

amatérský programátor

40 stran

číslo 2. - ročník 4. - březen 1993

ROTACE OBRAZOVÉHO OKNA

rozklad bytu na bity
ve strojovém kódu
ještě jinak

TAPE MONITOR

další výpotky
MUDr. Fr. Stögera

POSTŘEHY

TRAMPOTY S KAZETÁKEM

KURZ PROGRAMOVÁNÍ STROJOVÉHO JAZYKA

TM FOS

knihovna fontů
a semigrafiky
pro TEXT MACHINE

DIDAKTIR
Sinclair
D-40

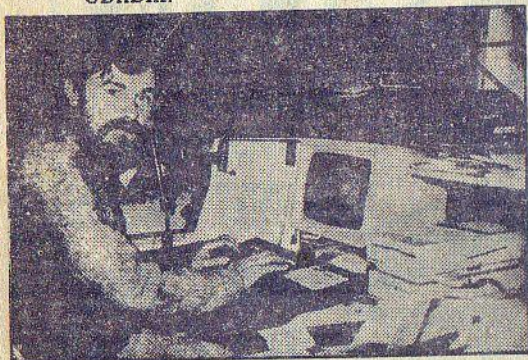
CENA 15.- Kč



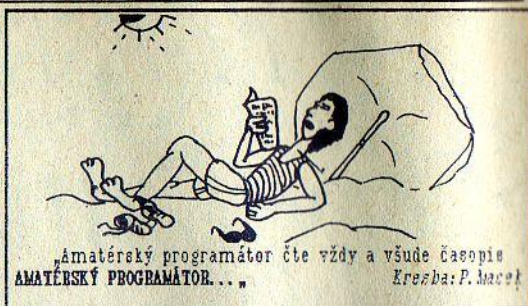


Jak to všechno začalo

V květnu letošního roku to bude právě tři roky kdy vyšlo první číslo tohoto časopisu, tehdy ještě pod názvem „BÁJIT“. Protože ve stejné době začal vydávat kolega Zajíček stejnojmenný časopis ale s orientací na počítače řady PC, byl náš časopis přejmenován na „Amatérský programátor“, což vlastně tehdy předurčilo jeho osud. Vznikl tak časopis, který pravidelně přinášel rady a informace z oblasti amatérského programování. Tedy nic pro profíky. Přesto si ho i oni oblíbili.



Od počátku se časopis připravuje výhradně na počítači ZX Spectrum a bylo přitom vyzkoušeno několik textových editorů.



Bylo to období těsně po tzv. revoluci. Republika byla plná originálních počítačů Spectrum dovezených z tehdy ještě těžko dostupného „západu“. Proto byly veškeré dostupné materiály převážně v němčině nebo v angličtině. Publikací, ze kterých by programátor amatér mohl čerpat bylo zoufale málo. To byl tedy hlavní důvod začít vydávat časopis, který by pomáhal seznamovat nové majitele s jejich počítačem.

Vzal jsem tedy svých poctivě uspořádaných 3.000 Kčs a zakoupil starší cyklostil, blány, barvy a dal se do výroby prvního čísla. Tehdy jsem ještě netušil do čeho se to vlastně pouštím. Celý den v obchodě jako prodáváč a večer, o sobotách a nedělích u počítače. To trvalo až do doby, než mě můj poslední zaměstnavatel propustil jako „nadbytečnou pracovní sílu“. A tak jsem to risknul. Rozhodl jsem se, že si vydávání AP vezmu jako svou celodenní podnikatelskou činnost.

Zájem o AP utěšeně rostl a tak bylo nutné uvažovat o dalším zkvalitňování jeho obsahu ale i tisku, který byl po celou dobu největším problémem. Z podnikatelského hlediska je mou největší chybou asi ten fakt, že netoužím zbohatnout a stát se milionářem. Proto nízké nasazené ceny všeho co jsem produkoval, mně asi přivedli do dnešní svízelné finanční situace. Podnikatelského ducha asi prostě nemám. S tím už jsem se smířil. Práce mě však baví a budu tedy ve vydávání AP pokračovat tak dlouho, jak to bude možné.



Zánovní ofsetové tiskařské stroje od firmy Xerox již umožňují zveřejňovat dodané předlohy a dokonce i fotografie. Současně lze již dosáhnout i vyššího nákladu a kvalitnější tisk.

Nejobtížnější pro každého vydavatele je zjistit o co mají jeho čtenáři největší zájem. Nelze totiž vyhovět všem stejně a k úplné spokojenosti. K tomu je nutná úzká spolupráce čtenářů s vydavatelem. Proto jsem rád, že právě ti moji čtenáři jsou velice trpěliví a poměrně hodně aktivní. Na stránkách AP už publikovalo velké množství autorů z řad čtenářů. Připomenu například Vladimíra Neckáře, Ondřeje Mihulu, Vladimíra Křepinského či Františka Slámu, který po dlouhý čas vedl rubriku „Poznáвам svůj počítač“. Ze současných autorů je to především Pavel Macek se svým oblíbeným seriálem „Kurz programování strojového jazyka“. Častým přispěvatelem je i Vladimír Vojta, který vždy najde v AP něco zajímavého a jeho reakci vznikají úpravy stávajících programů nebo úplně nové. Na stránkách AP se také často setkáváme se jménem Jiřího Brossmana, Eugena Beczce a v poslední době i Jiřího Drexlera.

Z toho všeho tedy plyne, že časopis

Amatérský programátor dává prostor amatérům i profesionálům prezentovat své poznatky a zkušenosti.

jak je to dnes

V současné době je na trhu poměrně slušné množství časopisů, které se věnují oblasti osmibitových počítačů. Snad až na ZX Magazin se tato periodika více zaměřují spíše na hry. Přesto, že nabídka počítačů řady PC je velmi široká, obyčejný smrtelník si nemůže dovolit zakoupit počítač řádově za několik desítek tisíc. A tak zájem o malé domácí počítače stále trvá. Dnes už se Vám asi těžko podaří dovést ze zahraničí nějaké to Spectrum a tak pokud nechcete Comodora nebo Atari, máte jedinou možnost - zakoupit si Didaktika ze Skalice. Ale ani to už asi nebude tak lákavé jako dřívě, neboť tento počítač se v současné době vlastně stane zahraničním výrobkem. To se zřejmě promítne do jeho až dosud zatím celkem příznivé ceny (včetně daně z přidané hodnoty).

