

# AMATÉRSKÝ PROGRAMÁTOR



ČÍSLO 3 - ROČNÍK 4 - KVĚTEN 1993

RAMDISK  
PRO "DG"

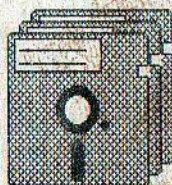
KURZ  
PROGRAMOVÁNÍ  
STROJOVÉHO  
JAZYKA

D40 A  
STROJOVÝ  
KÓD

SYSTÉMOVÉ  
PROMĚNNÉ

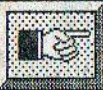


ANIMACE  
FIGUR  
V BASICU

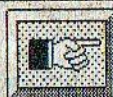


NA ZX SPECTRU  
PODOBŇĚ  
JAKO NA "PC"

PROGRAMY  
VLADIMÍRA  
VOJTY



"HITSOFT"  
ZAHAJUJE



Zpoždění ve vydání tohoto čísla zavinila firma SLOFPAP, Komárňanská 24, BRATISLAVA, která nebyla schopna během měsíce a půl(!) odeslat mnou včas zaplacenou kapalinu na výrobu ofsetových desek pro naši tiskárnu.

# Z redakce

Tak jsme se dočkali již třetího *ápička* v tomto roce.

Velikou radost mi působí, že přes prognózy jednoho nejmenovaného *profíka*, že Spectrum a tedy i Didaktik již mají na kahánku, hlásí se (i když jen spíše) noví odběratelé AP a ti stávající začínají být aktivní. Chtěl bych proto poděkovat těm, kteří mají velkou zásluhu na tom, že AP žije. Jistě jste si už všimli v poslední době často se opakujících jmen na jeho stránkách, jako je Pavel Macek, Vladimír Vojta, Eugen H. Bečz, Jiří Brossmann, Pavel Rak či Tibor Bartoš. Jsou všechno také amatéři a pokud na něco přijdou nenechávají si to pro sebe ani neuvažují o tom jak svůj objev zpeněžit, ale mají radost, že se o to mohou podělit s ostatními.

Ti totiž vydatně zásobují AP svými příspěvky *bez nároku na odměnu!* To zdůrazňuji! Jako soukromý vydavatel, který není nikým sponzorován (ne že bych o to nestál) si totiž nemohu bohužel dovolit honorovat autory přispívající do AP. Jedinou odměnou budiž jim zatím alespoň sláva a úcta ostatních uživatelů Spectra a Didaktika, kterým tak svými zkušenostmi vydatně pomáhají.

Chci také poděkovat i těm čtenářům, kteří ve svých dopisech nešetří chválou na lepší obsah AP v poslední době. Jak už jsem napsal, je to zásluha těch jmenovaných výše.

## ANONYMNÍ STÍŽNOSTI ŘEŠIT NEMOHU

Již podruhé jsem obdržel dopis od jednoho čtenáře z Prahy (soudě podle razítka na obálce), který bohužel ani jednou nevedl

svou adresu. Protože v obou dopisech si stěžuje na činnost některých známých firem a ve druhém dopise mě dokonce žádá abych varoval své čtenáře před využíváním jejich služeb, je důvod, že nevedl svou adresu, že mu odpovídám touto cestou.

"Vážený pane, časopis AP je určen pro amatérské programátory a je tedy jeho hlavním posláním poskytovat jim cenné informace pro jejich práci na počítači. Nezlobte se proto na mne, že se Vaším ztížnostem nebudu nijak zvlášť věnovat, neboť na vyřizování podobných stížností jsou tu jiné instance a nikoliv redakce AP.

## SECOM JE JIŽ POUZE TISKÁRNA

Z důvodů přeregistrace již firma SECOM v původní podobě již vlastně neexistuje. Zůstal pouze název SECOM, který je již názvem jen tiskárny, která AP tiskne. Z řádků uvedených ve vedlejším sloupci jste jistě pochopili, že redakci dnes již tvoří vlastně dopisovatelé a nemohu již tedy uspokojit Vaše dotazy, co a jak řešit v svých programech. Pokud se vyskytne nalehová potřeba pomoci Vámi, předám Váš dopis některému z našich dopisovatelů (redaktorů) a pak již záleží na nich zda a co Vám odpovědí.

Pokud se tedy objeví nějaké nové produkty, nebudou již pod hlavičkou SECOM, ale buďto jako produkt AP nebo přímo označené jménem autora.

## "AP" NA DISKETĚ

Pokud se podaří náš záměr, který připravujeme společně s Davidem Wenzelern, bude *ápičko* vycházet čtvrtletně také jako disketová verze. Přesněji řečeno to vlastně bude taková BANKA Basicových a strojních programů, které byly zveřejněny na stránkách AP za uplynulé období. Bude mít svůj RUN a MENU ze kterého si můžete zvolit jak

informace o zvoleném programu tak třeba i jeho předvedení jako DEMO a nebo jeho vložení na váš disk.

Pokud tedy i vy máte něco zajímavého o čem se domníváte, že by to pomohlo i ostatním, neváhejte, nestyďte se a svůj výtvar nám pošlete jako hotový a spustitelný program nebo rutinu. Může se pak objevit také na této disketě.

Disketa bude stát asi 50,- Kč a z utržených peněz bude podle rozsáhlosti a počtu příspěvků poskytován alespoň nějaký honorář autorům programů na této disketě. Mnozí z nich jsou totiž studenti a tak jim každá kačka jistě přijde vhod. A pro Vás z těchto 50,- Kč za plnou disketu programů nebude tak moc.

#### NOVÁ GRAFICKÁ ÚPRAVA "AP"

Až začnete v AP skutečně číst, určitě Vám neujde, že má novou grafickou úpravu. Je to tím, že jsem se konečně dočkal disketové verze Text Machine. Pokud si tento program zakoupíte se vším co k němu patří, tedy jako komplet DTP-MACHINE za 739,- Kč zjistíte velice brzo, že tato částka, která se Vám nejdříve bude zdát moc vysoká, je skutečně přiměřená. Získáte totiž něco, co prozatím znáte jen od kolegů na PC. Nebudu se zde nyní zrnitovat o všech přednostech a vymoženostech včetně neuvěřitelně kvalitně vyřešené obsluhy programů a plného využití disketové jednotky (o tom píšu na jiné straně AP).

Přesto, se však bohužel najde vždy něco, co dobrou pověst programu kazí. Tentokrát to není autor programu, ale jeho distributor fa ULTRASOFT. Chtěl bych rád věřit tomu, že je to jen případ ojedinělý, totiž ten, který se týká našeho čtenáře, jemuž fa Ultrasoft nejprve poslala vadnou kopii a teprve po několikaletých urgencích, kdy už uživatel ztrácel trpělivost a

chtěl programy vrátit, mu teprve vyhověla. On však již mezi tím skutečně ztratil důvěru a poslal programy k nám do redakce k otestování, zda skutečně má již funkční programy. To se na konec potvrdilo. Domnívám se, že při takové ceně za produkt těchto kvalit by měl jeho distributor postupovat obezřetněji ke svým zákazníkům.

Protože bych se chtěl ještě i v dalších číslech tohoto kompletu DTPMACHINE vracet a psát o něm podrobněji, budu rád, když pošlete své dosavadní zkušenosti s programy a nebo dotazy, co vám není jasné nebo co si neurníte nainstalovat či nadefinovat. Je to totiž již program na vysoké úrovni (není to jako jen nějaký R-TEXT nebo DESKTOP) a má jen v textu přes padesát funkcí, proto je nutné se s programem dobře seznámit. Já pak na základě Vašich připomínek a dotazů v veřejném vše co vím, neboť s tímto programem již pracuji od jeho vzniku (tehdy ještě pro magnetofon).

#### SECUM PUBLIC DOMAIN

V prvním čísle jsem slíbil, že SPD bude obnoveno opět od měsíce března t.r. Doposud jsem však nezveřejnil žádný seznam programů a disket. Za to se Vám všem moc omlouvám (prostě jsem to nestihl) a činím tak až v tomto čísle. Systém zůstává zatím stejný, za 20,-Kč vložených do obálky společně s Vaší naformátovanou disketou obdržíte vždy jednu disketu. Horší to prozatím bude se zájemci ze Slovenska, kde ještě nemám smluvně zajištěnou pobočku (pro Vlada je prý 30% málo on chce víc). Ale už se nám nabídla fa BONO s r.o. z Košic a pan Ing. Makovec a Trenčína. Pokud se nějak dohodnem, budu Vás informovat.

-Petr Černý-

# JAK CHRÁNIT PROGRAMY



Tyto nesmysly volně navazují (jen aby...) na článek pana Pavla Raka z AP 10/92.

## JAK JINAK ZABRÁNIT SNAPSHOTU??

Jednou, když jsem lámal ochranu proti kopírování Art Studia od fy Perfekt (dnes Perpetum), tak mě zde, mimo jiné, zaujala jedna věc; když jsem zmátl SNAP tak se nic nestalo. Podíval jsem se tedy programu na kobytku. Díky neopatrnosti programátora, který ARTstudio upravoval pro D40 a díky mé genialitě (POZOR! smrdí tady sarnochvála...) jsem přišel na to, že stačí definovat jako hlavní pracovní drajv příkazem MOVE, třeba C nebo D a SNAP nebude fungovat. Vždy, když bylo u ARTstudia třeba pracovat s jednotkou, tak se rychle přeplo na A, provedla se operace a zase se definoval D jako hlavní drajv



Tak to by tedy byl SNAP, ale je tady další věc, kterou jsem objevil (taky při lámání ochrany) u George K. v jeho (??? jak on tvrdí...) hře HEROES92. Tam jsem si prohlížel Basic, který vypadala asi takto:

```
0 POKE #CODE "RUN",CODE "0"  
1 POKE 23659,0: POKE 0,0: POKE 0  
,0: IF COPY THEN CAT & DOG  
0 LOAD *"heroes cd"&CODE: CLS: P  
ASUE NOT PI: RANDOMIZE USER  
THE RETURN OF THE LIVING DEATH  
Welcome to the 1st round
```

Slo zde pouze o to, že druhý řádek s číslem nula byl je trik

jak někoho zmást (inu, někdo si potrpí na hovadinky...). Vlastně nebyl k ničemu, bral se z něj název "heroes cd" a proto to šlape i když tento řádek smažete.

Zjistit zde startovací adresu strojáku, který tedy nahrával blok CODE z diskety, nebylo těžké. Ty dva pouky na řádku jedna, jsou ve skutečnosti jiné (k čemu by taky byl POKE 0,0). Byly tam POKE 23613,nn a POKE 23614,nn kde nn byly hodnoty, které po složení daly onu adresu startovací rutiny. **A POZOR!** V system. proměnných 23613 a 23614 je dvoubajtové rozložena adresa, kam se má skočit v případě, že dojde k chybě, no a následující IF COPY THEN... by vyvolal hlášení "Nonsense in Basic" a tím se vlastně "nenápadně" skočilo na adresu obslužné rutiny a další už ovládal strojový kód.

A do třetice všeho ochranného tady mám ještě jeden figl, již tradičně z mojí hlavy.

Příklad: máme řádek 10 PRINT "AHOJ KARLE":REM to nic:RANDOMIZE ZE USR 0

Co byste řekli, kdyby třeba příkaz PRINT zmizel tak, že by nebyl vidět, ale jinak by se provedl... Je to velice jednoduché. Před příkaz PRINT si dáme mezeru a pomocí monitoru tam potom dáme místo této mezery (kód 32) kód 14. Co se stane? Systém Spectra to trochu zmáte. Hodnota 14 v systému znamená, že následuje 5 bajtů, které se nezobrazí, protože jsou považovány za údaje o číslu, které mělo následovat před kódem 14. O.K. Systém následujících 5 bajtů nezobrazí (ovšem když program běží, tak se kód 14 ignoruje).

Kdo to nepochopil, tak má smůlu a kdo to pochopil, tak je borec.

Zatím čau, a chraňte si dobře svoje programy (a to hlavně předemnou...)



-agent W24C-

# KLUB AMATÉRSKÝCH TVŮRČŮ SOFTWARE PRO MALÉ OSMIBITOVÉ POČÍTAČE ZX8 DELTA DIDAKTIK

Je tu pro všechny uživatele těchto počítačů, kteří mají vážný zájem programovat nebo pracovat s grafikou či se zvuky a hudbou.

Klub je tu, abyste jeden o druhém věděli a mohli vzájemně spolupracovat a pomáhat si.



## REGISTRACE DO KLUBU

Uživatel, který má zájem být v klubu registrován, sdělí písemně na kontaktní adresu klubu:

1. své jméno a příjmení
2. přesnou adresu
3. své zaměření (grafika, hudba, programování)
4. typ počítače na kterém pracuje
5. jaké periferie používá (typ disk. mechaniky, tiskárny, mgf aj.)

Na základě těchto informací pak bude založena velká registrační karta s přiděleným číslem, jejíž opis, malou reg. kartu, dostane uživatel. Tím vznikne klubový adresář různých zaměřených uživatelů a tedy základna pro kontaktní síť.

Klub nehospodář se žádnou finanční částkou, tudíž neplatí se



žádný registrační poplatek, pouze je nutné, aby každý kdo se písemně obrací na kontaktní adresu klubu (i registrovaní) vložil do obálky k dopisu také poštovní známku v hodnotě 3,- Kč. Jinak by se odpovědi nemohl dočkat. Pořádek dělá přátele.

Napište nám, co Vás na počítači nejvíce baví (grafika, hudba, programování), jaký počítač vlastníte a s jakým příslušenstvím (tiskárna, disketová mechanika, mgf. ap.)



