Lowbyte a H=Highbyte AD). V tabulce je uveden příklad.

AD	L	H
820	52	3
4096	0	16
32768	0	128
49152	0	192
50000	0	195

V Basicu je příklad o něco těžší. Pomůžeme si nejprve pomocí <RUN STOP/RESTORE>. Tím se vystoupí z programu, ovšem místo obvyklého hlášení READY se nechá provést RUN. Program z Basicu se tím znovu odstartuje. Na začátku programu musíte uvést následující řádky:

```
0 FOR H=53242 TO 53247: READ X:
  POKE H,X: NEXT:
  DATA 32,142,166,76,174,167
1 POKE 771,207: POKE 770,250: CLR
```

Po <RUN STOP/RESTORE> nebo po chybovém hlášení začne program od začátku. Neodstartuje-li se po <RUN STOP> program znovu, musí se pomocí POKE 788,52 zablokovat STOP. Po ukončení programu vypisuje počítač systémové hlášení READY. Chcete-li změnu, ukončete program nikoli повеlem END nebo STOP, nýbrž SYS 42115. Počítač okamžitě přejde do přímého módu, aniž by předtím ohlásil READY.

OCHRANA DAT

Každý si sem tam přeje uložit na disketu program tak, aby k němu mimo jeho osoby neměl přístup nikdo jiný. Následující trik znesnadní loading i pokročilému uživateli. K vyzkoušení je potřeba mechanika 1541, resp. kompatibilní.

Program v Basicu se dá uložit tak šikovně, že se v adresáři objeví jako SEQ nebo USR. K provedení zadejte za název typ datového souboru. Např.

```
SAVE „PRIKLAD,S“,8 nebo SAVE „PRIKLAD,U“,8
```

V directory uvidíte, že se z vašeho prvního programu stal datový soubor SEQ, z druhého USR. Schválně si vyzkoušejte takový datový soubor natáhnout obvyklým způsobem:

```
LOAD „PRIKLAD“,8
```

Zaručeně se objeví hlášení ?FILE NOT FOUND ERROR a rozblíká se červená LED. Program se může naládat pouze tím způsobem, jakým se předtím uložil, tedy:

```
LOAD „PRIKLAD,S“,8 a LOAD „PRIKLAD,U“,8
```

podle typu souboru. Ještě tajemněji se dá program uložit s nulovým bajtem v názvu:

```
SAVE CHR$(0) + „PRIKLAD“,8
```

V directory se potom objeví zašifrovaný název. Neodpovídá ani délka programu (okolo 10 000 bloků). Ve skutečnosti datový soubor není ani tak dlouhý. Zpět do počítače může program naládat jenom ten kdo zná fintu:

```
LOAD CHR$(0) + „PRIKLAD“,8
```

Jinak to nejde !

JOKER

Pomocí „Jokeru“ „*“ a „?“ se dá přesně specifikovat způsob vyjádření directory. Abyste dostali v directory jen datové soubory označené PRG, napište jednoduše:

```
LOAD „$0:*=P“,8: LIST
```

Písmeno P můžete nahradit písmeny S,U nebo R. Potom vylistujete všechny SEQ, USR, resp. REL datové soubory.

Naládování programu u něž znáte jen začátek názvu zajistí zápis:

```
LOAD „NAZ*“,8
```

To už asi víte. Rovněž tak je možno nahradit všechny neznámé znaky otazníky:

```
LOAD „FILENA?E“,8
```

Víte ovšem, že tato hra funguje také v directory? Chcete-li nechat vytisknout všechny datové soubory začínající MAT, které mají páté písmeno M, zadejte:

```
LOAD „$:MAT?M*“,8: LIST
```

STRUKTURA JMÉNEM „PŘEHLED“

Jako vtip zní věty ve kterých se hovoří v Basicu 2.0 o přehledném programování. Velmi často vypadají programové řádky následovně:

```
10 FORI=1TO20:FORJ=1TOI:A=4:GOSUB400:
  NEXTJ,I
```

Zkráceně napsáno – mimořádně úsporná metoda (36 bajtů), průhlednosti listingu však cosi chybí. Následující zápis je již pro oko poněkud přijatelnější.

Pro každé návěští byl použit zvláštní řádek:

```
10 FOR I=1 TO 20
20 FOR J=1 TO 130 A=4
40 GOSUB 400
50 NEXT J60 NEXT I
```

Listing spotřeboval teď o 30 kB více paměti. Pro další zpřehlednění výpisu doplníme text na některých řádcích mezerami. Při zadávání všechno perfektně klope. Po prvním LIST však člověk pozná, že veškerá námaha přišla v niveč. Listing ničeho ne dbá a přináší jen jednu mezeru za číslem řádku. To ostatně dělá interpreter zcela automaticky po převzetí řádku stiskem <RETURN>.

Problém odstraní uvedení dvojteček:

```
10 : FOR I=1 TO 20
20 :   FOR J=1 TO I
30 :     A=4
40 :     GOSUB 400
50 :     NEXT J
60 : NEXT I
```

Nevýhody rozhodně nechci zamlčovat. Krátký listing se prodloužil o dalších 28 kB. Mimo to se program zpracovává delší dobu, zato se však případné chyby snadněji odhalují!

TAJEMSTVÍ KLÁVES SHIFT

Možná, že už víte, že obě klávesy SHIFT se dají osahávat odděleně. Prohlédněte a vyzkoušejte následující listing:

```
10 POKE 56334,0
20 POKE 56320,253: IF PEEK (56321)=127
   THEN 100
30 POKE 56320,191: IF PEEK (56321)=239
   THEN 200
40 POKE 56334,1
50 PRINT „ZADNA KLAVESA SHIFT“
60 RUN
100 PRINT „LEVA KLAVESA SHIFT“
110 RUN
200 PRINT „PRAVA KLAVESA SHIFT“
210 RUN
```

V řádku 10 odpojíme systémové osahávání klávesnice, v řádku 40 ho opět zapneme. Klávesnice C64 je zapojena v matici 8 řádků x 8 sloupců. V řádku 20 způsobí příkaz POKE, že se testuje 2. řádek matice. V křížení s 8. sloupcem leží klávesa SHIFT (je identická s klávesou SHIFT LOCK). Je-li odpovídající bit nenastaven (0), vytiskne se v řádku 100 text. Ob-

dobně zkouší řádek 30 zda byla stisknuta pravá klávesa SHIFT (matice: řádek 7, sloupec 5). Této techniky se dá využít k zabudování do vlastních programů., např. pro řízení kosmického korábu. Pokud se nemusí mezi oběma klávesami rozlišovat, jde to také jednodušeji:

```
10 A=PEEK (653)
20 IF (A AND 1) THEN PRINT „SHIFT “;
30 IF (A AND 2) THEN PRINT „COMMODORE “;
40 IF (A AND 4) THEN PRINT „CTRL “;
50 PRINT: RUN
```

Program je schopen rozpoznat stisk tří zmíněných kláves (SHIFT levá a pravá, Commodore a CTRL).

DVOJITÝ PEEK

Pokud chce člověk zjistit, kde program v Basicu končí v paměti, zadá:

```
PRINT PEEK (45) + 256*PEEK (46)
```

Jako výsledek se obdrží první paměťová buňka za programem. Paměťové buňky 45 a 46 obsahují tuto adresu (ve formátu Highbyte–Lowbyte). V příručce jsou v přílohách popsány ještě další důležité adresy paměti. Pokud se s několika triky nasadí funkce DEF FN, může se hořejší řádek napsat podstatně kratším a elegantnějším způsobem.

```
DEF FN DEEK (X)= PEEK (X) + 256*PEEK(X+1)
```

Určitě řeknete: „Řádek je přece delší!“ Ovšem pro vlastní test použijete teď vyjádření

```
PRINT FN DEEK (45)
```

Další důležité adresy tohoto druhu jsou

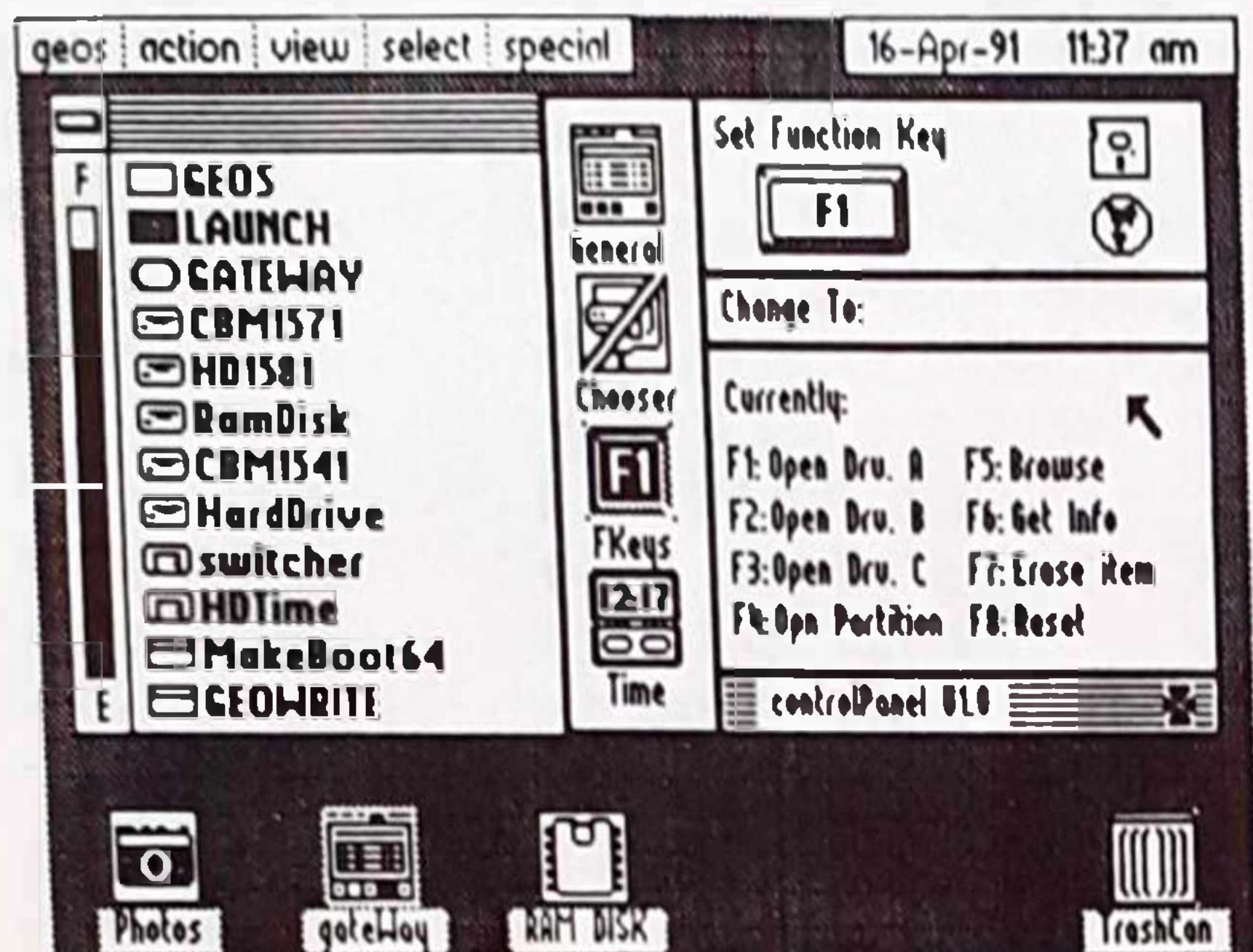
43/45	začátek programu v Basicu
57/58	okamžikové číslo basicovského řádku
59/60	předchozí číslo basicovského řádku
63/64	okamžikové číslo data řádků
187/188	ukazatel na poslední název file
209/210	ukazatel na start aktuálního řádku obrazovky
785/786	vektor USR
788/789	vektor přerušeni
792/793	vektor pro RESTORE

Všechny adresy mohou být čteny shora popsanou funkcí DEEK.