Pánové Klaus a Mečiar nám vůbec nadělali spoustu problémů, které se jich týkat nebudou. Známa česká a slovenská amatérská tvořivost je citelně ohrožena nedostakem finančních prostředků. Z Vašich dopisů, které v poslední době dostávám je zřejmé, že fandovství amatérského programování upadá a mládež se věnuje více zuřivému bláznění před obrazovkou s joystickem v ruce obdivující při tom potouchlé panduláky, kteří se mydlí hlava nehlava v hrách, které ani dnes už nelze zakoupit právě za levný peníz. Některé firmy už na to přišly a výborně na tom vydělávají.

Co nám tedy zbývá?

To je přeci jasné! Menechte se odradit. Dokažte, že Spectrum umí mnohem více. Vymýšlejte dále užitečné programy a posílejte je ke zveřejnění do Apicka. Tak se nám podaří nejen udržet při životě, ale i dále rozvíjet naše fandovství pro amatérské programování.

Dovoluji si zakončit mottem, které doporučuji k široké propagaci:

„Stejně jako hodinky,
i Spectrum do každé rodinky“

Petr Černý



znak jako grafické okénko

V této kapitole si povíme něco o grafických oknech. Grafické okno představuje v obrazovce jakákoliv plocha omezená nějakou velikostí. Dá se tedy říci, že znak je nejmenší grafické okno o velikosti 8x8 bodů. Naopak grafické okno takové velikosti představuje možnost znak zobrazit. Tvar znaku uložený někde v paměti už umíme v takovém okně zobrazit. Existuje samozřejmě možnost grafiku z okna v obrazovce přenést do paměti. Lze to provádět docela jednoduše následujícím programem: (číslo 1)

Postup je opačný než při zobrazování znaku. Bereme postupně bajty z jednotlivých linek obrazovky a zapisujeme je zasebou do paměti. V našem případě do předem připraveného zásobníku. V takovém zásobníku bude po proběhnutí programu osm čísel, představujících tvar, který byl v okénku na obrazovce. S grafickým oknem se dají provádět stejné kousky jako z celou obrazovkou. Lze jej tedy vymazat, obarvit, zaplnit, rolovat a skrolovat.

Možná, že vymazat, obarvit a zaplnit bychom už okně o velikosti znaku uměli, ale co to je rolovat a skrolovat?

Je to takové uvedení okna do pohybu. Můžeme si ho ostatně demonstrovat na jednoduchém programu (číslo 2)

Po překladu tohoto programu do strojáku si zkuste i několikrát za sebou program zpusit od návěští ZRT. ALE POZOR! Musíme mít na pozici v levém horním rohu obrazovky (sloupec a řádka nula) zobrazen nějaký znak, jinak nic neuvídíte. Rolování se totiž provádí s již zobrazeným znakem. Pokud je vše v pořádku, budeme moci pozorovat jak znak na této pozici se postupně zasouvá doleva a opět se vysouvá zprava - roluje. Pokud chcete rolovat na opačnou stranu, stačí dát za instrukci RLA instrukci RRA a za RL (HL) dáme RR (HL). Nebo zrušte instrukci RLA a nahraďte ji OR A, potom budete svědky efektu skrolování. Znak se postupně zasouvá doleva až zcela zmizí. Je však možné provádět rolování a skrolování i v jiných směrech. Zkuste si například další krátký program: (číslo 3.)

Tentokrát se bude znak posouvat směrem nahoru a vysouvat se opět zdola. Pokud instrukci LD C,(HL) nahradíte instrukcí LD C, 0 bude se znak zasouvat postupně vzhůru až celý zmizí (skroll). Program je popsán podrobně a myslím, že je možné pokusit se odvodit podobný program, který by uměl totéž směrem dolů. Pokud se o to. Stačí jen přemísťovat bajt z vyšší linky v obrazovce do nižší linky.



VYSYPALY SE MI
PŘI ROTACI BITY Z POČÍTAČE.

```

VEZMI    ld hl,16384      ;připrav adresu v obrazovce
         ld de,ZASOB     ;take adresu zasobniku
         ld b,8          ;opakování pro osm bajtu
VEM1     ld a,(hl)       ;vezmi bajt z obrazovky
         ld (de),a       ;ulož ho do zasobniku
         inc de          ;další adresu v zasobniku
         inc h           ;další linka na obrazovce
         djnz VEM1       ;opakování podle registru B
         ret             ;navrát
ZASOB    defs 8          ;vytvorení prostoru pro zasobník

```

(program c. 2)

```

ZRT      ld hl,16384     ;připrav adresu obrazovky
ROTL     ld b,8          ;pro osm bajtu tvaru znaku
ROL1     ld a,(hl)       ;vezmi bajt z obrazovky
         rla             ;rotuj ho doleva v registru A
         rl (hl)         ;rotuj doleva bajt v obrazovce
         inc h           ;další linka v obrazovce
         djnz ROL1       ;opakuji podle registru B
         ret             ;navrát

```

(program c. 3)

```

ZRH      ld hl,16384     ;připrav adresu v obrazovce
ROTH     push hl         ;uschovej jí
         pop de          ;vyjmi jí do reg. paru DE
         inc d           ;zvětší na další linku
         ld c,(hl)       ;vezmi bajt z horní linky
         ld b,7          ;do registru B počet opakování
ROL2     ld a,(de)        ;vezmi bajt z níže linky
         ld (hl),a       ;dej ho vyšší
         inc h           ;další linka pro HL
         inc d           ;další linka pro DE
         djnz ROL2       ;opakuji podle registru B
         ld (hl),c       ;vrchní bajt dospodu
         ret             ;navrát

```

zobrazování textů

Dostáváme se k tomu nejdůležitějšímu, k čemu jsme se všemi teoriemi okolo znaku prokousávali - totiž k zobrazování textů.