KLUB AMTERSKÝCH TVURCU

Pokud jste již na svém počítači něco vytvořili **pochlubte se!**

Pokud si s něčím nevíte rady **postězte si!**

Hlavní motto našeho klubu:  
**Důležité je, abychom jeden o druhém věděli.**

# VYHLAŠOVÁNÍ KLUBOVÉ SOUTĚŽE

# KLUPS

Klubová soutěž se vyhlašuje  
vždy ve čtyřech kategoriích:

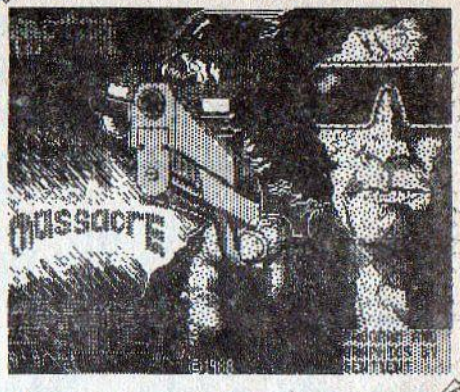
1. grafická
2. hudební
3. scénářistická
4. programátorská

Grafické kategorie soutěže se může zúčastnit každý, kdo namaloval nějaký pěkný obrázek (např. v ARTstudiu a pod.), nebo kdo vytvořil nějakou hezkou animaci či pozoruhodnou grafiku do programu (figury, místnosti počítačových her aj.)

Príspevek do této části soutěže však musí být uložen na kazetě nebo disketě ve formátu SCREENS

## HUDEBNÍ KATEGORIE SOUTĚŽE

se může zúčastnit každý, kdo vytvořil v nějakém hudebním editoru (WHAM, ORFEUS nebo i ve vlastním) zajímavou hudbu nebo znělku. Príspevek do této části soutěže musí být uložen na kazetě nebo disketě jako spustitelný program, který se po zavedení do počítače sám spustí.



## SCÉNÁŘISTICKÉ KATEGORIE SOUTĚŽE

se může zúčastnit každý, kdo vymyslel nějaký kvalitní scénář pro počítačovou hru nebo jiný zábavný, popř. vzdělávací program. Príspevek do této části soutěže musí být vypracován písemně (nejlépe na psacím stroji nebo vytisknutý z textového editoru na tiskárně). Rozhodně musí být zpracován přehledně a čitelně.

## PROGRAMÁTORSKÉ KATEGORIE SOUTĚŽE



se může zúčastnit každý, kdo vytvořil nějaký užitečný systémový program nebo nějakou zábavnou hru (jak v Basicu i v jeho variantách BB, Laser či Mega, nebo použil kompilátor, nebo pracoval v assembleru). Príspevek do této části soutěže musí být uložen na kazetě nebo disketě jako spustitelný program, který se po zavedení do počítače sám spustí.

## PODMÍNKY VYHLÁŠENÍ

« KLUBOVOU SOUTĚŽ má právo vyhlásit kterýkoliv z registrovaných členů. Ten se pak stává vyhlášovatelem.

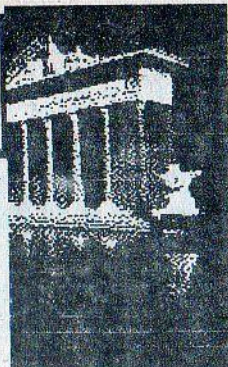
« k financování organizace soutěže může být jménem klubu získán vhodný sponzor (výrobce součástek, prodejce elektroniky ap.) Ten pak má právo po domluvě s klubovým vyhlášovatelem použít soutěže ke své bezvýhradní reklamě.

« soutěž se vyhlašuje přes kontaktní adresu klubu, kde jsou uloženy registrační karty členů, nebo přes redakci AP. Vyhlášovatel proto podává na kontaktní adresu vyhlásovací oznámení.

« vyhlásovací oznámení musí obsahovat doklad o nákupu a zaplacení cen pro vítěze všech kategorií soutěže, sponzorem nebo vyhlášovatelem. Dále pak údaje o časovém konání soutěže (od kdy do kdy mají zájemci zasílat své příspěvky do soutěže na kontaktní adresu).

« vyhlášovatel musí na kontaktní adresu poukázat minimální finanční obnos na poštovné, které bude vyžadovat organizace soutěže. Jedná se o náklady na rozeslání vyhlášení soutěže členům klubu, později na navrácení paměťových médií (kazety a diskety) se soutěžními příspěvky účastníkům soutěže a na konec na rozeslání výsledků soutěže členům klubu.

Tyto náklady si musí vyhlášovatel nejprve zjistit na kontaktní

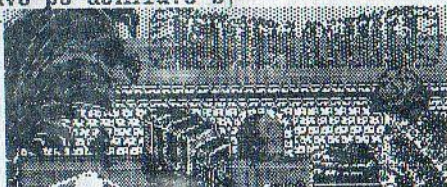


adrese klubu.

« vyhlášovatel má právo navrhnout jednoho člena dvou až tříčlenné hodnotící komise. Může to být zástupce jím získaného sponzora nebo on sám.

« hodnotící komise se sejde po datu ukončení zasílání příspěvků na místě, které bude náležitě vyhaveno, aby komise mohla soutěžní příspěvky dokonale prohodnotit. Toto místo oznámí vyhlášovatel rovněž na kontaktní adresu klubu.

« bližší informace lze získat na adrese tohoto klubu.



KONTAKTNÍ ADRESA:

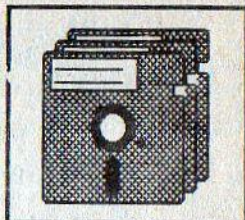
HITSOFT

Pavel Macek  
Nemanická 89  
HRDĚJOVICE  
373 61





# DTP MACHINE



Protože se na trhu již objevil komplet programů textového editoru TEXT MACHINE, který byl dosud znám jen ve verzi pro magnetofon, lze předpokládat, že si ho možná také zakoupíte.

Jeho cena není sice právě nejnížší, ale pokud vezmeme v úvahu, co vše nám celý komplet poskytuje, brzo zjistíme, že se tato investice opravdu vyplatí. Především tomu, kdo čas od času potřebuje vytvořit a pak i vytisknout kvalitní předlohu.

Základní program Text Machine (dále jen TM) je graficky orientovaný textový editor s vysokým komfortem ovládní, který umožňuje psát bez problémů znaky s diakritikou, pracovat s různými typy písma, kombinovat text s ilustracemi a semigrafikou, text zobrazovat metodou WYSIWYG a výsledný text tisknout na libovolné 7-mi nebo 9-ti jehličkové tiskárně nebo na BT100 či GAMACENTRUM 01.

Obrázek v textu lze prakticky umístit na libovolné místo buď celý nebo jeho zvolený výřez. To umožňuje vytvořit kolem obrázku např. rámeček (viz. obr. s disketami) nebo text psát kolem obrázku po obou stranách. V tomto případě to však jde jen pravoúhle. Skutečného obtékání textu kolem obrázku docílíme však také. Stačí obrázek přepsat do znakové sady a pomocí semigrafiky jej zobrazit v textu. Ukážku jak to jde udělat, vidíte na obrázku ze hry SABOTÉR, který byl takto vložen do textu.

K tomu je ovšem potřeba speciální program, který obrázek z obrazovky přepíše na místo zna-

kové sady, ke které si po jejím nahrání do TM vytvoříme semigrafickou rutinu v semigrafickém editoru (smgca). Pak si obojí (font i smgca) zase uložíme na disk pro další použití.

Program, o kterém zde píšou bude součástí knihovny fontů a semigrafiky TM-FOS2. Ta bude obsahovat další nové fonty pro psaní textu, fonty pro zobrazení semigrafikou, tzv. titulkové a fonty ve kterých už jsou nějaké ty obrázky a různé symboly vhodné pro použití při psaní různých návodů a pod. Dále bude tato disketa ještě obsahovat několik obrázků již připravených pro různé výřezy. Aby měl uživatel práci ještě více pohodlnější nalezneme na disketě ještě i poslední verzi programu ArtistII pro práci s disketovou mechanikou. Zde si totiž lze velice snadno obrázek předem připravit tak, že se zHAVÍ barev a nebo pomocí přesouvání okének lze sestavit konečnou podobu proti té původní.

Celá tato disketa TM-FOS2 by měla být dokončena asi tak za měsíc (konec května) a lze si jí objednat už nyní. Jen nezapomenete uvést, zda vlastně skutečně DTP-MACHINE od firmy ULTRASOFT nebo jen upravenou verzi podle AP11/92.

Její cena se bude pohybovat kolem 80,-Kč. Nezapomenete k objednávce přiložit také svou vlastní disketu.





Výhody a přednosti programu TEXT MACHINE není jisté nutné nějak zvlášť zdůrazňovat. Chceme Vám nabídnout jeho rozšíření.

V programu je totiž možné umístit obrázek na libovolné místo v textu a nechat ho kolem obrázku tzv. obtékat.

To však jde pouze v jeho pravých úhlech, tedy ve tvary docílit toho, aby text kolem obrázku "obtékal" podle jeho skutečných tvarů?

Jediné řešení, které se zde nabízí je, vložit ho do textu jako font (znaky) a pak lze z obrázkem libovolně v textu pohybovat, tj. využít pro něho všechny funkce jako pro běžný font. Jak obrázek umístit do fontu? Bude to celkem snadné, pokud si pořídíte program TM-FOS, který to umí.



## rozšiřte si možnosti svého programu

Kompletní disketa „TMFOSv04“ nabízí několik možností na přípravu vybavení Text Machine, tj. znakové sady (fonty), semi-grafiku a obrázky.

Pokud máte již větší množství disket a na nich uložené různé znakové sady ať již ve formátu pro Desktop nebo obyčejné 8x8 pixel a chcete je využít v TM, pak toho lze docílit za pomoci programu CONVERTOR, který umí přenést font z Desktopu do fontu pro TM (ale i opačně) stejně jako font 8x8 pixel vám převede na font 12x8 pixel pro použití v TM i v Desktopu. Jestli nějaké a jaké fonty na disketách máte, vám zase zjistí rutina, která podle vaší definice nalezne na disketě a vypíše vám definované soubory (fonty z Desktopu, TM, 8x8 a SCREENY)

Jak vlastně program TMFOS využít se dovíte jednak z příloženého manuálu ale také z ukázky, kterou můžete vidět na této stránce. V horní části je obrázek medvěda převeden do fontu úplně celý. Do textu se tedy vloží jako font pomocí utility „smgca“. Tím si lze obrázek libovolně posouvat v textu třeba i během psaní. Ten druhý obrázek, který je po pravé straně tohoto textu, nebyl vložen do fontu celý, ale jen jeho levá horní čtvrtina a to proto, aby mohl být textem „obtékán“. Druhá polovina je již obrázek, který byl do textu vložen opět jako jedna jeho čtvrtina. Tím bylo docíleno toho co zde vidíte.

Až se seznámíte se všemi programy, které disketa „TMFOSv04“ nabízí, jistě přidáte na další způsoby jejich využití.



● JAK SI „TMFOSv04“ OBJEDNAT ? ● Protože nemám v současné době žádnou zásobu disket, musíte si poslat svou vlastní disketu a uvést zda vlastníte originál TM pro D40 od ULTRASOFTU, nebo zda máte ještě kazetovou verzi u které jste si provedli úpravu podle AP11/92. Disketu včetně manuálu obdržíte za cenu 120,-Kč + poštovné jako dobírku.

Pavel Rak

# systemové proměnné

Dokončení z minulého čísla

Následuje přehled SP o kterých se mi podařilo něco získat. Na závěr je uvedena tabulka se stručnou charakteristikou.

\*\*\*\*\* KSTATE 23552 \*\*\*\*\*  
počítač používá při vyhodnocování stlačené klávesy nebo při stlačení dvou kláves.

\*\*\*\*\* LAST\_K 23560 \*\*\*\*\*  
obsahuje kód poslední stlačené klávesy s ohledem na stlačení SHIFTu, rozlišuje tedy mezi x a X (malým x a velkým X)

\*\*\*\*\* REPDEL 23561 \*\*\*\*\*  
obsahuje dobu po kterou musí být stlačena klávesa, aby došlo k opakování, čas je v 1/50 sec., pokud chcete někomu ztížit editování vašeho programu, tak dejte POKE 23561,1 nastavení v ROM na #1270 - hodnota je v registru H

\*\*\*\*\* REPPER 23562 \*\*\*\*\*  
doba v 1/50 sec. která určuje interval opakování při stlačení klávesy, pro rychlejší editování v Basicu tam dejte 1.

\*\*\*\*\* DEFADD 23563 \*\*\*\*\*  
adresa argumentu uživatelské funkce DEF FN

\*\*\*\*\* K\_DATA 23565 \*\*\*\*\*  
ukládá se barevná informace vložené klávesnicí před dalším zpracováním

\*\*\*\*\* TVDATA 23566 \*\*\*\*\*  
obsahuje řídicí znaky o AT, TAB a informace o barvě

\*\*\*\*\* STRMS 23568 \*\*\*\*\*  
obsahuje adresy kanálů připojených k jednotlivým proudům, při inicializaci je prvních 15 bajtů přeneso z ROM, přenos na adr. #123b, tabulka je na #15af

\*\*\*\*\* CHARS 23606 \*\*\*\*\*  
říká kde se nachází font pro výpis znaků na obrazovku, adresa je menší o 256, protože prvních 32 znaků z tabulky Spectrum nevypisuje jako znaky, má tam totiž své řídicí znaky např. pro barvy, pozici AT... pokud máte svůj font (např. z ART STUDIA) o délce 768 B a chcete jej nahrát do paměti na adresu 32000, tak ho tam nahrajte příkazem:  
LOAD "FONT"CODE 32E3

před LOAD je lepší snížit ram-top příkazem CLEAR 31999, pokud jste si nenastavili jinak; pak stačí napsat:  
RANDOMIZE 32E3: POKE 23606,PEEK 23670: POKE 23607,PEEK 23671  
po odeslání řádku kl. ENTER by jste měli vidět ten svůj font. Návrat k původnímu fontu v ROM udělá:

POKE 23606,0: POKE 23607,60  
nastavení fontu v ROM je na #121c, font v ROM začíná na #3c00 a má délku 768 B

\*\*\*\*\* RASP 23608 \*\*\*\*\*  
udává délku varovného BEEPu, např. při psaní řádku přes celou obrazovku, editaci řádku s REM ve kterém je nějaký stroják a pod.