PŘEDSTAVUJEME

GATEWAY – BRÁNA DO NEBE NEBO DO PEKLA ?

Pozoruhodné přístroje americké hardwarevé a softwarové firmy CMD, harddisk 200 MB, flopy 3,2 MB či RAM disk 1 až 16 MB lze provozovat tak, že emulují známé commodorácké jednotky 1541. Jejich hlavní výhoda je však v možnosti nakonfigurování tzv. nativního módu, tedy maximální kapacity.



Aby zmíněné periferie bezproblémově komunikovaly i s Geos, podpořila je CMD vlastním softwarem pod názvem Gateway. Formálně se jedná o náhradu dobrého starého desktopu, byť rozšířeného o řadu funkcí. Z nich jistě zaujme možnost změny velikosti pracovního stolu či jednodušší provádění předvoleb.

Pravá síla je však skryta jinde. Gateway umožňuje využít pod Geos plnou kapacitu CMD přístrojů.

Pokud má člověk k dispozici různorodé přístroje (RAM-link, harddisk, paměť REU 1750 či flopy FD4000), je konfigurace dost proble-

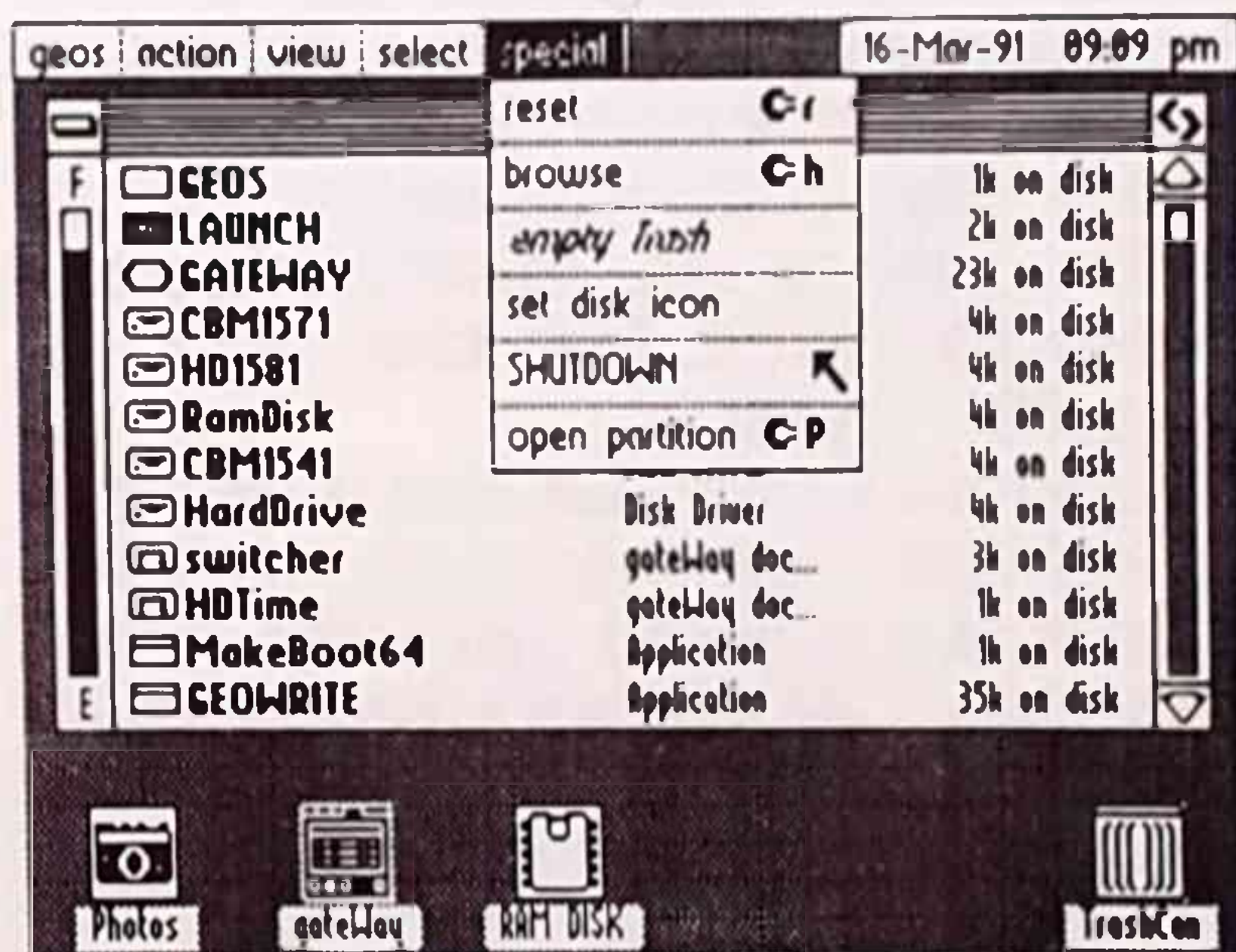
matická. Gateway používá pro řízení dražů, k nimž patří i RAM-link, vlastní budiče, které se na začátku naladují do počítače (boot). Konfigurační program v podstatě neexistuje, konfigurace spíše představuje pořadí v němž jsou na disketě pro booting seřazeny jednotlivé budiče. Pokud se zamění budiče, s jistotou se dostaví problémy. Naštěstí příručka pro Gateway počítala i s touto možností a proto je v ní uvedeno i pár tipů.

Naproti tomu opětovné nastavení není žádnou těžkou prací. Co je však chudší a na co jste zvyklí, je absence krásných bohatých ikon. Místo nich jsou malé symboly disket resp. pracovního stolu. Mimo to se názvy programů zobrazí jen v jednom sloupci a to i při maximální velikosti pracovního stolu.

Ikonou se šipkou na pravé straně se dá pracovní list bez námahy skrolovat. Nepřipomíná vám to již trochu PC?

Zdařilá je také kopírovací funkce. Jako obvykle se soubor dvakrát odklikne a ikona s duchem se přesune na cílovou disketu. Nakonec se otevře okno informující o průběhu kopírovacího procesu (ukazuje názvy a počty kopírovaných souborů).

Operace s více datovými soubory jsou možné



stejně jako z desktop. Provoz FD4000 v nativním módu je až na maličkosti špičkou. Formátování diskety se doporučuje provést předem (boot součástí diskety dodávané s FD4000).

Počet možných připojitelných disketových jednotek závisí na rozšíření paměti. Bez REU maximálně 2. S externí pamětí to potom jde samo. Čím je paměť větší, tím lépe. Se 128 kB připojíte až 3 mechaniky. S 256 kB se dá zřídit už RAM disk. Pokud připojíte 512 nebo více kB paměti, ukážou se veškeré přednosti Gateway. Můžete nakonfigurovat připojení 3 jednotek, z nichž jedna bude RAM disk a switcher. Switcher umožňuje přepínání mezi různými programy bez toho, aniž byste museli program ukončit.

Shrnutí

Gateway je přesně článek chybějící do řetězce produktů od CMD. Bez něj mají harddisk, RAM-link i FD 4000 pro Geos jen poloviční hodnotu. Pro práci však vyžadují prostředí min. Geos 2.0.

Program distribuuje fa Comotronic, u které je možno získat i bližší informace.

JEHLIČKY, INKOUST NEBO LASER ?

Následující příspěvek napoví, pro kterou tiskárnu se rozhodnout. Zda pro ink-jet, laserovou nebo osvědčenou jehličkovou.

Mezi tiskárnami připojitelnými k C64 řadu let dominují jehličkové tiskárny. Poslední dva roky však jejich postavením otrásají, díky rapidnímu poklesu cen, tiskárny inkoustové (tryskové) a laserové. Před koupí se tedy vyplatí udělat rozvahu.

O inkoustových tiskárnách vlastně dlouho nikdo nemluvil i když byly na trhu i předtím. Postarali se o to giganti typu Epson. Řečeno jinými slovy. Lidé zatvrzele kupovali přes velkou reklamu na ink-jety dál tiskárny jehličkové a nedbali hluku, který tyto tiskárny při práci doprovází. Inkoustové tiskárny byly zatracovány, ne neprávem, pro vysychání barvy, mohly tisknout jen na speciální papír a byly pomalé.

Čestnou výjimku, která napravila reputaci inkoustových tiskáren, představuje Hewlett-Packard-Desk-jet. Najednou se objevila tiskárna, která byla cenově přiměřená, psala stejně hezky jako laserovka a přitom byla stejně rychlá. Zkrátka ideál pro soukromníky. Nejlepší na tom bylo to, že nepohrdla žádným druhem papíru.

Potom se s podobnými tiskárnami doslova roztrhl pytel. Mnoho z nich se však k C64 nedá bezproblémově připojit, neboť jak známo, většina programů pro C64 je vybavena pouze budiči tiskáren kompatibilních Epson nebo IBM. S tiskárnami HP si může C64 začít něco jen prostřednictvím Geos. Proto se také C64 inkoustovým tiskárnám dlouho bránil. Experiment Commodore s názvem MPS 1270 rozčaroval neuspokojivou kvalitou tisku. Boom inkoustových tiskáren má tedy u C64 zpoždění.

O něco později došlo potom i ke snížení cen laserových tiskáren. V roce 1992 se objevila první vlaštovka s cenou pod 2000,- DM (Seikosa OP 104). Za další rok potom přišla OL-400e s cenou blízkou 1400,- DM. Pro připojení těchto tiskáren k C64 platí to stejné, co pro inkoustové tiskárny. Pokud jsou tyto přístroje kompatibilní Epson a IBM, lze je na C64 bez větších problémů připojit. Výrobky s emulací HP zůstávají opět vyhrazeny Geosu. Nabízí se tedy otázka, který druh tiskáren je konkurentem jehličkových a další otázka, jaké výhody a nevýhody mají jednotlivé druhy tiskáren?

Inkoust

Inkoustové (tryskové) tiskárny pracují všechny na principu vytváření písmen z jednotlivých kapiček barvy slitých na papíře. Jakým způsobem se dostane barva na papír je otázkou použité techniky. V zásadě se rozlišují dva principy. Buble-jet byla vyvinuta Canonem, zatímco Epson favorizuje piezotechniku. Při prvním principu se inkoust ohřívá topnými články. Tím vznikne z odpařené barvy bublina, která zabírá větší objem, než tekutá barva. Zvětšení objemu způsobí vystříknutí kapiček barvy z malých trysek. V druhém případě se k vypuzení barvy z trysek využívá piezoelektrických vlastností krystalů, kdy se přiloženým napětím krystaly roztáhnou.

Způsob jakým se z jednotlivých kapek barvy

tvoří písmo se velmi blíží jehličkovým tiskárnám. Je pochopitelné, že inkoustové tiskárny chtějí správný papír. Všichni výrobci sice ujišťují, že ta jejich tiskárna píše na všechny druhy papíru, sami jsme však měli možnost se přesvědčit, že tomu tak docela není. Z experimentů vyplývá, že vhodné jsou papíry, které příliš nesají barvu. Papír nesmí také sát málo, neboť barva by dlouho schla a navíc povrch papíru by nebyl hladký. Nejlepší je hlazený, lehce savý papír.

Pokud se použije správný papír, výsledky jsou překvapivě dobré. Výhodou těchto tiskáren je i skutečnost, že s relativně nízkými náklady se dá tisknout i barevně. Barevné tiskárny přijdou jen o něco draže než černobílé.

Laser

Všeobecný pojem laserová tiskárna zahrnuje stroje pracující několika možnými principy. Vlastní „laser“ je jedním z nich. Další tiskárny pracují tzv. postupem LED. Jednotlivé principy mají však jedno společné. Celá stránka se sestaví a vytiskne najednou. Obraz se vytvoří na fotoelektrickém bubnu, který se osvětlením elektricky nabije. Ponořením bubnu do toneru se elektricky nabitá místa obarví. Potom se toner z bubnu přenesení na papír a na něm se teplem zafixuje. Výsledek má prakticky kvalitu strojního tisku. Teprve při silném zvětšení se vidí, že litery laserového tisku jsou tvořeny z jednotlivých bodů. Pro technickou náročnost laserových tiskáren se v současné době používá jen jednobarevného tisku.