Vytvoříme si pomocí programků, které jsme si už předvedli takové zobrazování textů, které by bylo co nejvíce univerzální.

Text se v assembleru zapisuje pseudoinstrukcí (instrukce použitelná jen uvnitř konkrétního assembleru) DEFM a text je za ní uveden v uvozovkách. Tak se to např. uvádí u assembleru PROMETHEUS a GENS 3.

U assembleru MRS. je taková malá výjimka. Text se zde zapisuje pomocí pseudoinstrukce DB (definice bajtu) a text je za ní uveden v apostrofech. My si však nejprve definujeme vlastní systém v psaní textů. Navrhují, aby každý text začínal několika důležitými číselnými údaji:

1. řádka pozice naniž chceme text psát
2. sloupec pozice odkud chceme text psát.
3. barva celého textu
4. délka textu (maximálně může být 32)

Za těmito údaji už budou následovat znaky textu. Jak bude tyto údaje náš program zpracovávat?

Musíme ho to naučit. Nejprve si připravíme do registrového páru DE adresu textu. Vezmeme nyní první bajt z této adresy. Víme, že je to údaj o řádce pozice a proto jej vložíme do proměnné parametru řádky. Zvětšíme reg. pár DE na další bajt, který pak musíme vložit do proměnné parametru sloupce. Další bajt na další adrese v reg. páru DE představuje bajt barevných vlastností, který musíme hned rozložit v řádce tolikrát, kolik znaků má text. To se dozvídáme z další adresy, na které reg. pár DE zvětšíme a bajt na této adrese je počet znaků textu. Tento údaj použijeme ještě pro vlastní zobrazování textu, abychom věděli kolikrát zobrazit znak. Během zobrazování znaků textu zvětšujeme proměnnou parametru sloupce, aby vždy další znak se zobrazil do dalšího sloupce. Jinak

bychom psali všechny znaky jeden na druhý na jediné pozici. V programu používáme prográmek ASCR k hledání adres tvarů znaku podle jejich kódu a prográmek POZC k vypočtení adresy v obrazovce pro každou pozici na níž text zobrazujeme. Jako hlavní zobrazovací prográmek může být použit ten s návěstím ZOBR a nebo UKAZ, který má mnoho možností. Poslouží nám zde také prográmek ADAT, který najde adresu barev na pozici našeho textu, abychom jej mohli obarvit. V programu se také často vyskytují instrukce pro úchovu a obnovu registrů ze zásobníku (PUSH a POP). Je tomu tak proto, že některé prográmky používají těchto registrů při své práci a údaje, které v nich máme a dále ještě potřebujeme, bychom pak ztratili. A teď vlastní program, který je možné si zapsat jako je uvedeno ve výpisu číslo 5.

(program c.5 - zobrazení textu)

TEKST	ld de, TEX1	; adresa textu do reg. paru DE
TESP	ld a, (de)	; údaj o radce do registru A
	inc de	; zvys na dalsi udaj
	ld (RAD), a	; cislo radku do promenne
	ld a, (de)	; udaj o sloupci do registru A
	inc de	; zvys na dalsi udaj
	ld (SLOP), a	; cislo sloupce do promenne
	push de	; uloz registry par DE
	call ADAT	; spocti adresu barevne pozice
	pop de	; vrat reg. par DE
	ld a, (de)	; udaj o barevných vlastnostech atribut
	inc de	; zvys na dalsi udaj
	ld c, a	; do registru C
	ld a, (de)	; udaj o delce textu
	ld b, a	; do registru B
BARV	ld (hl), c	; barvu do obrazovky
	inc hl	; dalsi adresa v obrazovce na radku
	djnz BARV	; opakuj podle delky textu
	ld a, (de)	; udaj o delce textu
	inc de	; zvys na první znak textu
	ld b, a	; delka text do registru B
TEP1	push bc	; uloz reg. par BC
	ld a, (de)	; znak textu do registru A
	inc de	; zvys na dalsi znak
	push de	; uschovej reg. par DE
	call ASCR	; spocti adresu tvaru znaku
	push de	; uloz J1
call POZC	; spocti adresu v bodove casti obrazovky	

```

pop de          ;vrat adresu tvaru znaku
call ZOBR      ;zobraz znak
ld hl,SLOP     ;priprav promennou sloupce do HL
inc (hl)       ;zvetsi tuto promennou
pop de         ;vrat obsah reg.DE (adresa textu)
pop bc         ;obnov BC reg. par
djnz TEP1     ;opakuji podle delky textu
ret            ;navrat
TEX1          defb 3,6,5,10 ;parametry textu
              defm "Dobry den!" ;text

```

namaluj si bod

Ano. Tak jednoduchou částí malování začneme i naše malování na obrazovce počítače. Jistě si říkáte, že namalovat na obrazovce jeden bod snad ani nemůže být žádný problém. Zkusme si to tedy.

Víme, že jedničkový bit bajtu na obrazovce je zvýrazněn jako bod. Proto některé z Vás může napadnout udělat jednoduše to, co je uvedeno ve výpise číslo 6.

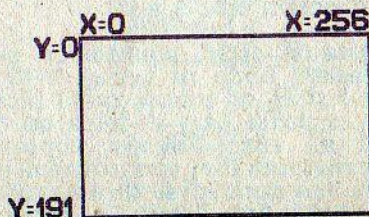
Na obrazovce se skutečně objeví jeden bod. Bajt s hodnotou jedna má jedničkový pouze jeden svůj (nejnižší) bit. Rozjasnil se tedy krajní pravý bit bajtu. V rámci jednoho bajtu lze, jak už víme, zobrazit bodů osm.