\*\*\*\*\* PIP 23609 \*\*\*\*\*  
délka pípnutí při stisku klávesy (vyjímáje CAPS a SYMBOL SHIFT), ideální hodnota je 0; nastavení RASP a PIP je v ROM na #1212, dává se do reg.BC

\*\*\*\*\* ERR\_NR 23610 \*\*\*\*\*  
na této adrese je hodnota o 1 menší než je kód chybového hlášení, například při hlášení

0 OK, 0:1 je v této SP hodnota 255, pokud chcete program ukončit nějakým efektním chybovým hlášením, tak dejte na poslední řádek prováděného programu: POKE 23612,XX kde XX je kód chybového hlášení -1 (seznam chybových hlášení najdet v příručce k počítači)

\*\*\*\*\* FLAGS 23611 \*\*\*\*\*  
různá návěští pro Basic, nespecifikovatelné

\*\*\*\*\* TVFLAG 23612 \*\*\*\*\*  
návěští pro řízení obrazovky

\*\*\*\*\* ERR\_SP 23613 \*\*\*\*\*  
adresa položky zásobníku ve kterém je uložena adresa pro obsluhu chyby (stručně řečeno; pokud se interpret vrací s nějakým chybovým hlášením, tak sejme z ERR\_SP nějakou adresu, z této adresy si vyzvedne hodnotu (od 0 po 65535) na kterou nastaví reg. PC (Program Counter) a skočí na tuto adresu, pokud v paměti na adrese určené obsahem ERR\_SP budou nuly, program při jakémkoliv pokusu o výpis chybového hlášení provede reset (vtipné že) **Nezapomeňte, že mezi chybové hlášení patří i OK**

\*\*\*\*\* LIST\_SP 23615 \*\*\*\*\*  
adresa kam skočí interpret pokud úspěšně ukončí listing programu, v praxi je totožná s obsahem ERR\_SP, hodnota je vždy o 3 bajty menší než je hodnota ramtop

\*\*\*\*\* MODE 23617 \*\*\*\*\*  
obsahuje režim kurzoru při editaci řádku

\*\*\*\*\* NEWPPC 23610 \*\*\*\*\*  
obsahuje číslo řádku na který má skočit příkaz GO TO, GO SUB, změnou lze počítač donutit aby skákal na námi zvolenou řádku

\*\*\*\*\* NSPPC 23620 \*\*\*\*\*  
určuje číslo příkazu na který má skočit GO TO, GO SUB, kombinací NEWPPC a NSPPC je možno spustit program příkazy:  
POKE 23618,0: POKE 23619,0: POKE

23620,0

\*\*\*\*\* PPC 23621 \*\*\*\*\*  
číslo právě prováděného řádku, probíhá od 0-9999, protože Basic větší čísla nezvládá

\*\*\*\*\* SUBPPC 23623 \*\*\*\*\*  
číslo vykonávaného přík. v PPC

\*\*\*\*\* BORDC 23624 \*\*\*\*\*  
barevná informace pro spodní část obrazovky (editační zóna), zkuste toto:

POKE 23624,166: INPUT "": PAUSE 50: POKE 23624,7

pokud změníte BORDC změní se barva okraje obrazovky (BORDER) a naopak, změníte-li BORDER stejnojmenným příkazem, nastaví se BORDC a tím spodní část obrazovky na barvu BORDERu

\*\*\*\*\* E\_PPC 23625 \*\*\*\*\*  
číslo řádku na kterém je kurzor

\*\*\*\*\* VARS 23627 \*\*\*\*\*  
adresa počátku proměnných

\*\*\*\*\* DEST 23629 \*\*\*\*\*  
adresa proměnné při jejím vyhodnocování, jinak 0

\*\*\*\*\* CHANS 23631 \*\*\*\*\*  
adr. začátku informací o kanálech, bez Interface 1 je tam 23734 (#5cb6)

\*\*\*\*\* CHURCHL 23633 \*\*\*\*\*  
obsahuje adr. z oblasti kanálových dat pro právě otevřený kanál

\*\*\*\*\* PROG 23635 \*\*\*\*\*  
adr. počátku Basicového programu, bez INTRF 1 23755 (#5ccb), lze využít k utajení prvního řádku programu příkazy:

POKE 23755,0: POKE 23756,0  
který ovšem již nelze editovat (editaci umožní změna adresy 23755)

\*\*\*\*\* NXTLIN 23637 \*\*\*\*\*  
začáteční adresa dalšího programového řádku, který má být programem proveden

\*\*\*\*\* DATADD 23639 \*\*\*\*\*  
obsahuje adresu čárky za poslední položkou v řádku DATA, která byla přečtena příkazem READ

\*\*\*\*\* E\_LINE 23641 \*\*\*\*\*  
obsahuje adresu editované oblasti v RAM, kde zrovna editujete řádek

\*\*\*\*\* K\_CUR 23643 \*\*\*\*\*  
adresa kurzoru (že by řádek, sloupec?)

\*\*\*\*\* CH\_ADD 23645 \*\*\*\*\*  
adresa znaku v oblasti INPUT nebo oblasti proměnných, který bude jako příští vypsán na obrazovku

\*\*\*\*\* X\_PTR 23647 \*\*\*\*\*  
adresa prvního znaku ve kterém je syntaktická chyba, tj. znaku následujícího po >?<

\*\*\*\*\* WORKSP 23649 \*\*\*\*\*  
adresa dočasného pracovního prostoru

\*\*\*\*\* STKBOT 23651 \*\*\*\*\*  
adresa začátku oblasti zásobníku kalkulátoru

\*\*\*\*\* STKEND 23653 \*\*\*\*\*  
adresa vrcholu zásobníku kalkulátoru (začátek volné paměti)

\*\*\*\*\* BREG 23655 \*\*\*\*\*  
používá kalkulátor jako své počítadlo

\*\*\*\*\* MEM 23656 \*\*\*\*\*  
adresa začátku zásobníkové paměti kalkulátoru, většinou MEMBOT

\*\*\*\*\* FLAGS2 23658 \*\*\*\*\*  
další parametry pro Basic, změnou na 8 lze programově přepnout na velká písmena při psaní textu (jednoduše řečeno zapne CAPS LOCK), zpětně lze přepnout na malá písmena změnou na >0<

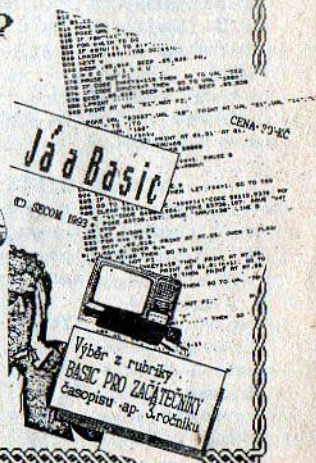
\*\*\*\*\* DF\_SZ 23659 \*\*\*\*\*  
udává počet řádků v dolní části obrazovky; zajímavý je efekt při POKE 23659,1 - v programu můžete dát POKE 23659,0 a bude vám fungovat i PRINT AT 22 nesmíte se však vrátit do Basicu s nějakou hláskou, např. při INPUTu, scrollu?, BREAK, OK a pod., protože počítač nemá kam vypisovat text a zakousne se (né do Vás, ale sám do sebe)

\*\*\*\*\* S\_TOP 23660 \*\*\*\*\*  
obsahuje číslo řádku, který bude zobrazen jako první při automatickém listingu  
(dokončení příště)

## UŽ JSTE SI JE OBJEDNALI ?

Pokud ještě ne  
pak honem využijte  
příloženou  
objednávku!

Pokud si budete  
objednávat TM-FOS  
na disketě, NEZAPOMENTE  
UVĚST zda skutečně  
vlastníte disketovou  
verzi DTP-MACHINE,  
nebo máte jen úpravu  
podle AP 11/92



# LOGICKÁ HRA

## SAMOTÁŘ

Jistě každý někdy hrál a nebo alespoň hrát zkusil velice rozšířenou a oblíbenou společenskou hru "DÁMA". Tato hra, založená na vzájemném přeskakování a odebírání figur má totiž se hrou *Samotář* něco společného.

Po spuštění hry se před Vámi objeví hrací plocha, složená z čtvercových políček, na které se rozmístí figury a to tak, že na každém políčku se nachází jedna figura. Na hrací ploše se rozhlíká ovládací kurzor o velikosti jednoho políčka. Směr pohybu kurzoru ovládáme klávesami Q, A, O, P nebo *keypston joystickem*.

Stojíme před prvním úkolem. Kurzorem najdeme figuru, kterou pak aktivaci (klávesa *N* nebo *keypston Fire*) vyjme z hrací plochy. Nyní může začít vlastní hra. Úkolem hráče je vyskákat navzájem všechny figury, které na hrací ploše zbyly tak, aby zůstala jen jedna - *samotná* (odtud název). Skoky provádíme tímto způsobem: najdeme kurzorem figuru s níž chceme skákat a potom stiskneme aktivaci. Tím jsme figuru uchopili do ruky (dalším stiskem aktivace ji můžeme opět pustit, pokud jsme se spletli).

Zbývá ukázat směr, kterým chceme skákat. Jestliže je za

figurou, přes níž chceme skákat volně políčko, skok se po zadání směru uskuteční. Když už není možné žádnou figurou skákat, stiskněte pro ukončení a vyhodnocení hry, klávesu O (nula).

Na počátku hry, než vyjmete první figuru, je možné klávesou I měnit světlost hrací plochy.

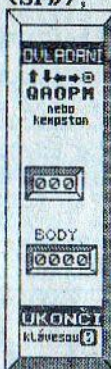
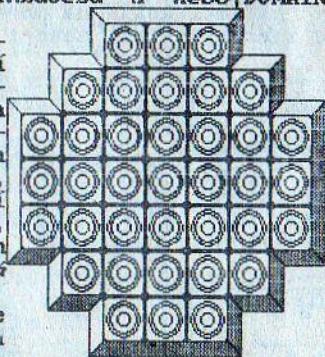
Hra je z větší části naprogramována v Basicu a doplněna o několik krátkých strojových rutin.

Je to vlastně prvotina, která vznikla pod hlavičkou klubu amatérských tvůrců software NITSOFT.

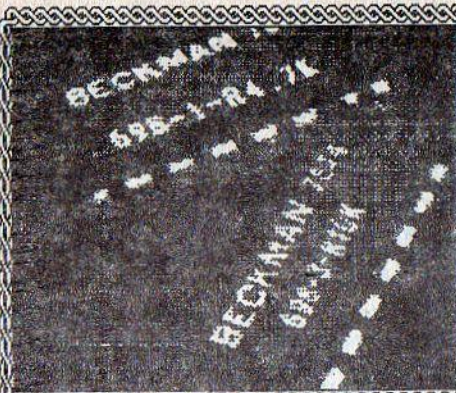
Všem hráčům přeji mnoho dobrých zábav s touto hrou.

*Přesto, že nebývá zvykem zveřejňovat na stránkách AP popisy her (redakce se totiž hrám záměrně vyhýbá), udělám rád výjimku vždy, pokud půjde o hru vytvořenou amatérem.*

Tuto hru si můžete objednat v rámci programů SÍŤE PUBLIC DOMAIN (SPD), pokud si pošlete vlastní kazetu, nebo disketu a přiložíte 20,- korun českých.



i jinde, ještě více podporovat amatérské programování na ZXS a DIDAKTIK. Pokud tedy i u Vás existuje nějaký podobný klub či skupina programátorů, dejte nám do AP vědět, nebo rovnou pošlete nějaké své výtvary ke zveřejnění.



# KURZ PROGRAMOVÁNÍ STROJOVÉHO JAZYKA

pro AP píše Pavel Macek



Vítám Vás u dalšího pokračování seznamování se strojovým jazykem assemblerem.

Kdo se poctivě prokousal kurzem dosud zveřejněným v AP, jistě pochopil, že to vše jsou základy velice nutné k tomu, abychom na nich mohli stavět další vědomosti.

Hned na počátku jsme se dozvěděli, že počítač zpracovává čísla (co taky jiného, že?) Řekli jsme si rovněž o různých způsobech vyjádření čísel v počítači. Nyní si povíme jakým způsobem lze čísla z paměti nebo registrů počítače zobrazit na obrazovku.

## JAK VYPADÁ ČÍSLO NA OBRAZOVCE

Na obrazovce je číslo vyjádřeno pomocí číselných znaků. Protože musíme uvažovat o všech typech vyjádření čísla (dvojkové, šestnáctkové, desítkové), bereme v úvahu tyto znaky:

0 1 2 3 4 5 6 7 8 9  
A B C D E F  
nebo a b c d e f

Abychom mohli číslo na obrazovku zobrazit, musíme ho nejprve převést na sled kódů znaků, ze kterých je složeno. Ten ovšem bude při různých typech vyjádření čísla také různý.