Co je lepší? Odpověď na otázku je třeba hledat v předchozích odstavcích. Inkoustové tiskárny mohou mít malé rozměry a hodí se proto využívat jako kompaktní pracovní tiskárny pro mobilní provoz. Příkladem mohou být tiskárny Brother HJ-400 nebo Canon BJ-10SX. Jejich nevýhodou vzhledem k laserovým tiskárnám je nižší rychlost tisku vícestránkového dokumentu. Na druhé straně mají dlouhou životnost, tisknou levně a kvalitou tisku se blíží laserovým. Příjemný je i bezhlučný provoz a tisknutí okamžitě po zapnutí přístroje. Stejně jako laserové ani inkoustové tiskárny netisknou s průklepem. Kdo tedy tiskne protokoly s více kopiemi, zůstane pravděpodobně nadále věrný jehličkové tiskárně.

Z technických a technologických důvodů nebudou laserové tiskárny nikdy tak málo hmotné a kompaktní jako inkoustové. Proto se hodí především pro profesionální nasazení. Kresba laserové tiskárny je lepší než u inkoustové. Nevýhody byly částečně uvedeny. Patří mezi ně robustnost strojů, doba náběhu před tiskem, vysoká hmotnost a v neposlední řadě vysoké nároky na servis. Jako u každého přístroje, kdy se pracuje s tonerem, musí se i laserová tiskárna pravidelně udržovat a čistit. Citlivý fotoelektrický buben se může špatnou obsluhou zničit k nepotřebě a částka vynaložená k jeho výměně se pohybuje kolem 8000,- Kč.

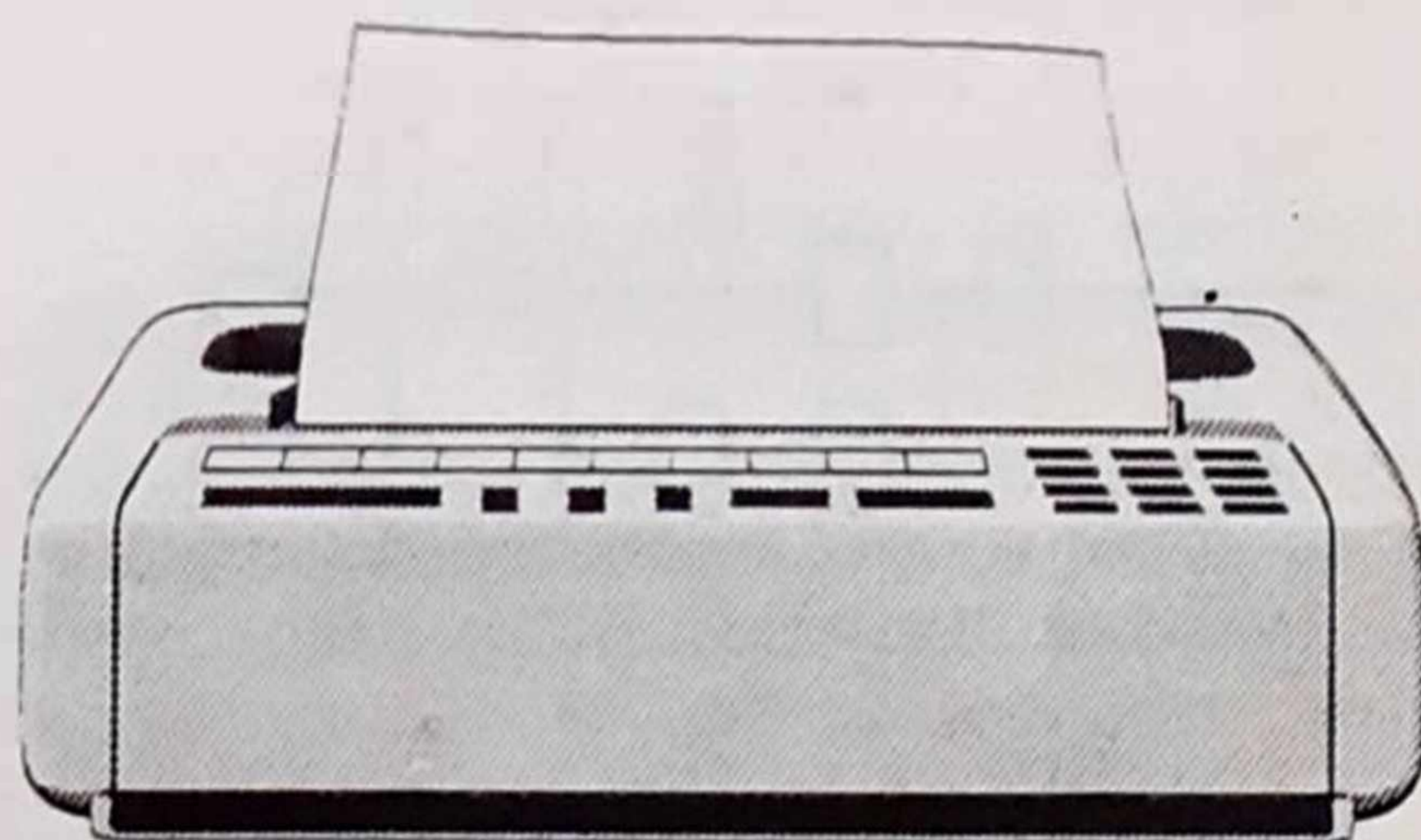
Ve věci nákladů na provoz jsou oba druhy hodnocených tiskáren zhruba na stejné úrovni. Patrona s inkoustem pro ink-jet se vyměňuje častěji než toner, který je naopak dražší. Přesné porovnání a zhodnocení může být provedeno pouze mezi konkrétními druhy. Ceny se liší model od modelu, výrobce od výrobce.

Závěr

Velmi brzy trh ukáže, že inkoustové tiskárny jsou pro domácí počítače ideální. Lze předpokládat, že časem plně nahradí dosud rozšířené jehličkové tiskárny. Už proto, že se principiálně dají vyrobit levněji. V profesionální oblasti se zcela jistě uplatní spíše laserová tiskárna, neboť jejich cena se neustále snižuje a kvalita tisku je nepopiratelná.

Jedno je jisté, jehličkové, maticové tiskárny jsou pro budoucnost ztrátou. I uživatelé C64, dnes spokojeni s jakoukoli tiskárnou, budou chtít zítra za rozumnou cenu vysokou kvalitu, kterou jim zajistí jen laser nebo trysky s inkoustem.

(JV)



SPOLEHLIVÝ DATASET

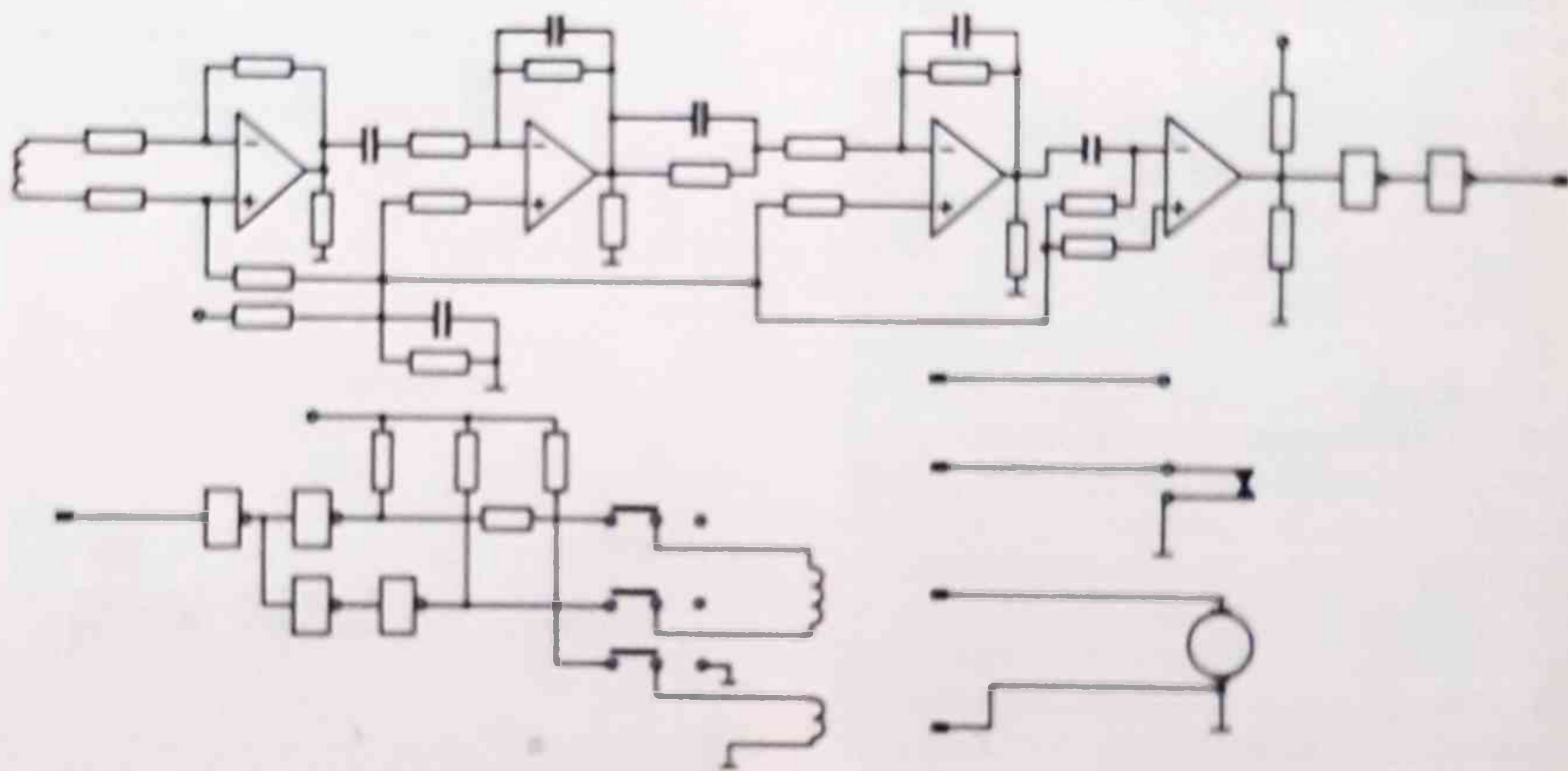
Nejčastější chybou, která se vyskytuje u datasetů je přestavená záznamová hlavička. Nejpravděpodobněji se s tímto problémem setkáte při práci s programy zaznamenanými v režimech Turbo. V dalším textu je popsáno zapojení, kterým se dá jednoduše indikovat postavení tónové hlavy vzhledem k záznamu vložené kazety s programy.

Pro justování datasetu (nastavení kolmosti hlavy) existují v zásadě dvě možnosti, mají elektronický obvod, jímž se hlava nastaví nezávisle na počítači anebo program, který to též zobrazí grafickou formou na monitoru. Pokud však uvedený program nemáte v modulu, resp. v záloze na disketě, nemáte jej jak dostat do počítače a jste odkázáni na elektronický obvod. Jistě se shodneme na tom, že opisování justovacího programu je pro jednobřelovost jeho použití poněkud nepohodlné.