Bod bývá zpravidla popisován souřadnicemi ve vodorovném směru X a ve svislém směru Y. Pokud budeme na obrazovce pracovat s body, musíme uvažovat její nejmenší rozlišení tj. 256 bodů v ose X a 192 bodů v ose Y. Budeme muset zase počítat. Pokud je bod vzdálen na ose X 92 bodů, znamená to, že tuto vzdálenost představuje $11 \times 8 = 88 + 4$ body, tedy 11 bajtů a čtvrtý bit dvanáctého bajtu. Když tento bod budeme chtít zobrazit, musíme rozjasnit právě čtvrtý bit ve dvanáctém bajtu.

A právě v této chvíli zvolíme jisté zjednodušení. Pevná paměť ROM v počítači obsahuje mimo tvarů znaků také řadu důležitých podprogramů. Využijeme jednoho z nich. Musíme jen vědět s čím do něj

vstupovat a co a kde očekávat na výstupu. Náš program v ROM leží na adrese 8880. Před vstupem do něho, musíme mít v registru C souřadnici X v bodech a v registru A souřadnici Y též v bodech. Program voláme instrukcí CALL 8880. Po jeho ukončení máme v registrovém páru HL adresu bajtu na obrazovce a v registru A číslo bitu, který představuje bod na zadané pozici. Zbývá nám rozmyslet, co se získanými údaji provést, abychom zobrazili jeden bod. V registru A nastavíme levý bit na jedna, vznikne číslo 128. Podle údaje kolikátý bit v bajtu je náš bod, odrotujeme tolikrát bit v registru A. Pak nám zbývá vložit číslo z registru A do adresy, kterou máme v registrovém páru HL (viz. program číslo 7)

Povšimněte si, že nulové souřadnice X a Y se nacházejí v levém horním rohu obrazovky. Z toho je důležité vycházet, pokud volíme pomocí souřadnic polohu bodu:



(program c. 6)

```

BD      ld hl,18442      ;adresa obrazovky
        ld a,1          ;cislo 1 do registru A
        ld (hl),a       ;cislo do obrazovky
        ret

```

(program c. 7)

```

BOD     ld c,8          ;zadej souradnici X
        ld a,8          ;zadej souradnici Y
BDA     call 8880       ;volej podprogram v ROM
        ld b,a         ;kolikaty bit dej registru B
        ld a,128       ;nastav nejvyssi bit v reg. A
        dec b          ;sniz registr B
        inc b          ;zvys registr B
        jr z,BD2       ;Je nulovy,odskoc na zobrazeni
BD1     rrca           ;rotuj registr A pres CARRY
        djnz BD1       ;opakuji podle registru B
BD2     or (hl)         ;smickej s obrazovkou
        ld (hl),a      ;vloz do obrazovky
        ret            ;navrat

```

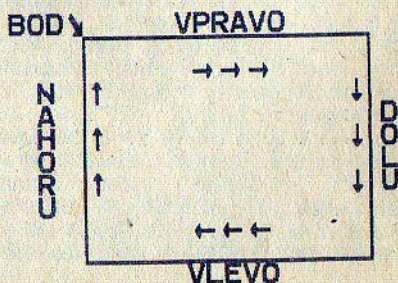
Od malování bodu je už jen malý krůček k malování čar. Jak bychom měli znát ještě ze školních lavic, čára je množinou bodů. Skutečně také program pro malování čáry bude používat program pro malování bodu. Bude postupně z bodů skládat celou čáru. Urcíme si pro začátek bod na pozici $X=10$ a $Y=10$. Čáru budeme malovat od tohoto bodu směrem doprava. Je to důležité, protože tak budeme měnit jednu souřadnici, kterou vlastně určíme směr. Jistě každý pochopil, že budeme zvětšovat postupně souřadnici X a malovat jeden bod za druhým (viz. program číslo 8).

Vytvořme si podprogramy také pro další směry. Doleva to bude snižování souřadnice X. Dolů zase zvyšování souřadnice Y. Nahoru budeme souřadnici Y naopak snižovat. (viz. program číslo 9. a 10)

Zkuste si zadat souřadnice pro vykreslení bodu na pozici $X=170$ a $Y=95$. Potom od tohoto bodu zkoušejte malovat čáry o délce asi 60 bodů na různé strany. Doposud jsme malovali čáry vodorovné nebo svislé. Lze namalovat tímto způsobem také jiné čáry? Po bodech jsme schopni namalovat jakoukoliv čáru, pokud víme jakým způsobem rozmísťovat body.

Zkusme např. zvětšovat souřadnici X a Y najednou při kreslení čáry. Ta bude směřovat šikmo dolů. Můžeme také zkusit měnit souřadnice tak, že X budeme snižovat (DEC C) a Y zvyšovat (INC A). Zkuste si to na programu číslo 11.

V programech se často vyskytují rámečky textů. Dokázali bychom namalovat pomocí čar čtvereček nebo obdélník? Oba se skládají ze svislých a vodorovných čar. Pokud je budeme malovat od jednoho bodu, použijeme tyto čáry ve všech čtyřech směrech:




```

;----- malovani car -----
CAR1      ld c, 10      ;souradnice X
           ld a, 10      ;souradnice Y
           ld b, 72      ;delka cary v bodech
           call BD3      ;podprogram smeru cary
           ret           ;navrat

;doprava
BD3        push bc
           push af
           call BDA
           pop af
           pop bc
           inc c
           dec b
           jr nz, BD3
           ret

;dolu
BD4        push bc
           push af
           call BDA
           pop af
           pop bc
           inc a
           dec b
           jr nz, BD4
           ret

;nahoru
BD5        push bc      ;uschovani registru
           push af
           call BDA      ;zobrazeni bodu dle souradnic
           pop af        ;obnova registru
           pop bc
           dec a          ;snizeni souradnice Y o 1
           dec b          ;snizeni bodu delky
           jr nz, BD5    ;pokud není delka nulova, opakuj
           ret           ;navrat

;doleva
BD6        push bc
           push af
           call BDA
           pop af
           pop bc
           dec c
           dec b
           jr nz, BD6
           ret

```

Do dalších čísel pro Vás autor seriálu Pavel Macek připravil další pokračování na téma efekty s obrazovkou, titulky, okénka, barevné efekty, oživlá okénka a povídat si také budeme i o klávesnici našeho počítače.