Zkusme si tedy nejprve dvojkové

vyjádření čísla.

## DVOJKOVÉ BINÁRNÍ ČÍSLO

Jak víme, v osmibitovém počítači je dvojkové číslo vyjádřeno kombinací osmi nul a jedniček. Víme také, že jednotlivá číslíčka vyjadřuje hodnotu jednoho bitu.

Budeme tedy postupně zjišťovat hodnoty jednotlivých bitů registru a hned je za pomoci znaků nula nebo jedna zobrazovat.

Věříte, že je možné zobrazit dvojkové vyjádření čísla kamkoliv na obrazovku programem, který zabere sotva tři desítky bajtů?

Prostudujte si pečlivě program č.1, protože Vám ukáže jak na to!



## PROGRAM ČÍSLO 1.

```
ld a,129
binar ld hl,18432+10
      ld b,8
bin1  rla
      ld de,15616+128+8
      jr c,bin2
      ld de,15616+128
bin2  push af
      push hl
      ld c,8
bin3  ld a,(de)
      ld (hl),a
```

```

inc de
inc h
dec c
jr nz,bin3
pop hl
inc hl
pop af
djnz bin1
ret

```

Instrukce před návěstím **binar** ukládá do registru A číslo, které budeme zobrazovat. Ukažme si jak bude toto číslo, vyjádřeno dvojkově, vypadat:

```

128 64 32 16 8 4 2 1
1 0 0 0 0 0 0 1
128 + 1 = 129

```

Od návěští **binar** potom začíná vlastní zobrazovací program. Do registrového páru HL je uložena adresa, na niž budou znaky, ze kterých je vyjádření čísla složeno, zobrazovány do obrazovky. S tím musíme při zadávání této adresy také počítat.

V našem programu se bude vyjádření čísla zobrazovat na prvním řádku druhé třetiny obrazovky a deset znaků od levého okraje.

Protože doptávat se budeme postupně na osm bitů čísla, následuje zadání tohoto parametru do registru B. Pišeme-li po obrazovce zleva doprava, logicky musíme začít s největším (levým) bitem čísla. Rotace bitů doleva rla nám umožní testovat bity v požadovaném pořadí.

Zatím do registrového páru DE připravíme adresu tvaru čísla jedna (jestliže mezera má kód 32 a nula kód 48, pak rozdíl mezi těmito kódy je 16).

Tvar znaku představuje osm bajtů od adresy tvaru mezery bude adresa tvaru nuly vzdálena  $16 \cdot 8 = 128$  a tvar jedničky o dalších osm bajtů. Adresa tvaru mezery je 15616, proto adresa tvaru jedničky bude

```
15616 + 128 + 8
```

Nyní se testovaný bit dostal do CARRY a my se můžeme ptát zda

je CARRY nastaveno. Pokud bude CARRY nastaveno na jedna, znamená to, že bit, který jsme testovali je rovněž jedna a program pokračuje od návěští **bin2**. Uschová obsah registru A (kde je zobrazené číslo) a také obsah registrového páru HL (kde je adresa do obrazovky, kam se číslo vypisuje). Do registru C připraví počet bajtů tvaru znaku. Dál následují instrukce, které Vám jsou dobře známy z programů, zobrazujících znak na obrazovku. Tvar znaku, adresovaný registry DE, se postupně pro osm mikrolinek vkládá na adresy do obrazové paměti, které jsou v registrech HL. Počet mikrolinek (registr C) se přitom snižuje, až je znak zobrazen celý. Pak následuje obnovení adresy do obrazovky a její zvětšení na pozici znaku dalšího (registry HL). Obnoví se také obsah registru A a pokud se již nezobrazily všechny bity, opakujeme rotaci a test dalšího bitu.

V případě, že bit v CARRY byl nulový, doplní se do registrového páru DE adresa tvaru znaku nula a program pokračuje dál od návěští **bin2**.

Program si napište v assembleru jako zdrojový text a zkusíte si je spustit s různými čísly v registru A.

Vidíte, nyní už dokážete zobrazit obsah registru dvojkově na obrazovku. Když budete chtít zobrazit obsah jiného registru nebo paměťového místa, uložíte ho vždy nejprve do registru A a pak spustíte tento program od návěští **hina**.

ŠESTNÁCTKOVÉ (HEXADECIMÁLNÍ)  
ČÍSLO

Šestnáctková číselná soustava má za základ šestnáct čísel:

```
0123456789ABCDEF
```

K zobrazení takových čísel na obrazovku už nevystačíme se dvěma znaky jako tomu bylo u

čísel dvojkových, ale budeme pracovat se všemi šestnácti znaky. Osmibitové číslo v registru můžeme pro tento účel rozdělit prakticky na dvě stejné poloviny. Získáme tak dvě čtyřbitová čísla, která mohou nabývat pouze hodnot od nuly do patnácti a každé tak lze vyjádřit jediným znakem šestnáctkové soustavy. Podívejme se například na číslo 62:

```

128 64 32 16
 8   4   2   1
0 0 1 1 1 1 0
      3E

```

Tímto způsobem rozkládáme osmibitové číslo pro šestnáctkové vyjádření i v následujícím programu:



## PROGRAM ČÍSLO 2.

```

        ld    a,62
hexa    ld    de,18432+10
        push af
        and #f0
        rlca
        rlca
        rlca
        rlca
        call hex1
        pop  af
        and #0f
hex1    ld    bc,15616+128
        cp    10
        jr   c,hex2
        sub  10
        ld    bc,15616+264
hex2    ld    h,0
        ld    l,a
        add  hl,hl
        add  hl,hl
        add  hl,hl
        add  hl,bc
        ex   de,hl
        push hl
        ld    b,8
hex3    ld    a,(de)
        ld   (hl),a
        inc  de

```

```

inc    h
djnz  hex3
pop    de
inc    de
ret

```

Na počátku zadáme do registrového páru DE adresu kam chceme číslo vypsat. Nyní uschováme registr A a funkcí `and` maskujeme horní čtyři bity čísla v tomto registru. Co se stane? Dolní čtyři bity v registru A se vynulují. Potom stačí posunout horní čtveřici bitů na místo dolní, abychom bity dostali pod váhy čísel od nuly do patnácti.

Číslo, které registr A v tuto chvíli obsahuje hned zobrazíme U návěští `hex1` se připraví do reg. páru BC adresa tvaru číselného znaku nula. Provedeme test, zda číslo není větší jak deset. V našem případě (zobrazujeme číslo 62), tam bude číslo 3, takže skáče rovnou na návěští `hex2`, kde toto číslo vynásobíme osmi a přičteme k adrese tvaru znaku nula. Tak získáme adresu tvaru znaku tři který v zápětí u návěští `hex3`, zobrazíme.

Protože jsme návěští `hex2` volali instrukcí `call`, po instrukci `ret` se ocitne program zpátky za instrukcí `call`. Obnoví se obsah registru A a tentokrát se funkcí `and` maskují dolní čtyři bity. Ty se nacházejí přímo na vahách od nuly do patnácti a proto pokračuje program rovnou návěštím `hex1`.

Tentokrát bude v registru A číslo 14. To je větší než deset a tak dojde k odečtení deseti a nastavení nové adresy do reg. páru BC. V registru A je nyní číslo 4 a v BC je adresa tvaru znaku "A". Znakem čtvrtým od "A" bude znak "E" a to je právě znak, který potřebujeme, proto program pokračuje náv. `hex2`.

U instrukce `ret` program tentokrát končí.

(pokračování příště)



**B S T****NEZNAMENÁ BRITISH SUMMER TIME****ALE BROSOFT A NEB****PROGRAMY JIŘÍHO BROSSMANNA**

Ač to asi někoho udiví, lze kromě grafiky na obrazovce zobrazovat i texty.

S tímto efektem se setkáváme v některých textových hrách, systémových programech, systému ZXS a ostatních programech, které používají PRINT nebo jeho ekvivalenty.

S PRINTem se dají provádět různé zajímavé věci (např. tisk na obrazovku). Občas však potřebujeme dělat věci ještě zajímavější a potom si musíme pomoci nějakým programem. V životě programátora se například vyskytnou situace, kdy musí vypisovat větší množství textů a nechce pořád používat PRINT "ble, ble, ble". Je to sice rychlé, ale neefektivní a neefektivní. Mnohem lepší je tisknout text přímo naformátovaný po slovech. Vyhne se tak trapné situaci, kdy se část slova zobrazí na jednom řádku a zbytek na řádku druhém.

K tomuto účelu slouží uvedený program. Je ve své podstatě primitivní a přesto účinný. Rozdělí text uložený v proměnné Y\$ po slovech (text mezi dvěma mezerami). Přesahuje-li některé slovo přes řádek, je vytisknuto až na řádek další. Nejlépe to uvidíte, když to vyzkoušíte.

Pokud máte větší obrazovku a vleze vám tam víc znaků než 32 pak si můžete program upravit. Na řádek 130 dáte místo čísla 31 délku řádku-1 (protože se zobrazuje od 0).

Potřebujete-li text odřádkovat pak použijte ASCII kód ENTER, což je možno realizovat například

pomocí příkazu CHR\$ 13 (uložen je v programu v E\$).

Chcete-li mít několik slov vedle sebe na řádku, pak si můžete vyrobit vlastní znakovou sadu, kde bude třeba podtržítka nahrazeno mezerou. Program to samozřejmě nepozná, a tak lze např. udržet "a" před slovem, nebo tři tečky za ním, na jednom řádku.

Program je celkem chytrý (hm...), ale přesto bude určitě rychlejší když ho zkompilujete

**Poznámka:** pozor na to, že si program pokaždé smaže obrazovku.

```

1 REM PRO VAŠE POČÍTAČE TO
NEJHORŠÍ! PROGRAMY BroSoftu!
10 BORDER 0: PAPER 0: INK 7:
BRIGHT 1: CLS
20 LET E$=CHR$ 13
99 GO TO SE2
100 REM TISK
110 CLS: LET Y$=Y$+" "
120 LET a=1: LET b=1: LET c=0
130 IF c>=31 THEN LET c=b-a:
PRINT
135 IF Y$(b)=CHR$ 13 THEN LET c
=0: PRINT Y$(a TO b):LET a=b+1
140 IF Y$(b)="-" THEN PRINT Y$(
a TO b):LET a=b+1
150 IF b=LEN Y$ THEN RETURN
160 LET b=b+1: LET c=c+1
170 GO TO 130
500 REM PROGRAM
510 LET Y$="MÁTE JEDINOU ŠANCI
NA PŘEŽITÍ:OBJEDNAT SI AP!!!"
520 GO SUB 100

```

(Tento článek berte jako ukázkou toho, kolik věcí se dá napsat o osmi řádcích programu).

Tento program je zejména určen pro majitele didaktiků (tzv. "didaktisté")

Jak všichni víme, má Didaktik Gama více bank - celkem 2. Banka je adresový prostor od adresy 32768 (\*8000) do konce RAM. Banky se mezi sebou přepínají pomocí příkazu OUT 127, číslo banky (mimořádně, přes port 127 se také programuje paralelní interface).

U Didaktiků 1989 se navíc zachováva obsah banky 1 po resetu. Basicovský OUT 127,1 přenese navíc obsah banky od 32768 po RAMTOP (nastavíme pomocí příkazu CLEAR X). Tento problém v assembleru odpadá - banka se pouze přepne.

Toho se dá celkem pěkně využít viz např. GAMARAMDISK2, který je ale příliš dlouhý a často se nedá použít, protože zabírá asi 3,5 kB. Tento problém odpadá, když si přepíšete následující program.

Program je umístěn od adresy 23296, tedy v bufferu tiskárny, ale velice snadno si ho upravíte **Vždy však musí ležet pod adresou 32768 !**

Jak program funguje?

Nejprve se naplní do registrů HL, BC, DE a A hodnoty, které určují:

odkud - HL - 23295  
odkud se má provádět přenos kam - BC - 23298  
kam se má blok přenést kolik - DE - 23300  
kolik bitů chceme přenést bank - A - 23302  
ze které banky do které se má přenést

Do registru A vložíme nulu chceme-li přenos z 1. banky do 0. banky, a jedna pokud jde o přenos z banky 0. do banky 1. Instrukci `ex af,af` se provede výměna párových registrů `af` a

`a'f'`. Tato výměna se provádí protože není další registr, do kterého by se mohlo uložit číslo banky. Pak se registr A xoruje 1 což má za následek jeho nastavení, je-li v A nula, nebo vynulování pokud je A 1. Příkaz `OUT (*7F),A` vyšle hodnotu z A na port 127, a pak se opět vymění párové registry `af` a `'f'`. Do A se vloží přenesená hodnota z adresy HL a po přepnutí banky se vloží na adresu BC. Registr DE slouží jako počítadlo.

Po spuštění programu je třeba dát pozor na to, že počítač zůstane v cílové bance! Snadno se toho zbavíte tím, že po skončení smyčky přepnete pomocí OUTu ze strojáku banku zpět. Není to nic slavného, ale kdo vám dneska dá RAMDISK zdarma? Navíc se dá program umístit prakticky všude (to se může hodit crackerům).

Program berte spíše jako inspiraci k vytvoření něčeho lepšího a velkolepějšího.

**P.S.:** Nevím jistě jestli nebude mít program destruktivní vliv na dlouhý Basic, proto si ho raději vyzkoušejte, než si něco zničíte.

Výpis programu najdete na další stránce.