Abyste bylo zapojení co nejsrozumitelnější, zmíním se pár slovy o elektronice datasetu. Sestává ze dvou obvodů, dvoustupňového zesilovače a tvarovače. Úloha zesilovače je jasná, zesí-

lit analogový signál přicházející z tónové hlavy. Analogový proto, že digitální signály se nedají na pásek uložit. Pokud budete experimentovat a pošlete na hlavu digitální signál, neuloží se na pásek v této formě, nýbrž jako sinusovka. Při loadingu se potom musí sinusovka znovu přetvořit do formy, které rozumí počítač, do pravouhého signálu. Transformaci zajišťuje druhý obvod - tvarovač, tvořený hradly se Schmittovými klopnými obvody. Na výstupu Schmittových obvodů je signál ve formě pravouhlých oscilací o amplitudě 0 až 5 V. Digitální signál se pro nastavování tónové hlavy nehodí, neboť amplituda signálu nezávisle na postavení hlavy osciluje mezi 0 a 5V.

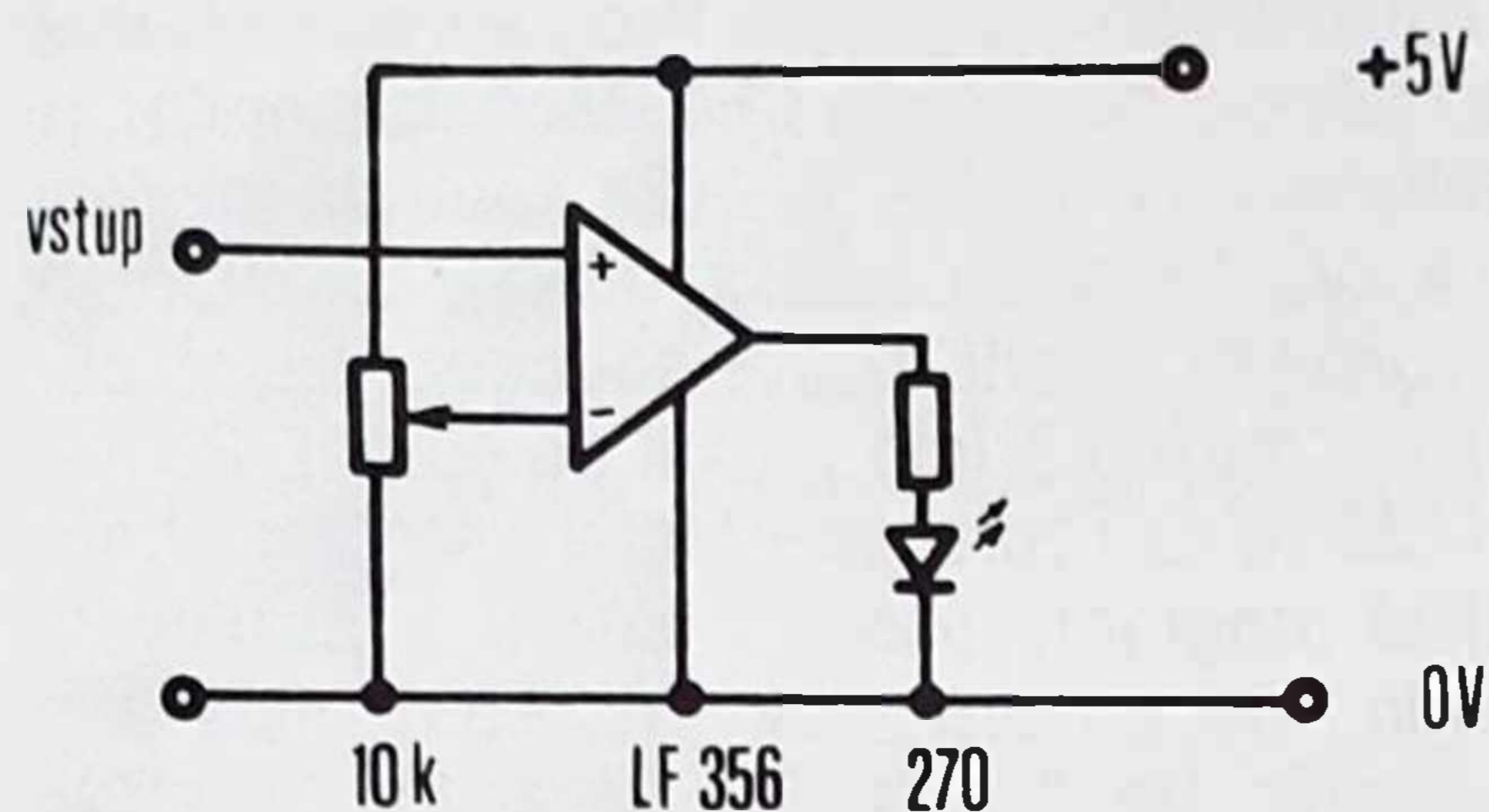
Z osciloskopických měření je však zřejmá závislost amplitudy analogového signálu na nastavení hlavy. Je-li tónová hlava optimálně nastavena, je amplituda napětí maximální. U dejustované hlavy se potom snižuje. Tím samozřejmě nechci říci, že by si každý uživatel datasetu měl pro nastavování hlavy obstarat ještě osciloskop.



Stavba přípravku

Ke stavbě přípravku doplujícího osciloskop je potřeba pár součástek.

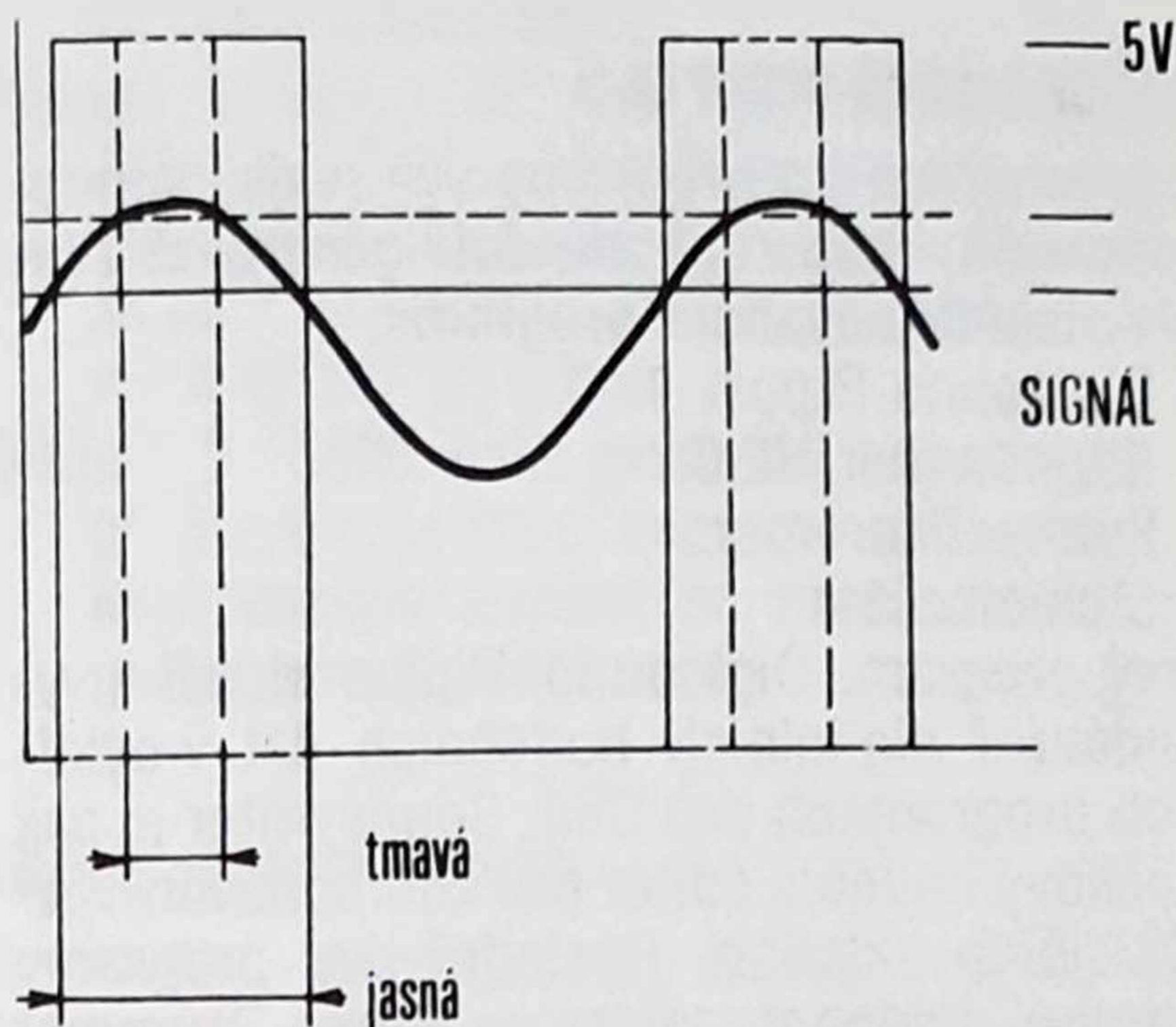
Schema uvádí obrázek.



Zapojením se dá indikovat v určitých mezích maximum signálu. Jádrem obvodu je operační zesilovač LF 356, zapojený jako komparátor. Oproti jiným operačním zesilovačům má tento výhodu ve vysokém vstupním odporu. Proto přípravek nezatěžuje měřený signál. Odporový trimr slouží k nastavení komparační úrovně.

Je-li okamžitá hodnota sinusoidy menší než nastavená hodnota, nepouští operační zesilovač do výstupu žádný proud. Je-li naopak okamžitá hodnota vyšší než komparační úroveň, přenesse se na výstup hodnota napětí 5V a LED se rozsvítí. Je-li komparační úroveň blízko maxima signálu, zkracuje se doba, po kterou dioda svítí, tedy její jas se snižuje. Naproti tomu, když se amplituda signálu zvětšuje, zvyšuje se i jas diody, neboť časový interval, po který je na výstupu napětí 5V je delší. Z toho vyplývá celý postup nastavování. Trimrem se nastavuje minimální a šroubkem tónové hlavy maximální jas diody.

Zhotovení plošného spoje ponechám plně na vás. Jeho umístění v datasetu taktéž. Na vhodném místě horního dílu krytu datasetu vyvrtejte díru pro LED. Napájecí napětí obvodu vytáhněte ze svorek motoru (pozor na polaritu!). Vstup 3 operačního zesilovače připojte stíněným kablíkem na výstup prvního, resp. druhého stupně operačního zesilovače. S úspěchem se dá využít pin 8 čtyřnásobného operačního zesilovače typu LM 324. Stínění potom spojte s minusovým pólem.



Nastavení

Před opětovným sestavením datasetu je třeba provést základní nastavení. Zapněte počítač. Do datasetu vložte kazetu s programem a stiskněte klávesu PLAY. Po zadání příkazu POKE 54272+24,15 otočte na maximum knoflík zesilovače zvuku a vyčkávejte až z reproduktoru nějaký zvuk zazní. Dále střídavě opakujte nastavení jasu diody trimrem na minimum a šroubkem tónové hlavy na maximum. Šroubek tónové hlavy je dosažitelný při stisknutí klávese PLAY pod ca 5 mm velkým otvorem v horním dílu pouzdra datasetu.

Při nastavování dodržujte následující postup:

1. Trimrem otáčejte tak dlouho, až dioda LED slabě bliká.
2. Šroubkem tónové hlavy otáčejte tak dlouho, až světelná dioda svítí naplno.

Druhý bod je nutno provádět jen tehdy, když dataset není optimálně nastaven, resp. pokud je používána kazeta zapsaná na jiném přístroji. V tom případě musí být postup opakován tak dlouho, až je dosaženo postavení, kdy tónová hlava i při minimálním odklonu zhasne. Po základním nastavení dataset zpět zakrytujte.

Pokud chcete nyní loudovat program zaznamenaný na přístroji s dejustovanou tónovou hlavou potřebujete zapnout počítač, založit kazetu, stisknout PLAY a otáčet šroubkem tak dlouho až je dosaženo maximum jasu diody.