VYČIŠTĚNÍ CELÉ OBRAZOVKY.

```

;----- síkma cara -----
SIKM  ld c,50          ;souradnice X
      ld a,20          ;souradnice Y
      ld b,80          ;delka cary v bodech
SIK1  push bc          ;uchova registru
      push af
      call BDA         ;vykresleni bodu podle souradnic
      pop af           ;obnova registru
      pop bc
      inc a            ;zvetseni souradnice Y
      inc c            ;zvetseni souradnice X
      dec b            ;zmenseni delky cary
      jr nz,SIK1      ;pokud je delka nenulova, opakuj
      ret

```



značkové kazety jako SONY, BASF, TDK a další - vhodnou značku pro svůj MGF si postupem času jistě naleznete sami. Kazety skladujte mimo dosah slunečního a tepelného záření a mimo magnetická pole (nepokládat na TVP, MGF ani do blízkosti zdroje pro počítač). Jede-li tramvaji, metrem nebo elektrickou železnici, nepokládejte tašku s kazetami na zem. Magnetické pole od motoru kazetu sice nesmaže, ale může záznam zeslabit. Pokud skladujete kazety několik let, pak jí asi po 5 letech zkopírujte na jinou kazetu (záznamy totiž slábnou a dochází k prokopírování). Svědomitý uživatel si archivaci každého programu zdvojuje, tzn. že daný program má uložen na

Některé - zejména začínající - uživatele ZX8 trápí nahrávání z kazetového magnetofonu. Pro ně uvádíme několik rad a praktických zkušeností.

Začneme záznamovým materiálem - kompaktními kazetami CC. Nevyplácí se používat levné (např. EMGETON) nebo výprodejní kazety. Ty se totiž vyznačují nehomogenní magn. vrstvou a z toho plynoucími možnými výpadky (tzv. DROP-OUT) signálu. Mimoto taková nekvalitní kazeta dokáže po zpuštění v MGF znečistit páskovou dráhu a hlavy natolik, že bez speciálního vyčištění je pak celý MGF prakticky nepoužitelný. Malá kvalita se projeví i v tom, že na stejné místo na pásku můžeme nahrát nový program jen několikrát po sobě a pak již to vzhledem k opotřebování magn. vrstvy není možné. Z těchto důvodů používejte pouze



dvou různých kazetách. Dojde-li pak k poškození, ztrátě nebo "Tape loading error" na jedné, program snadno obnoví z druhé. Programy často používané (např. kopírky, editory, tiskové programy atd.) si dejte na zvláštní kazetu. Jejich častým nahráváním z archivační kazety by brzo došlo k jejímu opotřebování a později i zničení.

Co se týče magnetofonu, platí totéž co o kazetách. Nedoporučuji tuzemské výrobky (např. Diamant) ani z bývalé RVHP (např. MIRA z bývalé NDR) natož laciné šmejdý ze Západu (např. ELTA nebo Daewoo asi za 900,-Kčs). Jak je známo, např. ELTA z NSR při nahrávání umazával částečně již nahrané záznamy (existovala úprava, kterou se daný jev do jisté míry dal potlačit). Magnetofony je vhodné mít dva: jeden pro vlastní nahrávky s pevné nastavenou rychlostí a rychlostí posuvu, druhý pracovní pro nahrávky cizí, kde je možnost nastavit kolmost hlavy, případně rychlost posuvu pásku. Jako MGF nejsou příliš vhodné walkmany (většinou laciné nekvalitní MGF s malou rezervou v zesílení), ani levné typy (tzv. jezevce) kombinovaný s radiem nebo druhou mechanikou. Pozor na stereofonní kazetové MGF, signál je vhodné odebrat jen z jednoho kanálu (vyzkoušet, ze kterého je to lepší, korektory hloubek a výšek (resp. ekvalizer) mít ve střední poloze, resp. tonovou clonu nastavenou na maximum vysokých tónů.

Cizí nahrávky většinou nemůžeme na svém MGF přečíst z těchto důvodů:

- a) kazeta byla nahrána na mgf s jinou kolmostí hlavy
- b) kazeta byla nahrána na mgf s jinou výškou hlavy
- c) kazeta byla nahrána na mgf s rychlostí, resp. s kolísající rychlostí
- d) hlava nebo pásková dráha jsou znečištěny
- e) špatné nastavení mgf (malá úroveň signálu, nevhodné nastavení korektoru barvy zvuku)

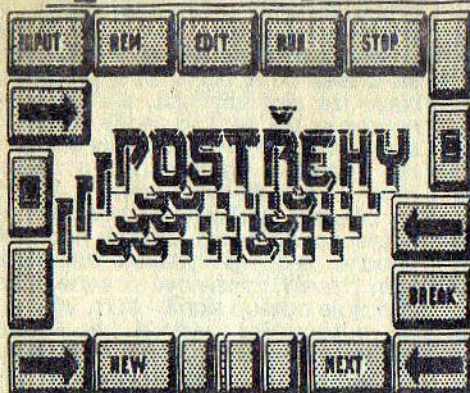
Příčiny a) a b) lze odstranit nastavením kolmosti resp. výšky kombinované hlavy v mgf. Většinou stačí nastavit kolmost hlavy a to šroubkem na držáku hlavy (u některých mgf je v místech hlavy díra v krytu mgf, jinde ji musíme udělat). Při reprodukci opatrně točíme šroubovákem tak, až uslyšíme v reprodukci maximum vysokých tónů. Po skončení nahrávání z cizí kazety však polohu hlavy vrátíme zpět podle vlastních nahrávaných kazet, abychom neměli problémy s jejich nahráváním. Byla-li kazeta nahrána jinou rychlostí nebo je-li záznam rozhoupaný, lze rychlost korigovat trimrem v obvodu regulátoru otáček

motoru mgf. Hlavu nebo páskovou dráhu čistíme chomáčkem vaty na špejli namočeným v lihu. Hlavu prohlédneme, není-li již na konci své životnosti (pásek do ní udělal zářez). Takovou hlavu lze sice zbrousit, ale pokud to neumíme, tak se o to raději nepokoušejte. Při špatném nastavení korigujeme regulátorem hlasitosti a tónové clony (zpravidla na maximum a nebo na předem vyzkoušený bod). Při nahrávání programu na pásek není příliš vhodné, je-li mgf vybaven automatickým řízením zánamové úrovně, která zhoršuje odstup signál - šum. Vhodné je automatiku vyřadit a úroveň záznamu nastavit ručně a nejlépe tak, že záznamový materiál poněkud přebudíme. Nahrávka pak bude podstatně čitelnější.