# B S T

## RAMDISK PRO DIDAKTIK GAMA

### VÝPIS PROGRAMU

```

4      0000
5      0000      ;*****
6      0000      ;* PRENOS *
7      0000      ;*****
8      0000
9      0000      ;      org 23296
10     0000
11     0000      odkud equ 32768
12     0000      kam  equ 16384
13     0000      kolik equ 6912
14     0000      bank  equ 0
15     0000
16     0000      ;bank=0 - z banky 1 do banky 0
17     0000      ;bank=1 - z banky 0 do banky 1
18     0000
19     0000 0000      dw  odkud
20     0002 0040      dw  kam
21     0004 001b      dw  kolik
22     0006 00      db  bank
23     0007
24     0007 2a005b      ld  hl,(23296)
25     000a ed4b025b      ld  bc,(23298)
26     000c ed5b045b      ld  de,(23300)
27     0012 3a065b      ld  a,(23302)
28     0015 00      ex  af,af
29     0016 06      zac  ex  af,af
30     0017 ee01      xor  #01
31     0019 d37f      out (#7f),a
32     001b 00      ex  af,af
33     001c 7e      ld  a,(hl)
34     001d 00      ex  af,af
35     001e ee01      xor  #01
36     0020 d37f      out (#7f),a
37     0022 00      ex  af,af
38     0023 02      ld  (bc),a
39     0024 23      inc  hl
40     0025 03      inc  bc
41     0026 1b      dec  de
42     0027 7a      ld  a,d
43     0028 b3      or  e
44     0029 20eb      jr  nz,zac
45     002h c9      ret
46     002c      end

bank 0000 * kam 4000 * kolik 1000 * odkud 0000 *
zac 0016 *

```

errors:0

# DISKETOVÁ JEDNOTKA DIDAKTIK A STROJOVÝ KÓD

Miroslav Mošat

Môj článok nadväzuje na programovanie v strojomom kóde. Ale na začiatok by som chcel objasniť problematiku disketovej jednotky D-40, ktorú používa veľa užívateľov. Tento článok popíše podrobný popis ovládania disketovej jednotky zo strojáku, ale najskôr Vám opíšem pre orientáciu, ako to v mechanike funguje.

Mechanika má dva motorky, z ktorých jeden je na otáčanie diskety a druhý je krokový motor, ktorý posúva hlavu po diskete. Sú v nej dva snímače: jeden sníma identifikačný otvor na diskete, ktorým sa zisťuje, či je v mechanike disketa, a či sú zatvorené dvierka. Druhý snímač slúži na ochranu proti nežiadúcemu zápisu. Hlavy sú navzájom spojené, posúvajú sa spoločne, ich presný pohyb zabezpečuje krokový motor. Motor určuje 44 pozícií - stop - podľa mechaniky. Nakoľko už 44. stopa je až na doraz, tak sa zvyčajne nepoužíva. Spodná hlava je pevná, pričom hornú hlavu pritláča mechanizmus, ktorým zatvárame dvierka, tak pritlačíme aj hlavy. Pokiaľ sú zatvorené dvierka, tak aj hlavy sú pritlačené k diskete. Zvuky, ktoré sa ozývajú pri zápise, čítaní a formátovaní je vlastne pohyb krokového motora.

V prípade, že chceme čítať 45 stopu a väčšiu ozve sa z mechaniky známe vrčanie, ktoré je spôsobené tým, že krokový motor chce posunúť hlavu na ďalšiu pozíciu, ale hlava je už na doraz, takže motor už hlavu nemôže posunúť ďalej a tak sa ozve známym zavrčaním, ktoré

je podobné pri RESETE počítača. Najskôr treba poznamenať, že čítanie diskety prebieha len po sektoroch a teda systém MDOS si musí vybrať na ktorý sektor môže niečo zaznamenať, alebo sektor čítať.

## ČÍTANIE DISKETY SYSTÉMOM MDOS

Ako prvé sektor 0, potom tabuľka súborov, kde sa zistí, či požadované meno je na diskete, potom je zistený, ktorý sektor je pre súbor prvý a potom podľa FATky sa zisťuje, na ktorom sektore je ďalší záznam, až dovtedy, kým nie je načítaný posledný sektor súboru.

## ZÁPIS NA DISKETU SYSTÉMOM MDOS

Ako prvé sa načíta sektor 0, a zistí sa, či je nejaké voľné miesto na diskete, ale nezisťuje sa, či je dost voľného miesta. Ak je miesto, tak sa do adresáru zapíše meno nového súboru a až potom prebieha zapisovanie na disk a pritom využíva voľných sektorov použiteľných vo FATke; ak zistí, že sa už požadovaný sektor nezmeští na disk, tak je vypísané chybové hlásenie. To je ale chyba v ROMke, pretože pri katalógu diskety je súbor označený ako celý, ale nie je vôbec použiteľný - chýba časť progra-

mu. Niektoré programy majú tieto základné chyby ošetrené a teda nedochádza ku kolízii dát na diskete.

Základným systémom čítania, alebo zápisu je formát diskety. Formát diskety je zapísaný v nultom sektore, ktorí slúži aj ako identifikačný sektor. Základom formátovania je počet sektorov na stopu a počet stôp na diskete. Pre 360 kB je formát 40 stôp po 9 sektorov, rozšírený formát je 43 stôp po 10 sektorech (u počítačov PC s HD disketami je počet sektorov na stopu až 15).

Teda ak máme disketu naformátovanú na 360kB, teda 40\*9 a prepíšeme jej z inej diskety nultý sektor, ktorý má formát 43\*10, tak pri čítaní sa ozve hlásenie typu "Sektor not found" čo svedčí o tom, že systém kontroluje iba nultý sektor, a teda systém nezaujíma, či je prevedený aj iný formát (táto vec sa dá použiť aj na ochranu diskiet, ktoré nechceme, aby sa dali skopírovať sektor po sektore. Podobná ochrana je použitá v programe KOMANDER2, kde je disketa naformátovaná na 43\*10, a pritom v nultom sektore je označená ako formát 40\*9, čo pri bežnom kopírovaní diskiet po sektorech spôsobí chybu v načítaní sektoru práve takého, ktorý nieje naformátovaný a nieje prístupný).

Následujúci program umožňuje načítanie nultého sektoru z disketovej jednotky. Princíp čítania ostatných sektorov je podobný, avšak treba len zmeniť niektoré čísla. A ešte by som chcel upozorniť, že tento program je veľmi zjednodušený a teda sa neprevádzajú žiadne testy. Ak program spustíte bez zatvorených dvierok na disketovej mechanike dôjde k zablokovaniu programu a program bude pokračovať až po zatvorení dvierok, alebo sa môže aj niekde zaseknúť. No ale ešte pred prvým spustením programu je treba zadať POKE \*247,79 pretože inak RST 0 spôsobí reset počíta-

ča, no ale pokiaľ je uvedené POKE zadane, tak sa prepne na ROM D40 a program sa vráti za inštrukciu RST 0. Podprogram na adrese 55000 sa nesmie krokovať, v opačnom prípade sa program zasekne, vtedy sa totiž uskutočňuje priamo čítanie, alebo formátovanie diskety!!! Pri inštrukcii OUT 129,136 sa hardwarevo vnúti inštrukcia, ktorá skočí na adresu v registri IX a tým sa uskutoční prenos z diskety. Na adrese 9706 je inštrukcia INI a RET. Register HL je zvýšený o hodnotu 512 bajtov a od počiatočnej hodnoty HL je načítaný požadovaný sektor.

Z hľadiska MDOsu sú sektory číslované od nuly až po 860, pre ľahšie pochopenie sa uvádza len zvolený sektor na stope. Výpočet sektoru z hľadiska MDOsu sa vypočíta nasledovne: piaty sektor na 12.stope druhej strany by bol v MDOse očíslovaný ako:

na diskete naformátovanej na..

43\*10 to je:

$$12*10*2+5=245+10*1=255 \text{ sektor}$$

43\*9 to je:

$$12*9*2+5=221+9*1=230 \text{ sektor}$$

40\*6 to je:

$$12*6*2+5=149+6*1=155 \text{ sektor}$$

ak načítame 5 sektor na 12 stope prvej strany, tak pri 40\*9 to je;

$$12*9*2+5=221+9*0=221 \text{ sektor.}$$

```

*****
ent $      ;signál odkiaľ sa
           ; má program
           ; spustiť
rst 0      ;prepní na rom D40
di         ;zakaž prerušenie
ld hl, 9104 ;do hl adresu pre
           ; skopírovanie
ld de, 55000;načítanie na adre-
           ;su 55000
ld bc, 14  ;14 bajtov
ldir      ;kopíruje
ld a, 201  ;inštr. RET do A
ld(55014), a; na ukončenie pod-
           ; programu na adr.
           ; 55000

```

ld hl, 50000 ; do hl počiatočné  
uloženie načítané-  
ho sektora  
ld a, 5 ; do A číslo 5  
out (137), a ; a spusti motor  
ld a, 0  
out (131), a ; nuluj pozíciu  
ld a, 5  
out (137), a  
ld a, 0  
out (131), a ; to isté ako hore  
ld a, 208  
out (129), a ; signál pre ziste-  
nie pripravenosti  
mechaniky

\*ld a, 1  
out (133), a ; zapiš číslo 1. sek-  
tora na stope (po-  
číta sa od 1)

ld a, 0  
out (135), a ; zapiš číslo stopy  
(0-43)  
ld a, 28 ; hlavy na nultú  
stopu -  
out (129), a ; -do základnej po-  
zície (pri tomto  
kóde sa nastaví  
vždy nultá stopa)  
call CAKAJ ; čakaj pokiaľ me-  
chanizmus nastaví  
základnú pozíciu

ld a, 69  
out (137), a ; signál pre priamy  
prenos medzi D40  
a procesorom v  
počítači

ld ix, 9706 ; do ix adresu in-  
strukcie pre čítanie  
INI

\*ld a, 136 ; čítaj zvolenú  
stranu diskety  
(pre druhú stranu  
je to 138)

ld c, 135 ; nastav pred čítaním  
alebo zápisom

ld de, 256  
ld b, c  
call 50000 ; volaj čítací pod-  
program

ld a, 5  
out (137), a ; zmenš signál späť  
in a, (129) ; čítaj kód chyby  
(0=bez chyby,  
10. sekt. nenajdený)

push af ; uschovaj ho pre  
ďalšie použitie  
in a, (133) ; načítaj číslo se-  
ktora na stope

dec a ; sniž ho  
out (133), a ; a zapiš pre ďal-  
šie správne nastave-  
nie ďalšieho  
sektora  
call CAKAJ ; počkaj chvíľu  
xor a ; nuluj reg. A  
out (137), a ; zastav motor  
ld hl, 16119; do hl adr. pamäte,  
na ktorú  
ld (hl), 79 ; sa zapiše hodnota  
79 pre ďalšie spu-  
stanie programu  
call 5888 ; prepni na romku v  
počítači  
ei ; povol prerušenie  
pop af ; v reg. A kód chyby  
ret ; vráť sa

CAKAJ ld bc, 85534  
C1 djnz C4  
dec c  
jr nz, C1  
ret

**Blok, ktorý je medzi hvie-  
zdičkami sa mení, teda zápis  
nultého sektoru (ináč známe  
RESTORE \*a:0,.....) by bol:**

```

* ld a, 1
  out (133), a
  ld a, 0
  out (135), a
  ld a, 28
  out (129), a
  call CAKAJ
  ld a, 69
  out (137), a
  ld ix, 9709 (pozn. 1)
  * ld a, 168 (pozn. 2)

```

(pozn. 1; do IX adr. instr. pre  
zápis OUTI)  
(pozn. 2; pre 2. str. je to 170)

**Hore uvedeným programom sa  
dá aj opraviť nultý sektor na  
diskete. Avšak musíme mať už  
predtým načítaný nultý sektor  
z inej diskety (1.sektor na  
nulte stope prvej strany).**

**Čítanie 5. sektora na 11.sto-  
pe a druhej strany:**

```

* ld a, 5
  out (133), a
  ld a, 11
  out (135), a

```

```
ld a, 28
out (129), a ; (viz poznámka
call CAKAJ v ďalšom
ld a, 69 textu)
out (137), a
ld ix, 9706
* ld a, 138
```

Ak bol pred týmto zadaný OUT 131,0 tak sa hlava posunie na 11.stopu!!! Z toho vyplýva, že ak spustíme 5krát po sebe mechanika sa bude snažiť nastaviť 55. stopu. Teda začne vrčať krokový motor! Ale pokiaľ bola hlava nastavená na 40.stope, tak vrčanie sa ozve už pri prvom čítaní. Ak chceme čítať viac sektorov po sebe, tak opakujeme program označený od prvej \* so zmenenými údajmi o stope, sektore a strane. Preto je vhodné, aby pri čítaní diskety ste najskôr nastavili pozíciu do základnej polohy a až potom čítali ďalšie sektory.

Následujúci program je ukázkový, ktorý už aj testuje, či sú zatvorené dvierka a či je aj nájdený sektor. Ak dvierka nebudú zatvorené, tak program čaká až to urobíte. Ak bude v mechanike nenaformátovaná disketa, tak program skočí na chybové hlásenie v romke počítača. A znovu pred spustením musíte zadať POKE #247,79 a potom už program spustiť.

Program načíta 1.sektor na 0 stope prvej strany (teda načíta sektor 0 ako READ\*), potom načíta na 33. stope 2. sektor na druhej strane, ďalej zapíše na 2. sektor na 16. stope druhej strany, zase načíta na nulte stope 1. sektor na druhej strane a načíta na 30. stope 5. sektor prvej strany.

Chcem ešte poznamenať, že sektory sa počítajú od 1. jednotky a stopy od nuly!