PROGRAMOVÁ DISKETA 5

–je zaměřena na programy využívající schopnosti C64, resp. obvodu SID generovat zvuky. Disketa obsahuje programy:

Digisound Ripper
Soundwriter V2.0
Funky-Drummer
Musicmaster

Prvý program, Digisound-Ripper slouží k vyhledávání digitalních hudebních dat v ostatních programech pro C64. Soundwriter je pak špičkový hudební editor pro komponování rozsáhlejších skladeb. Poslední dva programy simulují hudební nástroje, Funky-Drummer bicí a Musicmaster klávesové.

Cena 120,- Kč

PLOTBASIC II

Program firmy Billa je určen pro spolupráci se všemi dostupnými plotry, které lze k C64 připojit, ALFI, AMAGRAF 507 (Minigraf), XY 4150. Program dokáže při kreslení otáčet libovolně souřadné osy, pracuje i v polárních souřadnicích. Libovolná velikost a sklon písma jsou samozřejmostí. Program je vybaven různými fonty s českou diakritikou pro psaní textů. Psát je možno v režimu OPEN (obdoba psacího stroje, nebo v režimu TEXT (editor). Paměť C64 je využívána jako buffer, do kterého se ukládá až 8000 kroků plotru.

Program je dodáván pro konkrétní typ plotru buď na kazetě nebo disketě.

Cena 280,- Kč

FONTEDITOR

Program využitelný ve spojení s Plottbasicem II. Slouží k vytváření vlastních znakových sad nebo ikon (viz FUN 8).

Cena 150,- Kč

CARTRIDGE MAKER CM256

CM256 je modul zasouváný do expanzního portu C64. V pamětech eprom, zasunutých do objímek, se dají uchovávat programy (uživatelské i hry). Modul se vyznačuje vysokou

rychlostí přenosu dat, kompatibilitou a snadností obsluhy.

CM obsahuje výkonný modulgenerátor jehož úlohou je přestrukturování basicovských dat či strojových programů. Programy se nejprve přetáhnou do paměti C64. Modulgenerátor je automaticky doplní několika kontrolními daty (název, startovací adresa, délka), autostartem a nakonec vyrobí datové soubory, jejichž obsah je možno přímo vypálit do eprom. Pomocí CM se dají zpracovat téměř všechny jednodílné programy psané v basicu či strojovém kódu. Do eprom jsou programy ukládány úsporně. Do čipu s větší kapacitou se může uložit více programů. Možné jsou i alternativy, kdy delší program přesahuje do dalšího čipu. V tomto případě se modulgenerátor postará o správné uložení programů do jednotlivých čipů.

Software pro řízení CM je umístěno ve zvláštní eprom (je součástí dodávky).

Význačné vlastnosti CM 256 kB:

- 4 patice pro eprom typu 2764 až 27512
- celková kapacita programů uložených v eprom činí 256 kB
- automatické rozeznání typu eprom
- vlastní directory programů v eprom abecedně seřazeno
- softwarově odpojitelný

CM je určen pro ty, kteří si chtějí do cartridge zabudovat programy podle vlastního výběru.

Cena 1200,- Kč

Pro majitele CM dále Comotronic nabízí epromer, resp. po předložení diskety s datovými soubory provede vypálení paměti.

TERMINÁTOR 2

Hitem letošních vánoc se stala souprava Terminátor 2 sestávající z počítače Commodore 64 a modulu slučujícího hru Terminátor

II (podle úspěšného filmu s A. Shwarzenegrem) a dva tzv. užitkové programy. Jedním z nich je program pro kreslení IMAGE SYSTEM (obdoba ART STUDIA), druhým potom program pro komponování a přehrávání hudby MODERN MUSIC MAKER.

TERMINÁTOR II

Děj se odehrává v blízké budoucnosti... mezi stroji a lidmi se rozvinula válka. Přes zdánlivou převahu robotů se zdá, že rebelové zvítězí. Skynet, počítač, řídicí stroje pošle na cestu v čase Terminátora, aby usmrtil Sarah Connorovou, matku vůdce rebelů.

To se mu nepodařilo.

Nezávisle byl v roce 1994 poslán jiný Terminátor, aby zaútočil na Johna Connorse, budoucího vůdce rebelů, který byl v té době dítětem. Odpůrci musí dítě chránit. Rebelům se podaří poslat osamělého bojovníka na ochranu Johna Connora. Tentokrát se setkají protivníci se stejnými zbraněmi. Bude bojovat robot proti robotovi.

Herní plán

Stupeň 1. Jako hráči řídíte Terminátora T800.

Aby John Connor mohl uniknout, musíte T1000 na čas vyřadit z boje.

Stupeň 2. Řídíte T800 a Johna na Harley-Davidsonu. Manévrujete motocykl mokrou silnicí, T1000 vás pronásleduje. Pokud narazíte do cizího objektu, ztratí T800 část energie.

Stupeň 3. V této části musíte provést na ruce T800 malou operaci. Pohybujte po rastru blikajícím kurzorem a seřaďte ve správném pořadí skládanku. Pokud úspěšně složíte v daném čase hlavolam, dostane T800 zpět celou svou energii.

Stupeň 4. Řídíte Sarah Connor. Je ozbrojena pouze dřevěnou tyčí. Musíte prchnout z labyrintu koridorů a výtahů.

Stupeň 5. T800 musí získat dostatek času, aby Sarah a John mohli výtahem prchnout k parkovišti.

Stupeň 6. Vaší úlohou je seřadit bloky s obličejem T800 tak, aby odpovídaly originálnímu obrazu. V této části hry běží čas. Po-

kud v daném čase hru 100% splníte, dosáhne T800 zpět celou svou energii.

Stupeň 7. Řídíte opět T800. Musíte si prostřílet cestu z laboratoří.

Stupeň 8. T800 řídí vůz, ve kterém prchá John a Sarah. T1000 je pronásleduje vrtulníkem. Sarah může ze zadní části vozu pálit na vrtulník. Musíte střílet na vrtulník a vyřadit jej z boje.

Stupeň 9. T800 čeká poslední boj s T1000, při kterém se musí robotovi odebrat veškerá energie a robot se musí shodit do pánve s rostavenou ocelí.

MODERN MUSIC MAKER

Stisknete-li v nabídkovém MENU modulu T2 klávesu M, otevře se před Vámi svět syntetické hudby, kterou můžete nejen poslouchat, ale také sami neuvěřitelně snadno vytvářet. Poslouží Vám k tomu řada funkcí tohoto programu, který dle Vaší volby pracuje jako ve třech volitelných stopách jako hudební nástroj, sekvencér, nebo elektronický bubeník. Velmi milým překvapením je také spolupráce programu s kazetovým záznamníkem, což je u programu obdobné kvality něco vyjimečného.

Po spuštění programu se do jeho zvukové paměti uloží první ze tří hudebních kompozic, které jsou v modulu k dispozici pro Vaše první pokusy. Ohromnou výhodou MDM je, že nemusíte pracně programovat každý tón, jeho parametry, ale můžete melodii rozdělit do opakujících se sekvencí, ty naprogramovat a pak je jen poskládat v potřebném pořadí a určit, kolikrát se mají opakovat. Pokud na to nemáte chuť a chcete jen experimentovat, můžete

použít hotové sekvence jak hudby, tak rytmičky, které jsou volitelné pod čísly \$00 – \$FF. Část z nich je již pro Vás připravena v modulu, další si můžete naprogramovat. Při tvorbě hudby ze sekvencí sestává programování jen z volby čísel sekvencí a zadání počtu opakování.

Co vše můžete s hudbou dělat?

Máte k dispozici následující funkce:

- programové editování nového kousku,
- LOAD z kazety nebo diskety,

- SAVE na kazetu nebo disketu, (přitom můžete ukládat jak jednotlivé hlasy tak celou hudbu).
- živé přehrávání a editování,
- kompilování hudby do strojáku,
- přepočítání z dekadického na hexadecimální kód, neboť hodnoty se zadávají v hex. tvaru,
- disketové DIRECTORY,
- mazání skladby,
- příkazy v DOSu

Funkce editování nového kousku umožňuje přímo hrát na klávesnici, definovat v přehledné tabulce všechny parametry tónu jako je ATTACK, DECAY, SUSTAIN, RELEASE, dále šířku pulzu, bránu, efekty jako je vibráto, kruhová modulace, filtry, zpoždění, atd.

Pro ukládání hudby máte k dispozici přes 6 kB volné paměti. Předdefinované melodie z modulu zabírají jen několik set bajtů, takže tato paměť je pro běžné použití neomezená.

Prostě, chce to mít program k dispozici, rozhodně to stojí za to!

(JK)

IMAGE SYSTEM

Program pro kreslení obrázků se ovládá joystickem v portu 2. Z klávesnice se využívají pouze klávesy

C=, CTRL a RUN/STOP.

C= slouží k aktivaci a deaktivaci MCS (Master Kontrol System), klávesa přepíná mezi obrazem, který se vytváří a panelem, ze kterého se aktivují různé funkce.

CTRL přepíná kurzor z tvaru hrotu, kterým se kreslí do tvaru šipky, kterou se potom aktivují jednotlivé funkce. V horní části obrazovky se nachází MCS s následujícími funkcemi:

DRAW - provádí volbu kreslicího nástroje, obrazce (bod, úsečka, obdélník, elipsa), popřípadě vyplnění uzavřené plochy barvou.

IMAGE - slouží k označení daného výřezu nakresleného obrázku pro jeho pozdější překopírování, zvětšení-zmenšení resp. uložení.

- BRUSH** - vytváření různých barevných vzorů.
- TEXTURE** - použijí se vzory vytvořené v předchozím paragrafu.
- TEXT** - funkce slouží pro vkládání textu do obrázků
- PRINT** - nastavení parametrů tisku
- DISK** - menu pro práci s disketami (scratch, rename, format...)

Cena originálního modulu T2 od firmy OCEAN je pouhých 720,- Kč!

(JV)

TISKÁRNA SEIKOSHA SP 180VC ZA 3300,- Kč!

Seikosha SP 180VC patří mezi nejznámější tiskárny, vybavené interfejsem COMMODORE a tím přímo připojitelné na C64. Je pravda, že dnešní tiskárny jsou výrazně rychlejší a chytřejší, ale pro běžné použití je tato tiskárna více než vyhovující. Použitá kazeta s dostupnou a levnou textilní tiskací páskou zaručuje minimální provozní náklady. Tiskárna tiskne grafiku, má NLO mód, úzké i široké písmo (2 typy), využívá běžně dostupný papír.

Pro posouzení Vám poskytneme ještě technické údaje:

- Tisková hlava** : 9 jehel
- Znakové sady** : ASCII, CBM-DIN, CBM-Standard
- Druhy písma** : PICA, ELITE, NLO, úzké písmo, široké písmo, SUBSCRIPT, SUPERSCRIPT, kurzíva
- Rychlost tisku** : 100 zn./sec
NLO - 19 zn./sec.
- Posun papíru** : Traktor i válec
- Programování** : 42 příkazů, grafický mod., ESC sekvence EPSON
- Kompatibilita** : pracuje s PRINTFOXem, FCIII, GEOSem
- Hmotnost** : 4.2 kg
- Rozměry** : 407 * 300 * 117 mm
- Napájení** : 220-240V / 20W
- Záruka** : 6 měsíců

Kazeta s textilní tiskací páskou, německý manuál, sériový kabel k propojení s C64/128 je součástí dodávky.

Z hlediska poměru cena/výkon jde o bezkonkurenční nabídku!

Tiskárny jsou okamžitě k dodání a je jich omezený počet, proto máte-li o tiskárnu zájem, neváhejte!

(JK)

C64 A VIDEO

2. pokračování

Videofox (Videofox II.)

je program, pro tvorbu titulků filmu, vlastních reklam, animaci nebo malých trikových filmů. To, co znamenají úspěšné tiskové programy Printfox a Pagefox pro hobby-redaktory, to znamená program Videofox pro fandý videa. Poskytuje komfortní a mnohostranné možnosti kombinovat na obrazovce text a grafiku spolu s mnoha zatemňovacími a skrolovacími efekty, pohybem a barvou.