V mgf se zbytečně nevrtejte, pokud nevíte, kde je přesně kombinovaná hlava, kde její seřizovací šroubek, kde trimr pro nastavení rychlosti posuvu pásku, pokud si nejste jisti, jak odpojit automatiku a pod. V takovém případě se zeptejte někoho povoláního ve vašem okolí nebo v servisu, případně si to přečtěte v odborné literatuře, např. Hoffhans: Magnetofony, jejich údržba a opravy, kterou vydalo SNTL.

Existuje i řada zapojení korigujících signál z mgf např. jeho zesílením. U ZXŠ někdy pomůže i malá finta: zavedení signálu do zdířky MIC namísto do EAR. Existují i zapojení odstraňující nutnost vytahování konektoru Jack ze zdířky EAR při SAVE.

V celku dobré a levné magnetofony vhodné pro ZXŠ a DIDAKTIK vyrábí firma PHILIPS a GRUNDIG. Mají počítadlo (důležité), opatření proti zamotání pásku při poruše a označení COMPUTER COMPATIBLE. Ve značkových prodejnách např. v Praze jsou občas k mání asi za 1500,-Kč. Walkmana sice seženete ve stánku od 100,- Kč výše, ale nižší cena se v takovém případě krutě vymstí promarněným časem a trampotami při nahrávání. Nehleďte na to, že jednoduché walkmany sice umí přehrávat, ale nikoliv nahrávat. Takže až Vám nepůjde nějaký program nahrát bez chyby, zamyslete se predevším nad tím, jestli používáte pro počítač vhodný mgf i kazety a chyby hledejte nejprve u sebe a potom až v počítači.



Isystémové proměnné:

23 561 REPDEL

prodleva od stisku klávesy do repeat. Normálně 35, možno změnit.

23 562 REPPER

rychlost repeat. Normálně 5, možno změnit.

23 606 CHARS

adresa tabulky znaků -256 od SPACE až do. Normálně adresa ukazuje na generátor v ROM (23606 = 0, 23607 = 60) ale změnou těchto proměnných je možno tabulku znaků umístit kamkoliv v paměti - možnost nadefinování vlastních znaků.

23 675 UDG

dtto jako CHARS, ale pro tabulku UDG znaků.

23 608 RASP

délka varovného BEEPu klávesnice. Možno změnit (na kratší při potřebě editovat řádek s chybou).

23 609 PIP

klávesnicový BEEP - délka. Vložením libovolného čísla 0 - 255 si můžeme nastavit délku pípnutí při stisku klávesy.

23 672 VARS

adresa začátku proměnných v RAM.

23 659 DFSZ

počet řádků v komunikační zóně. Při vložení nuly se nesmí použít BREAK a

program se nesmí zastavit chybovým hlášením, jinak se zhroutí systém (jediné východisko je RESET. Možno použít při utajení programů i při jejich odtajnění vložení normální hodnoty 2)

23 670 SEED

systémová proměnná používaná příkazem RANDOMIZE. Vhodné pro vkládání 16-ti bitových čísel do paměti - rozdělí do dvou 8-mi bitových na adresách 23 670 a 23 671.

23 672 FRAMES

trojbytová proměnná je zvětšena každých 20mS o jednotku. Možno použít při měření času (po zapnutí počítače je nulovaná, lze ji však znovu vynulovat kdykoliv v programu) a pro utajení programu tím, že se nahraje i s programem na pásek a po spuštění programu se porovnává s vloženou hodnotou. Pokud byla prodleva mezi nahráním a spuštěním programu, program se smaže. Toto používá firma ULTIMATE např. u COOKIE, PSSST a pod.)

23 692 SCROLL

čítač scroll, počet odskrolovaných řádků než se počítač znovu zeptá scroll?. Možno vložit cokoliv od 1 do 255.

23 613 ERR SP

návratová adresa pro chybové hlášení. Možno použít pro utajení programu:

POKE 23613,0 BREAK zhroutí systém

POKE 23613,82 odpojí BREAK rutinu

POKE 23613,84 odpojí BREAK klávesu

Příkazy RUN, CLEAR, GOSUB, RETURN vrátí do proměnné původní hodnotu a program je možno zastavit.

Postupem:

LET A = PEEK 23613 + 256 + PEEK 23614:

POKE A, 0 : POKE A + 1, 0

se provede RESET při pokusu o BREAK (smaže se program i za RAMTOPem).

23 658 FLAGS 2

vlajky - vložení 8 zapneme CAPS LOCK, 0 jej vypíná. Možno použít pro čtení klávesnice příkazem INKEY\$ pro přepnutí na velká písmena a testovat jen ta.

23 617 MODE

určuje, který kurzor bude na obrazovce. Vložením čísla 2 přepneme kurzor na G-mod, číslem 1

na E-mod a číslem 0 na mod určený proměnnou FLAGS 2 (L nebo C-mod).

2. kanálové informace:

23 570

vložením čísla 10 odpojíme klávesu EDIT, vložením čísla 16 zastavíme listing při ENTER.