A ešte niečo. Program skúšajte len na čistej naformátovanej diskete! Lebo tento program nezaujíma či tam máte niečo nahraté a nahrá na disketu to čo je na adrese 50000 a zničí záznam na diskete. Ak bude diske-

ta na kraji prelepená nálepkou, záznam na disketu se neuskutoční.

Programy sú len ukázkové, dajú sa značne zjednodušiť využitím podprogramov v ROM D40.

Program je odladený v assemblery Prométheus, ktorý bol nahratý v pamäti od adresy 24000 a spúšťaný len príkazom (ss+R) RUN. Pokiaľ by ste chceli krokovať po inštrukciách, tak si musíte kúpiť program Devastace 2, ktorý umožňuje i krokovanie v ROM D40.

```
org 41000
ent $
rst 0

di
ld hl, 9104
ld de, 55000
ld bc, 14
ldir
ld a, 201
ld (55000), a
ld hl, 50000
ld a, 5
out (137), a
ld a, 0
out (131), a
ld a, 5
out (137), a
ld a, 0
out (131), a
ld a, 208
out (129), a
call READY
ld a, 1
out (133), a
ld a, 0
out (135), a
ld a, 28
out (129), a
call CAKAJ
ld a, 69
out (137), a
ld ix, 9706
ld a, 136
ld c, 135
ld de, 256
ld h, c
call 55000
call DALSI
ld a, 5
```

```
out (137), a
in a, (133)
dec a
out (133), a
call CAKAJ
xor a
out (137), a
ld hl, 16119
ld (hl), 79
call 5888
ei
ret
```

```
CAKAJ ld bc, 65534
C1 d jnz C1
dec c
jr nz, C1
ret
```

```
DALSI ld a, 2
out (133), a
ld a, 33
out (135), a
ld a, 28
out (129), a
call CAKAJ
ld a, 69
out (137), a
ld ix, 9706,
ld a, 138
ld c, 135
ld de, 256
ld b, c
call 55000
call OBNOV
ld a, 4
out (133), a
ld a, 18
out (135), a
ld a, 28
out (129), a
call CAKAJ
ld a, 69
out (137), a
ld ix, 9709
ld a, 168
ld c, 135
ld de, 256
call 55000
call OBNOV
ld a, 2
out (133), a
ld a, 16
out (135), a
ld a, 28
out (129), a
call CAKAJ
ld a, 69
out (137), a
```

```
ld ix, 9709
ld a, 170
ld c, 135
ld d, 256
ld b, c
call 55000
call OBNOV
ld a, 1
out (133), a
ld a, 0
out (135), a
ld a, 8
out (129), a
call CAKAJ
ld a, 69
out (137), a
ld ix, 9706
ld a, 138
ld c, 135
ld de, 256
ld b, c
call 55000
call OBNOV
ld a, 5
out (133), a
ld a, 30
out (135), a
ld a, 28
out (129), a
call CAKAJ
ld a, 69
out (137), a
ld ix, 9706
ld a, 136
ld c, 135
ld de, 256
ld b, c
call 55000
ret
```

```
OBNOV ld a, 5
out (137), a
in a, (129)
push af
in a, (133)
dec a
out (133), a
pop af
or a
jr nz, CHYBA
ret
```

```
CHYBA call 5888
ei
rst 8
defb 18
```

```
READY ld b, 2
READY3 call READY1
```



```

jr nz, READY3
READY2 call READY1
      jr z, READY2
      -djnz READY3
      ret

```

```

READY1 in a, (129)
      and 2
      ret

      ret

```

A na záver jedna poznámka. Uvedené programy spolupracujú len s mechanikou "A". Keďže tieto informácie o ovládaní zo strojáku som získal len pokusmi, tak ospravedlňte niektoré nezrovnalosti, ktoré uverejnia iní autori, ktorý už dlho vedia ako na to, ale nechcú aby sa to dozvedeli iní.

V dobe, kedy už byl tento příspěvek vysázen, dostal jsem od M. Mošáča další dopis, ve kterém svůj příspěvek ještě dále doplňuje o další poznatky.

Doplňok k upresneniu informácii ovládaniu D40 zo strojáku. Týka sa to dvoch vecí:

1) kód chyby po načítaní sektoru je síce 10, ale hexa kód, teda je to 16 dekadicky, resp. je nastavený 4. bit.

2) to už nieje chyba, ale som zistil, že ak nastavujeme hlavy na nejakú pozíciu je lepšie použiť namiesto čísla 28 číslo 24, je to rýchlejšie nájdený sektor. Ak použijeme číslo 28, tak je treba viac čakať, aby bola nastavená pozícia a nájdený sektor

Uvediem ukážky strojáku, aby ste vedeli, kde treba číslo 28 zmeniť na 24;

```

* ld a,1
  out(133),a
  ld a,17
  out(135),a
  ld a,28 <<-24
  out(129),a
  call CAKAJ
  ld a,69

```

```

out(137),a
ld a,138
* ld ix,9706

```

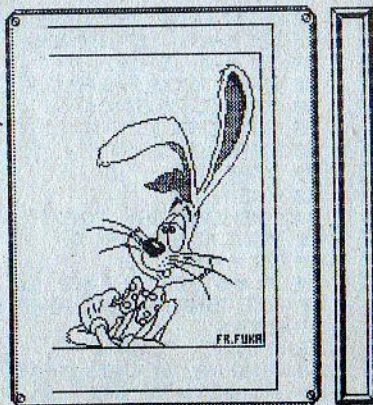
Ukážkový program funguje, takže sa to ani dená považovať za chybu, ale ak použijeme číslo 24, tak sa spoľahlivo načíta sektor aj na 43 stope, pokiaľ bola predtým nastavená iná stopa. Ak použijeme číslo 28 treba čakaciu slučku zväčšiť, aby bol sektor načítateľný. Čakacia slučka stačí ak použijeme kód 24, aj pri posune z multej stopy na 42. pozíciu, avšak ak je kód 28, tak treba dvakrát zavolať čakaciu slučku, aby bol sektor spoľahlivo načítateľný.

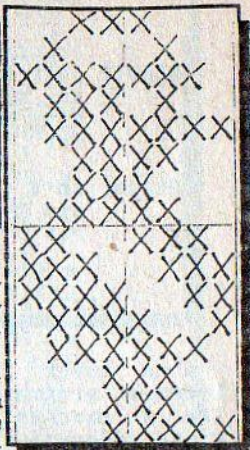
A na záver ešte uvediem stroják, ktorý sa kopíruje na adresu 55000.

```

out(129),a
in a,(129)
ld b,c
and d
jr z,55002
in a,(129)
ld b,c
and d
jr nz,55008
ret

```





Pre čitateľov AP, ktorí si hry vytvárajú v Basicu, by som chcel doporučiť, aby figúry pohybovali po 4 bitoch, vtedy už je pohyb plynulý.

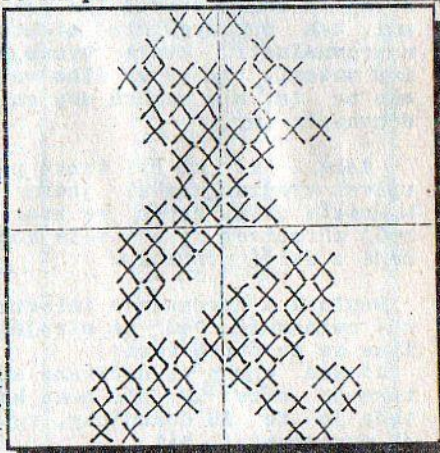
Figurku si nakreslíme do štvorčekov myslenu čiarou rozdelíme na dve polovice a druhý pohyb nakreslíme do dvoch štvorcov s posunom o 4 bity, ako to vidieť na priloženom obrázku.

Pri rozložení pohybu potrebujeme viacej grafiky, ktoré si môžeme vytvoriť zo znakovkej sady a uložiť na adresu ktorú určíte nebudeme potrebovať.

Najskôr si zistíme adresu PEEK v meniacom POKE:

```
10 INPUT "ADRESA";adr
20 PRINT 23607;" ";INT (adr/256)
-1
30 PRINT 23607;" ";adr-256*INT
(adr/256)
```

V prvom programe musí byť vynechaných osem bitov pre medzerník na riadku 15, aby nena-



```
-10 CLEAR 64597: LET ADR=64597
15 LET K=ADR: LET ADR=ADR+8
20 LET A=0: LET B=0
30 INPUT "KOLKO FIGUR? ";X
40 FOR F=33 TO 33+X-1
50 PRINT CHR$ F: " = ";
60 FOR N=ADR TO ADR+7
70 INPUT "DATA: ";D
80 POKE N,D
90 POKE 23607,251: POKE 23606,98
100 PRINT AT B,5,CHR$ F:
110 LET A=A+1
120 IF A=8 THEN LET A=0: LET B=B+1
130 POKE 23606,0: POKE 23607,60
140 NEXT N
150 PRINT " ";ADR+8
160 PRINT #0,"AK CHCES OPRAVIT,STLAC >0<": PAUSE 0
170 IF INKEY#="0" THEN LET B=B+1: GOTO 60
```

stali komplikácie pro písání programu.

V druhom programe je malá ukážka pohybu. Ďalej už záleží na fantázii programátora.

POKE 60000 je len miesto premennej. Takto vytvorený program sa nedá kompilovať, iba ak grafiku urobíme v grafickom móde.

```

180 LET ADR=ADR+8
190 NEXT F
200 PRINT "DLZKA = ",ADR - K + 1
205 REM **NATUKAJ DO IMPUTU V PORADI**
210 DATA 56,68,254,116,127,60,56,124
220 DATA 206,231,243,121,126,24,24,30
230 DATA 3,4,15,7,7,3,3,7
240 DATA 128,64,224,64,240,192,128,192
250 DATA 15,14,12,9,7,30,30,24
260 DATA 32,112,240,240,224,96,52,24
270 DATA 28,34,127,46,254,60,28,62
280 DATA 115,231,207,158,126,56,56,248
290 DATA 1,2,7,2,15,3,1,3
300 DATA 192,32,240,224,224,192,192,224
310 DATA 4,14,15,15,7,27,30,12
320 DATA 240,112,48,144,224,120,120,24
330 REM -----
400 REM **DAJ NEW A PIS PROGRAM**

```



```

1 GO TO 10
2 LOAD "CODE"
3 PAPER 7: INK 0: BORDER 6: CLS
10 PLOT 0,175: DRAW 255,0
15 DRAW 0,-175: DRAW -255,0: DRAW 0,175
20 FOR n=1 TO 30: PRINT AT 10,n," ": NEXT n
40 LET a$="!": LET b$=CHR$ 32+CHR$ 34: LET c$="*": LET d$="22"
45 LET e$="!": LET f$="( ": LET g$=")": LET h$="+",
50 LET y=2: LET x=14
55 GO SUB 200
99 REM **hlavny program**
100 IF x>28 THEN GO TO 115
105 IF INKEY$="p" THEN GO SUB 400
110 IF x<3 THEN GO TO 105
115 IF INKEY$="o" THEN GO SUB 500
150 GO TO 100
199 REM **padanie**
200 GO SUB 1000
220 FOR n=y TO 8
230 PRINT AT n-1,x+1," "
240 PRINT AT n,x,a$:AT n+1,x,b$
245 BEEP 1/10,1/3
250 NEXT n
255 LET y=n-1
260 GO SUB 1100: RETURN
399 REM **doprava**
400 GO SUB 1000
410 IF PEEK 60000-1 THEN GO TO 460
420 PRINT AT y,x,a$:AT y+1,x,b$
425 LET x=x+1
430 POKE 60000,1
435 BEEP 1/100,-1/5
440 GO SUB 1100: RETURN
450 REM -----
460 PRINT AT y,x,c$:AT y+1,x,d$
480 POKE 60000,0
485 BEEP 1/100,1/5
490 GO SUB 1100: RETURN
499 REM **doleva**

```

```

500 GO SUB 1000
510 IF PEEK 60000=1 THEN GO TO 550
520 PRINT AT y,x:es," ".AT y+1,x:fs," "
525 LET x=x-1
530 POKE 60000,1
532 BEEP 1/100,-1/5
540 GO SUB 1100: RETURN
545 REM -----
560 PRINT AT y,x:gs.AT y+1,x:ht
565 BEEP 1/100,1/5
570 POKE 60000,0
580 GO TO 1100: RETURN
999 REM **meniace poke**
1000 POKE 23607,251: POKE 23606,80
1010 RETURN
1100 POKE 23606,0: POKE 23607,60
1110 RETURN
1200 STOP
2000 SAVE "VRTKO" LINE 2: POKE 23736,181: SAVE "VRTKO" LINE 2000,200

```

\*\*\*\*\*  
**POZOR ! POZOR ! POZOR ! POZOR ! POZOR !**

# KLUB 602

knížky

časopisy

manuály

zásuvné paměti

programy

diskety

boxy

časopis AP

PRODEJNÍ DOBA

**PONDĚLÍ  
ZAVŘENO**

**ÚTERÝ AŽ  
ČTVRTEK**

10 <sup>00</sup> -12 <sup>00</sup>
14 <sup>00</sup> -17 <sup>00</sup>

**PÁTEK**

10 <sup>00</sup> -12 <sup>00</sup>
14 <sup>00</sup> -16 <sup>00</sup>

**Martinská 5  
P R A H A I**

# PROGRAMY VLADIMÍRA VOJTY

## TISK MEZI ZNAKY

Při psaní hlaviček do různých tabulek nebo pro přesné vystředění nadpisů, bychom potřebovali někdy tisknout mezi znakové pozice. Tento problém řeší rutina TKM. Ta vytiskne levou polovinu znaku, jehož báze je v reg. DE do pravé poloviny tiskové pozice, dané obsahem reg. HL. Pravou polovinu znaku pak do levé poloviny sousední tiskové pozice.