Protože všechny programy pocházejí od stejného autora, je jejich vnější vzhled podobný. Pokud už máte zkušenosti s některým programem od Scantroniku (Printfox, Pagefox, Eddison...), zapracujete se na Videofox velmi rychle. Přirozeně je Videofox plně kompatibilní s ostatními produkty Scantroniku, takže svou sbírku obrázků a znakových sad můžete používat bez omezení.

Vedle vlastního videoprogramu, znakových sad a sbírky obrázků je součástí balení i program Eddison, včetně zvláštního návodu.

Pro velké nároky na paměť musel být program Videofox II rozdělen do dvou programových částí. VIDEOFOX je vlastní titulkovací generátor, který umožňuje běh titulků. Druhým programem je EDITVIDEO, který slouží pro zadávání a zpracování tabulek. Každý z obou programů se dá natáhnout a odstartovat zvlášť.

U editoru Videofoxu je menu provedeno jako dvě lišty na spodním okraji obrazovky. Přejít mezi nimi se uskutečňuje odkliknutím prvního políčka zleva (ohnutého růžku) – symbolu pro otáčení stránek.

Jednotlivé funkce se aktivují klikem příslušné ikony. Ovládání je dublováno a dané funkce je možné též aktivovat z klávesnice stiskem kombinace různých kláves. To je přirozeně zajímavé jen pro majitele joysticku, neboť odkliknutí ikony joystickem je poněkud nepohodlné. Naproti tomu ovládání myší je snadné a přináší svým majitelům zvýšený komfort obsluhy. Mimo to existuje ještě pár zřídka používaných příkazů, které mohou být zadány pouze přes klávesnici.

Předtím než přejdete k sestavení vlastního titulu nebo jiného díla, chceme nejprve všechny efekty Videofoxu podrobně popsat. Zvolte v druhé liště „GO“ nebo zadejte SHIFT-RUN. Program ukáže menu pro efekty, v němž můžete zadat parametry pro posloupnost tabulí. Pohyb oběma směry, uvnitř tohoto menu, umožňují klávesy kurzoru. Přepsáním se mění předvolené hodnoty. Funkční klávesou F1 se startu-

je běh programu s předvolenými parametry. Klávesou STOP se přerušuje vkládání údajů a program skáče zpět do editoru. Pokud jste předtím odstartovali sekvenci titulků, použijte pro přerušení a návrat do editoru klávesu RESTORE. Přejít zpět do menu efektů lze vyvolat aktivací ikony „GO“. Tímto způsobem lze vyzkoušet vliv parametrů a jejich změnu na charakter titulků.

VIDEOFOX

Zum Starten F1 drücken

Effekt Nummer (0-27):	1
Startverzögerung (Zehntel sec):	0
Wartezeit (Zehntel sec):	10
Endlos-Wiederholung (J/N):	J
Geschwindigkeit (1-4):	2
Scroll-Tafelabstand (0-25):	0

Zabývejme se blíže významem jednotlivých parametrů. V prvním řádku je uvedeno číslo efektu. Protože se efekty jen těžko popisují slovy, sami si je nejprve vyzkoušejte. Zjistíte, že pod číslem 1 je prosté řazení tabulí za sebou. Následující čísla jsou vyhrazena pro skrolovací efekty a nejvyšší čísla pro různé překrývací efekty.

Ve druhém řádku může být zadána doba zpoždění, s níž po startu titulkování zůstane obrazovka prázdná do doby, než se objeví první titulek. Tento čas může být využit např. k tomu, aby se zapnul videorekordér pro záznam či kazetový mgf. pro přehrávání úvodní hudby. Údaje se zadávají shodně jako v dalších řádcích v desetinách sekundy od 0 do 250. Údaj 30 znamená tedy zpoždění 3 s.

Dobou čekání, zadanou ve třetím řádku, se stanovuje jak dlouho má každá tabulka zůstat na obrazovce, než je nahrazena (přesunuta, skrolována, překryta) novou. Vždy dbejte toho, aby se jednotlivé titulky daly přečíst bez zbytečného spěchu. U mnoha efektů má význam zadat i nulovou dobu čekání. Příkladem mohou být pomalé skrolovací efekty, kdy se dosáhne kontinuálního průběhu titulků jako na dlouhém pásu. U efektu číslo 1, prostého přepínání tabulek, se zadáním 0 zpoždění dosáhne rychlého přepínání titulků frekvencí 10 až 20 za sekundu (podle

stupně naplnění tabulky). To postačuje pro téměř neblízkající pohyb. Příklad najdete na zdrojové disketě s VIDEOFOXEM. Jemně odstupňovaná doba zpoždění v desetinách sekundy může být také použita k tomu, aby se titulek přesně načasoval. Například na kazetě máte pro titulek mezeru určité délky. Variací doby zpoždění lze dosáhnout přesného přizpůsobení délky titulku.

Zadáte-li v řádku „Endlos-Wiederholung“ písmeno „j“ na znamení souhlasu, začne po posledním titulku nový běh všech titulků od začátku. Zkouška poslouží pro demonstrační účely. Pokud zadáte „n“, zůstane po proběhnutí všech titulků na obrazovce titulek poslední. Pokud jste majiteli mixpultu pro video, který umožňuje zapnutí a potlačení obrazu, můžete poslední titulek nechat lehce mizet. Videokontrolér C 64 tuto možnost nenabízí.

V dalším řádku zadejte rychlost, s níž probíhají efekty, resp. s níž titulky rolují přes obrazovku. 1 je přitom nejnižší, 4 nejvyšší rychlost.

Nakonec můžete ještě zadat odstup tabulek při skrolování. Zadáte-li 0, budou tabulky následovat bezprostředně za sebou. Zadáte-li naproti tomu vyšší číslo, zůstane mezi tabulkami odpovídající odstup. 12 dává např. odstup poloviny výšky obrazovky. Za určitých podmínek určí Videofox odstup sám. Jak jistě později vyzkoušíte, smí být, pro omezení videokontrolérem, obsazena jen asi čtvrtina plochy tabulky. Je-li nyní tabulka zaplněna převážně v dolní polovině, potom bude při skrolování překročena maximální přípustná plocha. Aby se tomu zamezilo, zařídí Videofox v takovém případě nezbytný odstup automaticky.

Nyní se seznámme blíže s editorem Videofoxu. Jak již víte, jsou tabulky velikosti obrazovky základem titulkování. Tabulky jsou očíslovány. Čísla jsou uvedena vlevo v liště menu. Odkliknutím šipky vpravo a vlevo od aktuálního čísla se dají tabulky prohlížet oběma směry. Po poslední tabulce se ukáže zase první a obráceně.

Chcete-li připojit k souboru tabulek další, odklikněte dvojistou šipku. Tím se prolístují ze předu dozadu všechny tabulky. Po poslední tabulce však vloží program novou, prázdnou tabulku. Podle stupně zaplnění jednotlivých tabulek jich lze do paměti C 64 umístit 15 až 22. Je-li paměť zcela zaplněna, objeví se v liště s menu hlášení „Speicherüberlauf“.

Vedle omezení paměti C64 existuje ještě další, určené videokontrolérem. Každá tabulka smí být pouze ze čtvrtiny. Přesněji řečeno z počtu $40 \cdot 24 = 960$ polí, která odpovídají velikosti znaků textové obrazovky C 64, smí nést informaci maximálně 255 z nich, zbytek musí zůstat prázdný. Pro lepší získání přehledu zadejte nejlépe příkaz C=A. Tento příkaz označí prázdná a obsazená pole rozdílnou barvou

a zvýrazní tím obsazení plochy. Stisk na klávesu STOP zvolí opět normální barvy.

Pokud se při polohování textu nebo grafiky do tabulky překročí limit 255 polí, objeví se hlášení „Tafel überfüllt“. V tomto případě nebude, stejně jako při přeplnění paměti, poslední zadání akceptováno. Chcete-li při překročení limitu vědět, kolik chybí, zadejte opět C=A. Obsazená pole, která jdou přes limit, se objeví v jiné barvě. Pak víte přesně, kolik máte odmazat, aby se tabulka dostala ještě do paměti.

Možnost přecházení mezi Editvideo a Videofox probíhá bez ztráty dat jednoduchým způsobem. Z Editvidea přijdete do Videofoxu klikem na ikonu „GO“ nebo zadáním RUN (SHIFT a RUN/STOP). Obráceně se jde klikem ikony „Edit“, resp. zadáním E. Vždy se z diskety natáhne a odstartuje nový program. Pokud žádaný program není nalezen, Videofox upozorní hlášením. Menu Videofoxu se vysvětluje samo. Vedle možnosti prolístovat tabulky prostřednictvím ikony s šipkou (z klávesnice CRSR), můžete natáhnout soubor titulků (C=L), ukládat na disketu (C=S) a vyvolat editor (E) nebo sekvencer (S). Kliknutím ikony „GO“ se dá přejít do menu s efekty.

Pokud by některý z programů havaroval, můžete jej znovu aktivovat, bez ztráty dat, zadáním SYS 2077. Protože při restartu systém C 64 přepisuje některé paměťové buňky, může se stát, že některá z tabulek bude mírně poškozena.

Branou nových možností Videofoxu jsou sekvence tj. řady příkazů jimiž je definováno pořadí tabulek, efektů a jiných parametrů. Jednotlivé příkazy uvnitř sekvence sestávají z písmenek nebo symbolů, po nichž následuje argument tvořený číslem nebo číselným rozsahem. Je lhostejno, kolik příkazů je na jednom řádku zapsáno a zda jsou odděleny nebo nalepeny na sebe.

Klávesou F1 vystoupíte ze sekvenceru do menu efektů, což ostatně, jak hned uvidíte, není pro nastavení efektů podstatné. Videofoxem je přednastavena 0, prostá sekvence. Pokud chcete titulky nechat střídát

VIDEOFOX

F1=Startmenu, F3=Preload

```
E2 G4 W0 T1 G3 T1 G2 T1 G1 T1
E20 G4 W5 F9-16 T2 F1-8 W20 T3 R
F10-16 E7 G2 W10 T4 R
W0 T5 E1 (40 T6 T5)
E19 W30 T7 E16 T8 E18 T9
L"BABY.PIC" R9-24 T10 R
E13 G3 L"BABY.PIC",9-16 T10 R
E14 L"BABY.PIC",17-25 T2 R E18 T11
L"BABY.PIC" T11
E22 L"SCHLUMPF" T12 R20-25 T12
E12 T13 T14 R E23 T15 T16
```

jednotlivými efekty, zadejte jednoduše jejich číslo. Jednotlivé příkazy rozebereme. Nejdůležitějším je příkaz „T“. Něco vysvětlovat je zbytečné, rozumně byl tento příkaz zvolen pro tabulku, která se má objevit na obrazovce.

S jakým efektem a jakou rychlostí se tabulka objeví a jak dlouho zůstane stát se zadá dopředu příkazy „E“ (Effekt – efekt), „G“ (Geschwindigkeit – rychlost) a „W“ (Wartezeit – doba čekání). Jedná se o stejné parametry, které mohou být zadány v menu efektů, kdy platí pro celý průběh, zatímco v sekvenci mohou být libovolně předefinovány. Efektové menu potřebujete potom jen pro zadání zpoždění na startu a pro určení, zda mají titulky probíhat v nekonečné smyčce. Také ostatní parametry mohou být nastaveny jako předtím v menu efektů, pokud nejsou definovány v sekvencích.

Za pozornost stojí, že E,G a W platí vždy s následujícím příkazem T. Např.

E5 G3 T1 E7 T2

V překladu to znamená, že tabulka 1 se objeví s efektem 5 a tabulka 2 s efektem 7. Rychlost 3 platí pro obě tabulky. Pro malé animace v rámci titulků je nezbytné nechat určité pořadí obrázků proběhnout vícekrát za sebou. To umožní závorky, představující určitý druh smyčky FOR – NEXT. Vše co je uvedeno v závorce se opakuje. Počet opakování je dán číslem za první závorkou. Např.