23 743

vložením čísla 80 potlačíme tisk názvu programu při nahrávání, číslem 83 jej povolíme. Při vložení čísla 80 směruje vše na tiskárnu, 83 na obrazovku.

23 736

vložením čísla 187 před SAVE se tento příkaz provede bez zprávy Start tape... a bez čekání na stisk klávesy. Možno použít pro programové nahrávání dat bez stisku klávesy při více blocích. Vložení čísla 187 musí být provedeno před každým příkazem SAVE.

3. ostatní:

Příkaz:

PRINT #3 adresuje vše na tiskárnu
PRINT #2 adresuje vše na obrazovku
PRINT #1 a 0 adresuje řádky v editační zóně nepřístupné příkazem AT.

Chybějící příkaz SCROLL je možno nahradit příkazem RANDOMIZEUSR 3190
Chybějící příkaz FREE je možno nahradit PRINT 65535-USR 7962.
BORDER efekty jsou na těchto adresách:

bez zvuku 1314 - 1325

se zvukem 1331, 1333, 1334

Návrat do programu je stiskem SPACE.



PSÁT ČESKY NA POČÍTAČI
NENÍ PŘECE PROBLÉM.

4. řídicí znaky:

CHR\$ 4 = TRUE VIDEO	14 = EXTEND MODE
5 = INV. VIDEO	15 = GRAFIK MODE
6 = CAPS LOCK	16 = INK
7 = EDIT	17 = PAPER
8 = kurzor zpět	18 = FLASH
9 = kurzor vpřed	19 = BRIGHT
10 = kurzor dolů	20 = INVERZE
11 = kurz. nahoru	21 = OVER
12 = DELETE	22 = AT
13 = ENTER	23 = TAB

Při použití některých řídicích znaků v BASICU je nutno příkazem CHR\$ zadat i parametry např.:

PRINT CHR\$ 17; CHR\$ 1; "XXX..."
(modrý PAPER)

PRINT CHR\$ 22; CHR\$ 10; CHR\$ 10; "XXX..."
(je PRINT AT 10, 10)

Řídicí znaky je možno zadat POKE i do příkazu REM, SAVE, ...:

1 SAVE "xxxjmeno" (xxx-rezervace místa)

POKE 23761,22: POKE 23762,10: POKE 23763,10

Potom EDIT, kurzor doprava, DELETE, napsat číslo řádku, kde chceme tento řádek mít a ENTER.

Potom program nahrajeme skokem na tento řádek. Při nahrávání programu do počítače se název napíše na pozici AT 10, 10.

Místo řídicího znaku můžeme do příkazu poukém dát řídicí znaky pro barvu, flash, bright, atd. Musíme však pamatovat, že celková délka názvu i s řídicími znaky nesmí přesáhnout 10 znaků.

S řídicími znaky 8 lze dosáhnout i takovýto žertíků:

1 REM xxxxx - (xxxxx - libovolný znak, rezervace místa)

POKE 23760,8: POKE 23761,8: POKE 23762,8: POKE 23763,8: POKE 23764,230
LIST

Nyní máme na řádku i příkaz NEW, ale po zpuštění programu RUIem se neprovede (je vlastně v REMce)

Takto lze napsat i jakoukoliv zprávu odporující syntaxi:

1 REM XXXX TO KOUKAS NA SYNTAXI!
POKE 23760,8: POKE 23761,8: POKE 23762,8: POKE 23763,8
LIST

Takto lze v rámci jednoho řádku utajit i kousek strojového programu přepsáním např. svým autorstvím.

5. bezhlavičkový LOAD a SAVE:

Pokud chceme nahrávat bezhlavičkově, musíme mít v zaváděcím programu kousek strojového kódu, který nám zabezpečí nahrání zpět do počítače tím, že jej nahrajeme se samospuštěním a v zaváděcím programu odstartujeme strojový kód.

```
LOAD: 55          SCF
      62 255      LD A,255
      221 33 XXX XXX LD IX, na jakou
                        adresu
      17  XXX XXX  LD DE, kolik
                        byte
      205 86 5     CALL 1366
      201          RET
```

```
SAVE: 62 255      LD A,255
      221 33 XXX XXX LD IX, od jaké
                        adresy
      17  XXX XXX  LD DE, kolik
                        byte
      205 194 4    CALL 1218
      201          RET
```

Takto lze z bloku nahrát jen část (např. obrazovku).

6. podprogramy v ROM:

```
BORDER:
  ld a, číslo barvy
  call 8859
  ret (nebo pokračování prog.)

TEST KEY:
  call 654
  call 798
  vyhodnocení příkazem cp x.
  kód stisknuté klávesy je po návratu z
  podprogramu v registru A, pokud
  nebylo nic stisknuto, je v registru 255.

PAUSE:
  ld b,x délka pauzy v 1/50 s
  pau halt
  djnz pau
  Tohoto podprogramu se nesmí po-
  užít v programech, které mění mód
  přerušení !!!
```

(Pokračování příště)



V jedenáctém čísle loňského ročníku vyšel příspěvek Jiřího Brossmanna o rozkladu bytu na bity ve strojovém kódu. Hned 6.ledna t.r. jsem dostal příspěvek Vladimíra Vojty, který je vlastně reakcí na zveřejněné rutiny.

Oba problémy, které řeší ve svém článku „Rozklad bytu . . .“ pan Brossmann, se dají provést rutinami podstatně kratšími. Jedno řešení uvádím ve výpise „Rutina 1“:

Na adresu BYT+1 vložíme číslo, které potřebujeme rozložit na nuly a jednotky. Výsledek je pak uložen v osmi bytech od adresy 23500. Provedeme osumkrát rotaci reg.A vlevo. Sedmý byt je vždy přesunut do CY. Předpokládáme, že má hodnotu 0. Vložíme tedy na adresu v reg. HL nulu. Je-li náš předpoklad správný, CY=0, pak se provede skok na adresu NULA, zvýší se hodnota reg. HL a znova provedeme rotaci. Je-li však CY=1, pak upravíme již uloženou nulu na jedničku. Získáme tak postupně osm ASCII znaků "0","1".