Způsob použití rutiny TKM je patrný z programu TISKM.

TISKM	call CLS	Smažeme obrazovku
	ld de,TEXT4	Nejprve vytiskneme "normální"
	ld bc,16	způsobem TEXT4:DE-adresa textu,
	call #203C	BC-délka textu, skok do RDN,
	ld de,TEXT5	adresu textu do reg.DE,
	ld hl,#408C	tiskovou pozici do HL (AT 4,12),
	ld c,8	počet znaků do reg.C,
OTK	push hl	Uložíme si pozici tisku,
	push de	adresu tisknutého znaku a
	push hl	ještě jednou pozici tisku,
	ld a,(de)	Uzvedneme kód znaku
	call BAZE	Uvhlédáme jeho bázi do reg.DE
	pop hl	Obnova pozice tisku
	call TKM	Znak vytiskneme
	pop de	Adresa vytisknutého znaku
	inc de	Přejdeme na další znak
	pop hl	Utvoríme novou pozici tisku pro
	inc hl	následující znak
	dec c	Opakujeme pro všechny
	jr nz,OTK	znaky textu
	ret	

Pokud budeme pro tisk znaků používat tiskové pozice AT, lze v rutině TKM nahradit instrukci call DORNHL INSTRUKCÍ inc h. Ponecháme-li však instrukci call DORNHL a tiskovou pozici volíme až na 4. mikrořádek (v našem případě #448c), budeme tisknout mezi znaky i mezi řádky!

TKM	ld b,8	Budeme tisknout 8 bytů znaku
TM	rrd	Přesun (HL) o 4 bity vpravo
	inc hl	Sousední tisková pozice
	rld	Přesun (HL) o 4 bity vlevo
	ld a,(de)	Uzvedneme byt znaku
	rrd	(HL) zpět, tisk pravé poloviny
	rra	Není rotujeme levou polovinu
	rra	tisknutého bytu do pravé polovi-
	rra	ny reg.A. RRD a RLE pracují sou-
	rro	že s pravou polovinou reg.A
	dec hl	Zpět na původní pozici tisku
	rld	(HL) zpět, tisk levé poloviny

inc de	Další byt báze znaku
call DOWNHL	Výpočet nové tiskové pozice
dinz TM	Opakuj osmkrát
ret	

TEXT4	defb 22,3,10	Pozice tisku AT 3,10
	defm "Stare Krecany"	
TEXT5	defm "AP SECOM"	

## ROTACE

Rutiny ROTNAH, ROTDOL, ROTVLV a ROTVPR provádějí rotaci zvolené části obrazovky - okna. Okno definujeme jeho šířkou ve znacích, výškou v bitech a adresou, která závisí na směru rotace okna. Rozměry okna vložíme na adresy RSND+1 a RSLP+1. Nejdříve šířku, na další pak výšku okna. Adresu zvoleného okna zadáme takto:

směr rotace	adresa	HODNOTA
nahoru	ROटनाH+1	adresa levého horního rohu okna
vpravo	ROटनाP+1	- " -
dolu	ROटनाD+1	adresa levého dolního rohu okna
vlevo	ROटनाV+1	adresa pravého horního rohu okna

Jednoduchými úpravami lze rutinami provádět také scrollování okna. Jestliže na adresy MODE nahradíme instrukci jr nc, NULL instrukcí jr NULL, nebo odstraníme obě instrukce na adresách MODE a MOD2, získáme scroll do stran. První způsob používá program pro rotaci textu ROTEX (viz.dále). Použijeme-li druhý způsob úpravy, pak již samozřejmě nelze provádět rotaci!

Scroll nahoru a dolu lze provést několika úpravami. Na adr.13 vložíme místo instrukce ld ir pouze ret. Pro trvalou úpravu pak provedeme následující změny:

```
10 ld de,0
11 -      zrušíme
12 -      zrušíme
13 ret
```

Program DEMO provádí rotaci okna postupně do osmi směrů.

DEMO	ld h1,#200F	Rozměry okna
	ld (RSND+1),h1	
	ld (RSLP+1),h1	
	ld h1,#4056	Adresa pravého horního rohu
	ld (ROTULU+1),h1	
	ld a,#30	#30 - rotace #18 - scroll
	ld (MODE),a	
	call CLS	Smazat obrazovku
	ld b,0	Uytvoření jednoduchého obrázku
	call SCREEN	
	ld h1,ROटनाH	Rotace šikmo vlevo nahoru
	ld de,ROटनाL	
	call ROTACE	
	ld h1,ROटनाV	Rotace vlevo
	ld de,RET	
	call ROTACE	

	ld hl,ROTDOL	Rotace šikmo vlevo dolů
	ld de,ROTULV	
	call ROTACE	
2	ld hl,ROTDOL	Rotace dolů
	ld de,RET	
	call ROTACE	Rotace šikmo vpravo dolů
	ld hl,ROTDOL	
	ld de,ROTUPR	
	call ROTACE	
	ld hl,ROTUPR	Rotace vpravo
	ld de,RET	
	call ROTACE	
	ld hl,ROTUPR	
	ld de,ROTNAH	Rotace šikmo vpravo nahoru
	call ROTACE	
	ld hl,ROTNAH	
	ld de,RET	Rotace nahoru
ROTACE	ld (R1+1),hl	Modifikace
	ld (R2+1),de	
	ld b,160	Počet provedených kroků
	ei	Uvolnit přerušeni
R0	push bc	Uložit počítadlo
	halt	Počkej na přerušeni
R1	call ROTDOL	Proveď první rotaci
R2	call ROTUPR	Proveď druhou rotaci (nebo net)
	call WAIT	Volba rychlosti rotace
	pop bc	Obnov počítadlo
	djnz R0	
RET	ret	
BUFER	equ 23296	Bufery pro uložení prvního mikrořádku, t.j. max.32 bytů
ROTNAH	ld hl,#4048	Adresa levého horního rohu okna
	ld de,DOWNHL	Výpočet následující adresy
	jr NAHDOL	
ROTDOL	ld hl,#47A8	Adresa levého dolního rohu okna
	ld de,UFHL	Výpočet předcházející adresy
NAHDOL	ld (MODI+1),de	Ulož příslušné výpočty
RSND	ld bc,#00	Rozměry okna do reg.BC
I0	ld de,BUFER	Adresa uložení mikrořádku
I1	push de	Ulož adresu buferu
LINE	push bc	Ulož rozměry okna
	push hl	Ulož adresu mikrořádku
	push hl	Ještě jednou
	ld b,0	BC = šířka okna
	ldir	Přenes celý mikrořádek
	pop de	Jeho adresu do reg.DE
	pop hl	a také do reg.HL pro výpočet
MODI	call DOWNHL	adresy dalšího mikrořádku
	pop bc	Obnov rozměry okna
	djnz LINE	Přenes všechny mikrořádky
I2	pop hl	Zde je B=0 => BC=šířka okna
I3	ldir	Vyzvedni adresu buferu a přenes
	ret	mikrořádek,uschovaný v buferu.
		To je vše.
ROTULV	ld hl,#4056	Adresa pravého horního rohu okna
	ld a,#16	r1 (hl)
	ld bc,#2BC6	#2B - dec hl / #06 - set 0,(hl)
	jr ULVP	

ROTUPR	ld	hl, #4048	Adresa levého horního rohu okna
	ld	a, #1E	rr (hl)
	ld	bc, #23FE	#23 - inc hl    #FE - set 7, (hl)
ULUP	ld	(ROT+1), a	Příslušná modifikace rutiny
	ld	a, b	
	ld	(MOD1), a	
	ld	a, c	
	ld	(MOD2+1), a	
RSLP	ld	bc, #00	Rozměry okna
NEXT	push	bc	Ulož rozměry i
	push	hl	adresu mikrořádku
	and	a	Vynuluj CY
ROT	r1	(hl)	Rotace všech bitů mikrořádku ve
MOD1	dec	hl	zvoleném směru
	dec	c	
	jr	nz, ROT	
	pop	hl	Adresa krajního bitu mikrořádku
MODE	jr	nc, NULL	Je-li CY=0, skoč, vlož vždy na
MOD2	set	0, (hl)	vnější okraj bitů
NULL	call	DOWNHL	Adresa dalšího mikrořádku
	pop	bc	Obnov rozměry a
	djnz	NEXT	opakuji pro všechny mikrořádky
	ret		

## KRESLENÍ OBRÁZKŮ PO SLOUPCÍCH

Jde o rutinu, která vykresluje obrázek po sloupcích. Její délka je 65 bytů, je relokovatelná a má dva vstupy: VPRAVO a VLEVO. Vše je jasné z komentáře ve výpisu.

Před použitím rutiny musíme na adresu SCREEN nahrát libovolný obraz. Nemusí to být právě 55000, jak je uvedeno ve výpisu. Musíme však do instrukcí na řádcích 26 a 39 vložit odpovídající hodnotu DIF. Rychlost vykreslování lze ovlivnit přidáním další instrukce halt na řádek č.54, nebo sem vložit čekací smyčku:

```

WAIT    ld  bc,1000
        dec c
        jr  nz,WAIT
        djnz WAIT

```

\*Hodnotu v registru BC zvolíme podle požadované rychlosti

```

001          ent  UPRAVD
002
003 *****
004          Kreslení obrázku po sloupcích
005 *****
006
007 ;Na adresu SCREEN nahrajeme
008 ;obraz
009
010 SCREEN  equ  55000
011 DIF     equ  SCREEN-16384
012
013 UPRAVD  ld   hl,22528      Levý horní roh obrazovky - ATTR
014        ld   a,#23        #23 - inc hl
015        jr   ST

```



016	VLEVO	ld	hl,22528+31	Pravý horní roh obrazovky - ATTR
017		ld	a,#2B	#2B - dec hl
018	ST	ld	(SMER),a	Modifikace rutiny
019		ld	bc,24*256+32	B=24 C=32 : rozměry obrazovky
020		ei		Povolit přerušeni !!
021	DALSI	push	bc	Uložit počet řádků a sloupců
022		push	hl	Uložit adresu sloupce
023				
024	SLPC	push	hl	
025		push	hl	
026		ld	de,DIF	Vypočet adresy atributu v ulože-
027		add	hl,de	ném obraze - reg.HL
028		pop	de	Odpovídající adresa na obrazovce
029		ld	a,(hl)	Přenos atributu
030		ld	(de),a	
031				
032		ld	a,d	Vypočet adresy kresby z adresy
033		rfa		atributu
034		rfa		
035		rfa		
036		and	%01011000	
037		ld	d,a	Výsledek do reg.DE
038				
039		ld	hl,DIF	Vypočet adresy kresby v uloženém
040		add	hl,de	obraze
041		ld	c,8	Přenos osmi bytů "znaku"
042	BYT	ld	a,(hl)	
043		ld	(de),a	
044		inc	h	
045		inc	d	
046		dec	c	
047		jr	nz,BYT	
048		pop	hl	Adresa attr. přeneseného znaku
049		ld	de,3E	Vypočet následující adresy
050		add	hl,de	
051		dinz	SLPC	Čekat pro celý sloupec
052				
053		halt		Počkej na přerušeni
054				
055		pop	hl	Adresa vykresleného sloupce
056	SMER	inc	hl	Přechod na vedlejší sloupec
057		pop	bc	Obnovit rozměry obrazovky
058		dec	c	
059		jr	nz,DALSI	Čekat pro všechny sloupce
060		ret		
061				
062	AOLEN	equ	\$-UPRAVO	Délka celé rutiny (65 bytů)

## ROTACE TEXTU

Program ROTEX ukazuje jedno možné použití scrollu vlevo.

ROTEX	call	CLS	Smazat obrazovku
	ei		Uvolnit přerušeni !
	ld	b,5*32	Uytiskni 5 řádků znaků "x"
	call	SCREEN	
	ld	hl,#081C	Velikost okna
	ld	(RSLP+1),hl	
	ld	hl,#405D	Adresa pravého horního rohu okna
	ld	(ROTULU+1),hl	
	ld	a,#18	Upravíme pro scroll

SCT0	ld (MODE),a	
	ld hl,TEXT3	Adresa rotujícího textu
SCT1	ld a,#EF	Při stisknutí klávesy "0"
	in a,(#FE)	dojde k ukončení činnosti
	rra	
	ret nc	
	ld a,#DF	Po stisknutí klávesy "p" bude
	in a,(#FE)	text tisknut opět od začátku
	rra	
	jr nc,SCT0	
	push hl	Ulož adresu vstupujícího znaku
	push hl	
	ld bc,#1604	Nastav tiskovou pozici do pravě-
	call #00D9	ho konce okna
	pop hl	Obnov adresu znaku
	ld a,(hl)	Uzvedni kód znaku
	and 127	Vynuluj 7.bit
	rst #10	Tisk znaku
	ld b,8	Proveď osmkrát scroll
SCT2	push bc	
	halt	Počkej na přerušeni
	call ROTULU	Scroll o 1 bit vlevo
SPEED	ld bc,#0100	Nastavení rychlosti scrollování
	call OWT	
	pop bc	Obnov čítač kroků
	djnz SCT2	
	pop hl	Obnov adresu znaku
	ld a,(hl)	Uzvedni znak
	inc hl	Ukazatel na příští znak
	ria	Šlo-li o poslední znak textu,
	jr c,SCT0	skoč na začátek textu.
	jr SCT1	Jinak pokračuj dalšími znaky.
TEXT3	defn " AMATERSKY "	Poslední znak textu musí mít
	defn "PROGRAMATOR - "	nastaven 7.bit !
	defn "soukromy a zce"	Program je odložen assemblerem
	defn "la nezavisly "	PROMETHEUS , v jiných překlada-
	defn "casopis pro ama"	čích se nastavení 7.bitu posled-
	defn "terske programa"	ního znaku provede taktio:
	defn "tory na pocita"	
	defn "cich ZX Spectru"	defb " "+128
	defn "m,Didaktik a "	
	defn "kompatibilnych."	