E 1 (30 T4 T5 T6 T5).

Názvy jednotlivých efektů jsou obtížně přeložitelné do češtiny. Efekt 1 je obzvláště rychlý a tím vhodný pro animaci. Efekty 2 až 4 jsou nebarevné, barva zůstane podle první tabulky. Efekty 5 až 27 jsou vhodné pro grafiku s příkazem „F“.

PROGRAMUJEME DEMO

(pokračování)

V posledním článku PROGRAMUJEME DEMO jsme si popsali princip práce s přerušením IRQ. Při vlastním programování demo to však vypadá trochu jinak. V přerušení IRQ proběhne všechno, co později obdivujeme na obrazovce. Jednotlivé rutiny efektů jsou ze smyčky IRQ volány příkazem JSR jednotlivě volány a prováděny. Tak je vyvolán dojem, že grafické efekty a hudba běží současně, ačkoliv se nezbytně jedná o samostatné procedury prováděné postupně za sebou (Single-Step-Procedures). Pokud například voláme příkazem JSR PLAYMUSIC rutinu pro přehrávání hudby, pak se můžeme z této rutiny vrátit do smyčky až tehdy, když jsou jednotlivé noty zahrány, jinak jinak musí při vyvolání již aktuální nota hrát a při volání ji pomocí RTS přerušíme. Pak přijde na řadu například rastrová rutina atd. Díky tomuto sdílení času, které je nazýváno Timesharing nebo také Multi-Tasking a rychlosti našeho procesoru MOS6510 vzniká dojem současného běhu více programů. A opět poněkud komplikovanější je to při rastrovém IRQ. Ale žádnou paniku, i to se zdá komplikovanější, než to ve skutečném efektu je. Princip je jednoduchý (viz obrázky).

Videočip VIC je zaměstnán hlavně stavbou obrazu. Abychom měli na obrazovce neblíkající, čistý obraz, musí VIC obraz vytvářet celkem 25x za vteřinu. Každý jednotlivý obrázek sestává z 625 řádků (dva poloobrazy z 312.5 řádků), které jsou na obrazovce vykresleny elektronovým paprskem. To probíhá téměř rychlostí světla. Tento rastrový paprsek je videokontrolérem VIC rozdělen na rastrové řádky

a sloupce, takže každý bod obrazu je možno určit dvěma souřadnicemi. Počítač C64 je se svým pracovním kmitočtem 0.98 MHz tak pomalý, že prakticky funguje jen dotaz na řádek. Pomocí techniky Timingu je možno i sloupec přesně definovat, ale vyžaduje to od programátora spoustu trpělivosti a doporučuje se tedy jen zkušeným programátorům s nejpevnějšími nervy. Pomocí rastrového IRQ je možno zobrazovat současně grafiku i text zároveň. Jak již bylo u grafiky vysvětlováno, probíhá to tak, že se hlídá určitý rastrový řádek na obrazovce a přepíná se mezi textem a grafickým zobrazením. Realizace je celkem jednoduchá, budeme se jí věnovat později.

A ZDE MÁ HRÁT HUDBA

K profesionálnímu DEMO programu patří samozřejmě podmalování hudbou. Mnozí se teď zarazí, protože toho o SIDu, který vytváří v C64 zvuky, málo vědí. Ale ani to není zapotřebí. Téměř všechny již zkomponované hudební kousky mají již zabudovanou rutinu Single-Step-Player a tu můžeme po zabudování do vlastního programu jednoduše volat příkazem JSR. Hudba pak hraje v přerušení, aniž byste museli v SIDu naprogramovat jediný bit. Normálně je zapotřebí hudbu nejdříve inicializovat (nastavit ukazatele) a potom ji lze hrát. Ze známých hudebních programů zde uvedeme příklady:

Future-Composer : \$1000 (INIT), \$1006 (PLAY)

Future-Composer : \$1800 (INIT), \$1806 (PLAY)

JCH-Player : \$1000 (INIT), \$1003 (PLAY)

MAC-Player : \$1048 (INIT), \$1021 (PLAY)
 RoMuzak : \$8000 (INIT), \$8003 (PLAY)

Také jejich výstavbou se budeme zabývat později podrobněji.

Ještě jednu malou poznámku na okraj pro programátory, kteří již za sebou mají první úspěšné pokusy o programování dema. Pokud se vaše demo vyznačuje občasným blikáním nebo nedefinovanými blikajícími pruhy, potřebuje některá z použitých rutin více cyklů, než je v přerušení k dispozici. Omezíte-li například rozsah jednoho skrolování na maximálně 4 rastrové řádky, dojde zcela určitě k havárii. Proto na začátku přiřadíte raději rutinám více času a minimalizujete jej později při testování omezováním rastrového rozsahu krok po kroku.

DEKLARACE PROMĚNNÝCH

Přestože řada programátorů se málo stará o přehlednost svých programů, doporučujeme vám, abyste věnovali přehlednosti svých programů velkou pozornost. Jen tak zachováte možnost úprav programů poté, co jste je několik týdnů nebo měsíců neviděli. Jen tak se naučíte tvořit rozsáhlejší projekty bez zbytečných hodin zkoumání, co jste vlastně před časem vytvořili a jak to pracuje. Proto se například doporučuje na začátku podprogramů deklarovat proměnné. Viz listing 1.

LISTING 1

```

;----- Deklarace proměnných -----
;
;
IRQLOW      = $0314 ;VEKTOR IRQ, LOW
              BAJT
IRQHIGH     = $0315 ;      HIGH BAJT
ODLIRQ      = SEA31 ;STARA RUTINA IRQ

INITMUSIC   = $1000 ;INICIACE HUDBY
PLAYMUSIC   = $1003 ;HRANI HUDBY

FLASHCOUNT= $FB   ;CITAC 1 A 2
FLASHCOUNT= $FC   ;PRO EFEKTY FLASH

SCRHELP     = $03BF ;POMOCNY REGISTR
SCROLLREG   = $D016 ;PRO SCROLL.REGISTR

CHARSET     = $D018 ;DEFINOVANI ZN.SADY
RASTER      = $D012 ;POZICE RASTR.
              PAPERSKU
YSCROLL     = $D011 ;REGISTR Y-SCROLL
IMR         = $D01A ;REGISTR MASKY IRQ
  
```

Ale pozor! Jak již bylo uvedeno, spotřebují dlouhé názvy také mnoho paměti. Proto se doporučuje kompromis mezi délkou a srozumitelností. Jak jsme uvedli v minulé části, poté, co se definují proměnné, je iniciována nová rutina IRQ.

INICIACE IRQ

Při iniciaci je nutno dbát nejdůležitějších věcí. Tak například musí být tato rutina vždy „zarámována“ do příkazů SEI a CLI. To má následující důvod. Jak již pravděpodobně víte, maže C64 vždy po 1/60 sekundy přerušení a nechává blikat kurzor, dělá dotaz na klávesnici, atd. rutina, která se o práci v přerušení stará, je v paměti umístěna od adresy SEA31. Ale rutina IRQ není naštěstí naskakována přímo, ale přes vektory na adresách \$0314/\$0315, které lze libovolně měnit. Aby však počítač neskočil náhodou na napůl přepsanou adresu IRQ a nedošlo tak k zamrznutí počítače, musí se zabránit přerušení příkazem SEI a nakonec přerušení uvolnit příkazem CLI.

LISTING 2

```

;-----PRIPRAVA IRQ-----
;
;
SEI          ; ZABRANENI PRERUSENI

JSR INIT    ; INICIACE

LDA #<START ; NATAZENI LOW BAJTU
STA IRQLOW  ; A ULOZENI
LDA #>START ; NATAZENI HIGH BAJTU
STA IRQHIG  ; A ULOZENI

LDA YSCROLL ; SPRAVNE NASTAVENI
AND #$7F   ; REGISTRU
STA YSCROLL ; Y-SCROLL

LDA #$7F   ; NASTAVENI
STA $DC0D  ; TIMERU

LDA #$01   ; STANOVENI
STA IMR    ; RASTROVEHO IRQ

LDA #$00   ; INICIACE
JSR INITMUSIC ; HUDBY
CLI        ; UVOLNENI IRQ
JMP *      ; NEKONECNA SMYCKA
  
```

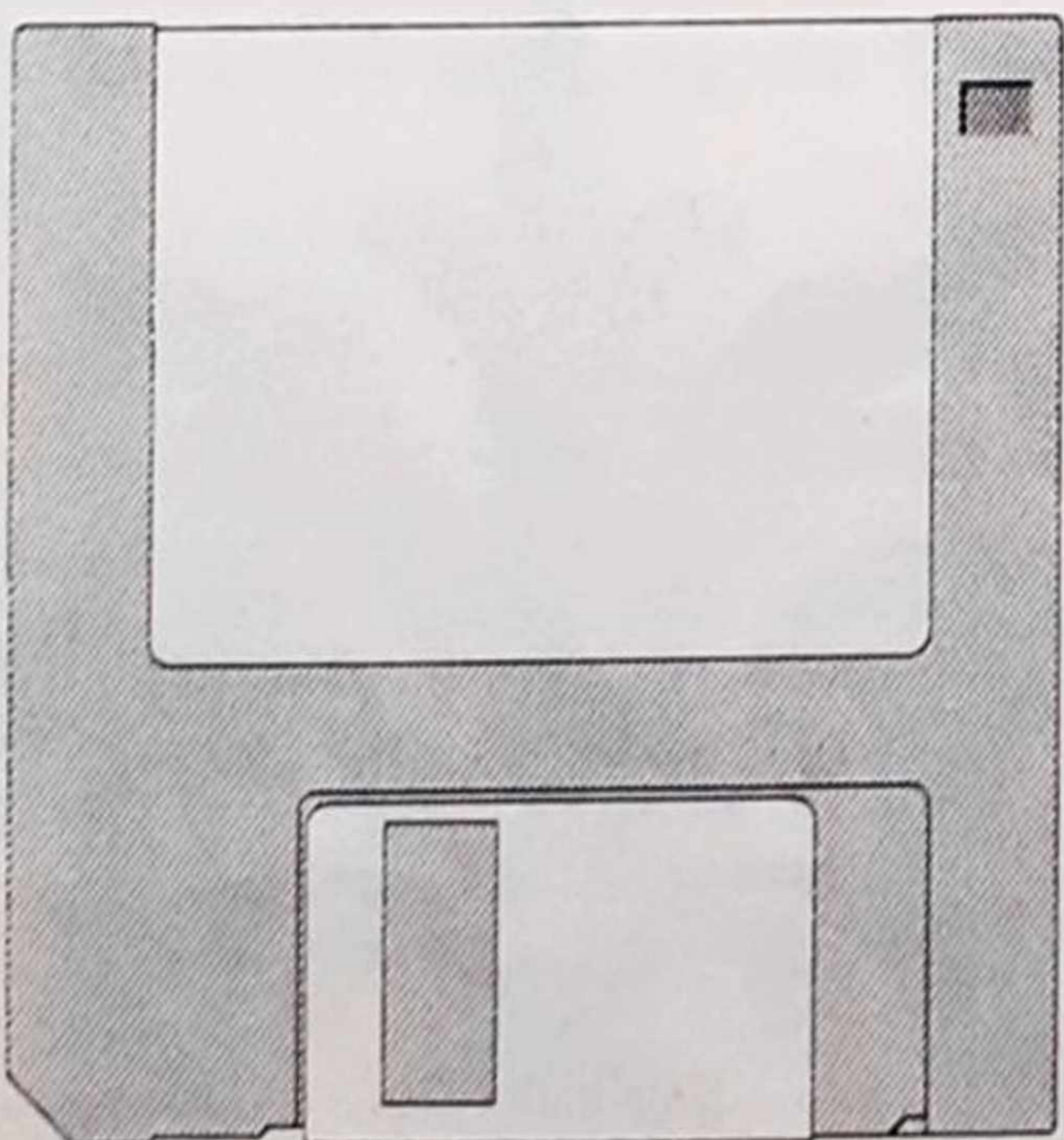
Tak nějak to celé může vypadat. Registr Y-scroll se nastavuje normálně na standardní hodnotu. Také na časovač CIA nesmíme nikdy zapomenout, zvláště když potřebujete jinou hodnotu. Potom nastavíme

vektor IRQ na naši rutinu rastrového IRQ (adresa \$D01A) a inicializujeme případný hudební doprovod. Tato iniciace jde celkem jednoduše. Normálně začíná hudební rutina na kulaté adrese (například \$1000, \$1800 nebo \$8000 atd.