## ZVĚTŠENÝ ZNAK

Rutina PRT4 provádí čtyřnásobné zvětšení znaku. V kombinaci s rutinou TKM jí lze psát různé titulky.

Před vstupem do rutiny PRT4 vložíme do reg. DE adresu báze znaku, tiskovou pozici do reg.HL. Program TITL vytiskne nápis, jeho poslední znak musí být incertován, tj. musí mít nastaven 7. bit.

TITL	ld hl,#4028	Pozice tisku AT 1,8
	ld de,NAPIS	Adresa textu
DALZN	push de	Uchováme adresu znaku,
	push hl	tiskovou pozici,
	push hl	ještě jednou.
	ld a,(de)	Uzvedneme znak,
	and 127	vynulujeme 7.bit a
	call BAZE	vyhledáme bázi znaku do reg.DE

	pop hl	Obnovíme pozici tisku.
	call PRT4	provedeme zvětšený tisk.
	pop hl	Upravíme pozici tisku pro
	inc hl	další znak.
	inc hl	
	pop de	Obnovíme adresu znaku.
	ld a,(de)	Opět vyzveaneme znak.
	inc de	upravíme pro příští znak.
	rla	7.bit do CY
	jr nc,DALZN	Nebyl-li to poslední znak,opakuj.
	ret	
PRT4	ld c,8	Počítadlo bytů
PT	ld a,(de)	Vyzveaneme byt báze
	push de	Uložíme jeho adresu
	call ROTDE	Byt převedeme do reg.D a reg.E
	ld (hl),d	Vytiskneme oba byty
	inc l	
	ld (hl),e	
	call DOWNHL	Přejdeme na další mikrořádek
	ld (hl),e	Oba byty vytiskneme ještě
	dec l	jednou
	ld (hl),d	
	call DOWNHL	Další mikrořádek
	pop de	Obnovíme adresu bytu báze,
	inc de	přejdeme na další byt
	dec c	Uže provedeme osakrát
	jr nz,PT	
	ret	
ROTDE	call ROTE	Každý bit reg.A zdvojnásobíme
	ld d,e	polovinu do D, pravou do E.
ROTE	ld b,4	Čtyři bity poloviny bytu
RLA	rla	7.bit do CY
	push af	Uchováme CY
	rl e	CY do reg.E
	pop af	Obnovíme CY
	rl e	Ještě jednou do reg.E
	djnz RLA	Opakujeme čtyřikrát
	ret	
NAPIS	defm "DIDAKTIK"	

Rutiny DOWNHL a UPHL provádí posun adresy na obrazovce o jeden pixel dolu nebo nahoru. Jsou převzaty z příručky fy Proxima "Assembler..."

DOWNHL	inc h	Posun uvnitř šestého řádku
	ld a,h	
	and 7	
	ret nz	
	ld a,1	Posun mezi textovými řádky
	add a,32	uvnitř jedné třetiny obrazovky
	ld l,a	
	ret c	
	ld a,h	Posun mezi třetinami obrazovky
	sub 8	
	ld h,a	
	ret	

UPHL	ld	a,h	Tatáž pro naplnění níže
	dec	h	
	and	7	
	ret	nz	
	ld	a,1	
	sub	32	
	ld	1,a	
	ld	a,h	
	ret	c	
	add	a,8	
	ld	h,a	
	ret		

Pro čekání na stisk libovolné klávesy využijeme rutinu ROM na adrese #1F3D. Na adresu PAUBC můžeme vstoupit s libovolnou hodnotou v reg.BC, která určuje délku čekání (jako v Basicu).

PAUSE0 PAUBC	ld ei res jr	bc,0 5,(iy+1) #1F3D	
CLS	call ld jr	#006B a,2 #1601	Opět využijeme rutinu ROM pro tisk do horní části obrazovky otevřeme 2.kanáli.
SCREEN	ld rst djnz ret	a,"*" #10 SCREEN	Nakreslení jednoduchého obrazku. Počet vytisknutých znaků "a" uložíme do reg.B.
BAZE	ld ld ld add add add add ex ret	de,(#5C36) 1,a h,0 hl,hl hl,hl hl,hl hl,de de,hl	Adresa znakového souboru = 256. Kód znaku uložíme do reg.HL. Uynásobíme osmi. Přičteme a získáme adresu báze "našeho" znaku, přivěšené do DE.
WAIT DWT	ld dec jr djnz ret	bc,#0200 c nz,DWT DWT	Toto je známé řešení spouštění, lze s libovolnou hodnotou v BC vstoupit na adresu DWT. Na rozdíl od PAUBC však není ovládnuto stisknutím klávesy.



# DIGITÁLNÍ

NEZBYTNÝ POMOCNÍK  
MODERNÍHO ČLOVĚKA

## D I Á Ť

Kolik zbytečného času stráví člověk cestou ze školy a do školy, z práce a do práce, čekáním na tramvaj, na vlak, ve frontách na cokoli a podobně? Jak by bylo účelné, kdyby Spectrista měl svého miláčka ve zmenšeném provedení v náprsní kapse a mohl na něm kdykoliv programovat. Kolik času a peněz by se ušetřilo.

Takováto možnost existuje, i když ne tak doslova a někteří nejen Spectristé, ale i Písičkáři ji hojně využívají. Řešení spočívá v miniaturním digitálním diáři s abecedně číslicovou klávesnicí a displejem z tekutých krystalů. Tyto digitální diáře jsou na západě tak samozřejmé jako např. digitální hodinky.

U nás jsou běžně k dostání ve specializovaných prodejnách v cenách asi od 1.000,- Kč výše, tedy asi 40x levněji než přenosné počítače typu "lap top". Do digitálního diáře lze obdobně jako do ZX S ukládat a nebo z něj číst data všeho druhu (poznámky, adresy, tel. seznam) ale také programy, dopisy a nebo např. jednoduché slovníky.

Vtip je v tom, že na rozdíl od ZX S je diář skutečně miniaturní (do kapsy), nepotřebuje monitor ani televizi, magnetofon ani napájecí zdroj (je napájen miniaturní baterií s dlouhou životností).

ZX Spectrum (nebo spíše jeho

náhrada) se z digitálního diáře stává teprve tehdy, když ho naučíme komunikovat s počítačem neboli přenést data ze ZX S do zápisníku a naopak. Pak je možné si třeba při cestování autem, vlakem nebo letadlem na zápisníku připravovat program nebo dopis, který pak doma přeneseme do ZX S.

Elektronický diář obsahuje také terminovník, buzení a mnoho jiných vymožeností. Je to zkrátka dobrý kamarád a rádce, podobně jako ZX S kdo chce, tak s jeho pomocí nejen ušetří čas i peníze, ale mnohemu se také naučí.

K nám se dováží a nejvíce je u nás rozšířen diář CASIO. Počínaje s pamětí 32kB a konče zatím nejsložitějším, ale také nejdražším R20

je lze připojit k ZX S, DG/DM a Kompaktu aj. Prostřednictvím počítače dokáže diář komunikovat i s D40. K tomu potřebujete propojení s počítačem, které lze obdržet na níže uvedené adrese buď jako návod k připojení včetně software a manuálu nebo jako finální výrobek. Musíte však přesně uvést typ diáře CASIO, typ počítače, používané periférie a přiložit zpětnou rádně vyplněnou a ofrankovanou obálku na odpověď.

Jan DREXLER

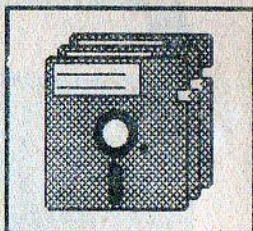
Jahodová 2889 - 106 00 PRAHA 10

POŘIŽTE SI  
ZX SPECTRUM  
DO  
NÁPRSNÍ  
KAPSY
















# SECOM

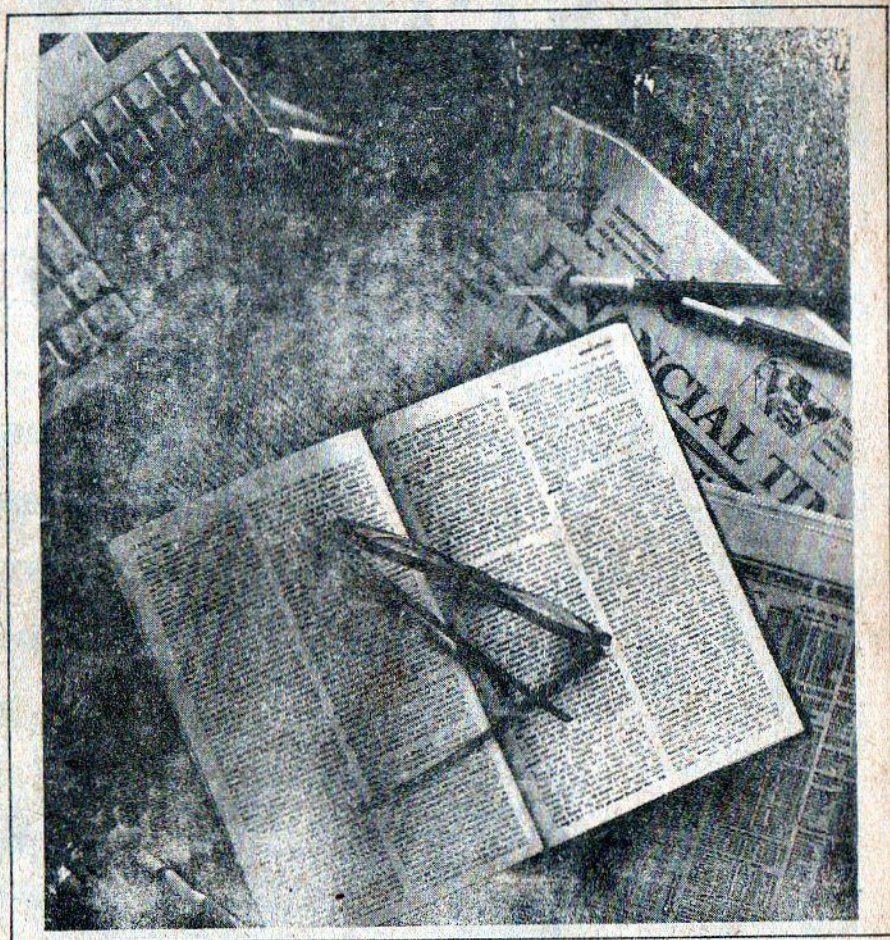
## NABÍDKA PROGRAMŮ NA DISKETÁCH 5.25"



Konečně v tomto čísle přináším přehled programů, které můžete získat za poplatek 20,- Kč za jednu disketu, kterou si naformátovanou na klasický formát sami zašlete. Přiložte rádně vyplněný objednávací listek, který v tomto čísle přiložte. Pokud některý komplet je i ve verzi pro magnetofon, je označen symbolem magnetofonu a pro disketu symbolem diskety.

C - L B - T R A N S - A K T I V N Í	<b>SPD1</b> 	<b>SPD3</b> 	<b>SPD5</b> 	D O M A N N
	ŠDGRAF LINLOM VRH VYPTR0J HYDRLIS PRINTER SUPERCODE	SPRITE 12X8 GEN/UDG GUCH 5xSCREEN\$ ARTIST II R-TEXT/BT	KVADR PREVOD PRINT POPIŠKA	
	<b>SPD2</b> 	<b>SPD4</b> 	<b>SPD6</b> 	
	PÍŠMA DLAŇ48K VAST OBSAH MP/T DISASSM(ALF I TASW /ALF I ASM80 (ALF I) 2xBT DIAGRAMY3 SPECTRAMON SPRITE	VERIFY/BT VERIFY AUTOLINE DEFKEYS LLIST HLAVIČKY PIRÁT MC2 COLT RENAMED ISK TRACE SUPERNUM COMPRES KURZOR OREM	LOGO PASCALL PROLOG FORTHTRAN MEGA-BASIC BETA BASIC3	
	<b>SPD8</b> 		<b>SPD7</b> 	
	COMPILER COMPILER2 HiBasic MCODER		WRITER ARTIST II ARTSTUDIO AW DTEXT /DaS RTEXT GMC TW BT 4GR DTEXT PRT DTEXT D10b DTEXT D100	
				





## AMATÉRSKÝ PROGRAMÁTOR



Soukromý a zcela nezávislý časopis pro amatérské programátory na počítačích ZX SPECTRUM a DIDAKTİK. Vychází každý druhý měsíc v rozsahu 40ti stran.

Cena jednoho čísla je 15,- Kč/16,- sK. Předplatné ve výši 132,- Kč nebo 192,- sK lze zajistit na adrese redakce. Sazba a předlohy stránek byly vyrobeny programem DTP MACHINE za podpory utilit SMGCA / FRAMES a programu TM-FOS2. Nevyžádané rukopisy a jiné příspěvky nevracím. Do tohoto čísla přispěli: D. Wenzel, P. Macek, P. Rak, J. Brossmann, M. Mošat, T. Bartoš, Vl. Vojta. Vydává Petr Černý, Staré Křečany, psč. 407 61. Tiskne soukromá maloofsetová tiskárna SECOM, St. Křečany



Adresa  
redakce



AP-SECOM  
Staré Křečany

Toto číslo vyšlo  
15. května 1993