V LOW-bajtu je tedy vždy hodnota \$00. Podle Playeru se může v jednom hudebním kousku nacházet současně více kompozic. Pro volbu některé z nich je třeba uložit do akumulátoru (občas do registru X) číslo kousku, který chceme přehrát (při jedné melodii je to obvykle číslo \$00 nebo \$01 a potom se naskočí iniciační rutina příkazem JSR \$xx00.

JMP TRIK

Příkaz JMP * na konci listingu 2 pro iniciaci IRQ je specialitou editoru Turbo-Ass. Hvězdička označuje programový čítač, to znamená, že smyčka je nekonečná. Pokud se nyní díváte, pomyslete na to, že náš program běží stále dále, ačkoliv je pomocí přerušení každou 1/60 sekundy přerušen a skáče do naší (ještě nenaprogramované) rutiny. Také rutina INIT nesmí být opomenuta (listing 3). Tak nebo nějak podobně byste měli ve vašem programu nastavit proměnné, vymazat obrazovku, iniciovat sprajty, zapnout znakovou sadu atd. Tato rutina musí být umístěna vždy na konci vašeho strojového kódu. To má jeden důvod. Vezměte si, že naše z hlavního programu volané subrutiny leží příliš daleko od volání. Přirozeně potřebuje procesor pro provedení JSR nějaký čas. Čím větší je při skoku překračovaný rozsah RAM, tím více času procesor potřebuje. Na začátku, při přípravě IRQ to není tak divoké, ale v pozdějším hlavním programu je to katastrofa. Protože programátoři demo programů trpí chronickým nedostatkem procesorového času, musí se na každém rohu a konci šetřit takty, také při maličkostech.



LISTING 3

```

INIT  LDA #C7          ; NASTAVENI
      STA SCRHELP      ; POMOCNEHO
                          ; REGISTRU

      LDA #00          ; INICIACE
      STA FLASHCOUNT ; CITACE
      STA FLASHCOUNT2;
      STA D020         ; OBRAZOVKA
      STA D021         ; A RAMECEK
      LDA #00          ; CERNE
      STA S0286        ; BAREVNA RAM
                          ; VYMAZAT
                          ; OBRAZOVKU

      LDX #00          ; „DEMO TEST
INIT1  LDA SCREEN1,X  ; VETSI ZNAKOVA
      STA S0454,X     ; SADA"
      INX              ; ZAPIS
      CPX #SEC        ; NA OBRAZOVKU
      BNE INIT
      LDX #77         ; „MENSI
INIT3  LDA SCREEN2,X  ; ZNAKOVA SADA"
      STA S0608,X     ; PSANI
      DEX              ; NA
      BPL INIT3       ; OBRAZOVKU

      LDX #27         ; ZAPIS
INIT4  LDA FLASHTAB2,X; DO BAREVNE
      STA D820         ; RAM
      DEX              ; (PRO BAREVNE
      BPL INIT4       ; SKROLOVANI)

      LDA #C7          ; NASTAVENI
      STA SCROLLREG   ; SKROLOVACIHO
                          ; REGISTRU

      LDA #1C         ; ZAPNUTI
      STA CHARSET     ; ZNAKOVE SADY
      RTS              ; KONEC

```

HLAVNÍ SMYČKA

Tak máme všechny definice, iniciace a deklarace za sebou a nyní přijde na řadu samotná hlavní smyčka. Z ní se volají všechny rutiny efektů, které později budou sloužit k okouzlení vašich kolegů. Hlavní smyčka může pochopitelně vypadat úplně jinak, než je ta naše, dále uvedená, což pochopitelně znamená, že je špatná. Každý programátor má přece svůj vlastní styl.

LISTING 4

```

:-----HLAVNI SMYCKA IRQ-----
:
START LDAYSCROLL      : CEKANI AZ JE
  STA START          : DOSAZEN
                    : DOLNI
                    : OKRAJ
                    : PREPNUTI
  LDA#S1A            : ZNAKOVE SADY
  STA CHARSET        : NASTAVENI
  LDA#SC8             : SKROL RE-
  STA SCROLLREG      : GISTRU

                    : HUDBA
  JSR PLAYMUSIC

                    : CEKANI
  LDA#S38             : NA RASTROVY
ST1  CMP RASTER      : RADEK S38
     BNE ST1

                    : RASTROVE
                    : BARVY
  JSR RASTERSHOW

                    : SKROLOVANI
                    : BAREV
  JSR SCROLL

                    : CEKANI
  LDA#S8C             : NA RASTROVY
ST2  CMP RASTER      : RADEK S8C
     BNE ST2

                    : ZAPNUTI MALE
                    : ZNAKOVE SADY
  LDA#S1C             :
  STA CHARSET

                    : 3 RADKY
                    : BLIKANI
  JSR FLASH

                    : CEKANI
  LDA#S88             : NA RASTROVY
ST3  CMP RASTER      : RADEK S88
     BNE ST3

                    : SKROLOVACI
                    : RUTINA
  JSR CHARSCROLL

                    : BAREVNE
                    : SKROLOVANI
  JSR CHARFLASH

                    : STARE IRQ
  JMP OLDIRQ

```

Ale pozor! Bez rutin efektů je naše rutina jen omezeně schopná běhu. Proto také nefunguje přepnutí znakové sady (na začátku) Ve srovnání se zbytkem je hlavní smyčka vystavěna podle celkem jednoduchého schéma. Jednoduše ve smyčce nastavíme rastrový paprsek a provedeme až k další smyčce odpovídající rutinu. Podle efektu se k tomu potřebuje větší nebo menší rastrový rozsah. Tak například hudba potřebuje jen \$1E rastrových řádků (\$38 - \$1A = \$1E), Scroller a Raster - Bar - Shover (rutina pro zobrazování rastrových pásů) (od návěští ST1) naproti tomu potřebují trojnásobek (\$8C-\$38=\$54). Nenechte se rutinami zmýlit, ty tu vůbec nejsou uvedeny. Všechno potřebuje svůj čas. Pokud jste pochopili princip přerušení v rastrových řádcích, jste připraveni udělat velký krok dále. Představená metoda není ta nejčastější, naproti tomu je velmi rychlá a především dostatečně průhledná a tím se také snadněji dále rozvíjí. V dalším pokračování se budeme zabývat podrobněji jednotlivými efekty. Dozvíte se, jak se programují rastrové pásy, jemné skrolování a mnoho dalšího.

(JK)



JARNÍ NABÍDKA FIRMY **3C** spol. s r.o.

HARDWARE

Commodore 64 II - 3 950 Kč
disketová jednotka VC 1541 II - 2 770 Kč

C64 TERMINÁTOR 2 - 4 270 Kč
magnetofon - 790 Kč
tiskárna C64 VC1351+SW - 490 Kč

PROGRAMY

PROTEXT 1.0 (kaz./disk) - 195 Kč

návodní český textový editor, B typů písma, 2 druhy písma, pro všechny běžné tiskárny, převod do formátu T602 atd.

PODNIKATEL 2.0 (disk) - 490 Kč

peněžní deník, faktury, databáze, sklad, kniha jízd, dodací list

LANDI 2.0 Plus -A nebo N (kaz./disk) - 240 Kč

výukové slovníky angličtiny nebo němčiny, možnost doplnění 10 000 hesel

CSL 3.0 (kaz./disk) - 195 Kč

komplexní výuka české gramatiky pro všechny ročníky

BABYMAT komplet (kaz./disk) - 350 Kč

komplexní výuka matematiky pro 1-7 tř. ZŠ

hry na disketách (65 disků) - 49 Kč/disk a hry na kazetách (20 kaz.) - 119 Kč/disk

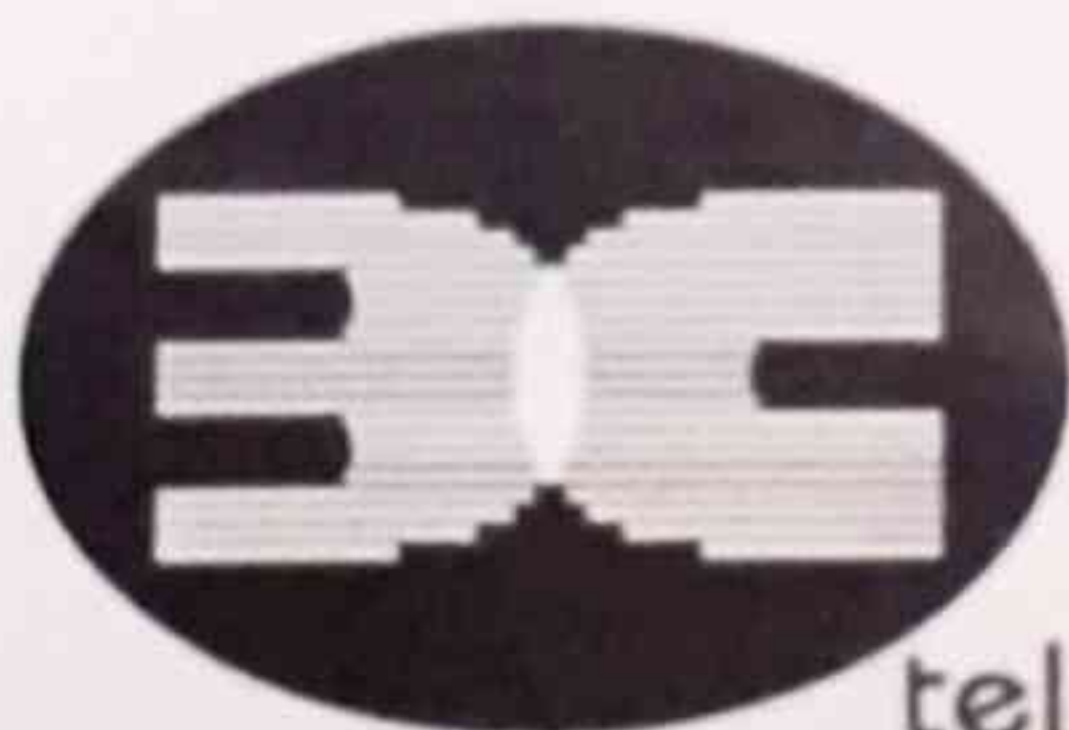
LITERATURA

Commodore 64 man.	95 Kč	Geos 1.2,1.3	49 Kč
Commodore 64 od A do Z	150 Kč	Geos 2.0	99 Kč
Paměť C64	25 Kč	Superbáze 64	99 Kč
Floppy 1541 - manuál	88 Kč	Programujeme C64	59 Kč
Floppy kurz	35 Kč	Grafika na C64	49 Kč
Kernal	29 Kč	Basic 64	49 Kč
MPS 1230 - manuál	55 Kč	Giga Cad	35 Kč
Textové editory na C64	89 Kč	Printfox	9 Kč

HLEDÁME DEALERY A ZASTOUPENÍ
pro SLOVENSKOU REPUBLIKU

HLEDÁME DEALERY A ZASTOUPENÍ
pro ČESKOU REPUBLIKU

*Dále nabízíme velké množství literatury,
doplňků, výukových programů a her
pro počítače C64 A AMIGA 500-1200*



3C spol. s r.o.

Sokolovská 76
186 00 PRAHA 8

tel: 02/ 24 21 84 72

PODROBNÝ KATALOG
nov. sw literatury a doplňků
pro C64 a AMIGY zašleme





Vydává Comotronic klub Šumperk