Jestliže místo ASCII znaků požadujeme pouze byty 0, 1, pak stačí změnit byt "0" na adrese ZNAK+1 na hodnotu 0.

Tohoto rozkladu lze také využít k převodu čísla do dvojkové soustavy, což řeší „Rutina 2“:

Pro její činnost si připravíme tento basicový program:

```
10 INPUT A: POKE 23500,A: RANDOMIZE
USR PREVOD: GO TO 10
```

Místo proměnné PREVOD samozřejmě vložíme adresu, na níž si uložíme stro-

jovou rutinu PREVOD. V ní nejprve otevřeme kanál 2 pro tisk do horní části obrazovky. Potom nastavíme pozici tisku AT 21,0. Nyní vložíme do reg. HL adresu, která je o 1 menší, než adresa na níž ukládáme číslo A v basicovém programu. Pro tisk tohoto čísla použijeme totiž rutinu ROM #1A28. Ta provádí tisk čísla, adresovaného registrem HL. Na adrese 29499 musí být „0“! Za číslem ještě vytiskneme mezeru. Nyní následuje již známý rozklad čísla a tisk jednotlivých nul a jednotek. Rutina na adrese #ODFE provádí scroll celé obrazovky.

Rozklad čísla na bity lze také použít pro jednoduché osminásobné zvětšení znaku, což feší „Rutina 3“.

Opět si připravíme basicový program:

```
20 INPUT "ZNAK ";a$: CLS : PRINT a$ : RAN
DOMIZEUSR : GO TO 20
```

Do reg. DE vložíme adresu vytisknutého znaku, jeho zvětšeninu provedeme na pozici AT 0,12 (#580C). Činnost rutiny je již zřejmá z výpisu.

Další program pana Brossmanna zjišťuje hodnotu zvoleného bitu daného čísla. Toto činí „Rutina 4“:

Nejdříve vložíme do reg. A zkoumané číslo, výsledek uložíme na adresu 29500. Nyní vložíme do reg. B pořadí bitu, jehož hodnota nás zajímá (0-7). Provedeme přesný počet rotací daného čísla vpravo, testovaný bit tak „přepadne“ do CY. Opět předpokládáme, že CY=0 a uložíme na adresu v reg. HL nulu. Byl-li náš předpoklad správný, provede se návrat, jinak zvýšíme obsah adresy na 1.

RUTINA 1:

```
ORG 60000
ROZKLAD LD HL, 23500
BYT LD A, 185
LD B, 8
ROTL RLA
ZNAK LD (HL), "0"
JR NC, NULA
INC (HL)
NULA INC HL
DJNZ ROTL
RET
```

RUTINA 2:
PREVOD

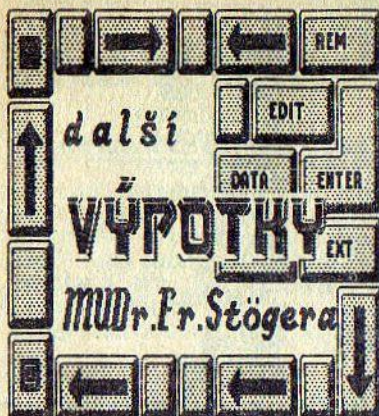
```
LD A, 2
CALL #1601
LD BC, #0321
CALL #0DD9
LD HL, 23500-1
CALL #1A28
LD A, " "
RST #10
LD B, 8
LD C, (HL)
ROTC RL C
LD A, "0"
JR NC, NULL
INC A
RST #10
DJNZ ROTC
JP #ODFE
```

RUTINA 3:
KOPIE

```
LD DE, 16384
LD HL, #580C
LD B, 8
OPAK PUSH BC
LD A, (DE)
LD B, 8
ROT RLA
ATTR1 LD (HL), 0
JR C, JEDNA
ATTR2 LD (HL), 40
JEDNA INC 1
DJNZ ROT
LD BC, 32-8
ADD HL, BC
INC D
POP BC
DJNZ OPAK
RET
```

RUTINA 4:
CISLO
ADR
BIT
ROTA
CHAR

```
LD A, 105
LD HL, 23500
LD B, 3
INC B
RRA
DJNZ ROTA
LD (HL), 0
RET NC
INC (HL)
RET
```



Kreslení drah oběžnic

Program je vypracován na počítači Didaktik Gama a slouží ke kreslení drah planet, komet a pod.

Planety se pohybují kolem Slunce, každá ve své rovině a tyto roviny svírají s rovinou eliptiky (Země) každá úhel θ , sklon dráhy. Sklon drah je malý a proto je můžeme kreslit v rovině papíru.

Kometární dráhy mají obvykle velký sklon. Při 90° je pohyb retrográdní.

Výsledky uvedených programů výpočtu dávají polohy oběžnic na dráze, heliocentrický oběh.

Tělesa obíhají kolem ústředního tělesa, Slunce, po kuželosečkách a Slunce je v ohnisku dráhy.

Dráhy planet jsou elipsy. Dráhy Merkuru a Pluta jest kreslit jako elipsy, ostatní mohou být kresleny jako kružnice. Střed dráhy, \odot Slunce.

Jako jednotky délky a času volíme:

- $k = 0.017\ 202\ 09895$,
- $M_1 = 1$, hmota Slunce,
- $m_2 = 0$, hmoty oběžnic,
- $AU = 1$, astronomická jednotka =
1,495 978 92 E 11m,

• $t = 1$ den

• 1 tropický rok = 365,242 198 78 dní

Úhly se udávají ve stupních a jejich desetinných zlomcích takto: minuty $\times 60$ se sečtou s vtečinami a součet se

vydělí 3600, s/3600.

Počítač ale pracuje s radiány úhlů. Převody:

• radian - (PI=stupeň)/180,

• stupeň - (180=radián)/PI,

Základní směr, je směr k jarnímu bodu Y, obvykle k sobě, a úhly se počítají odtud v kladném směru, proti pohybu hodinových ručiček.

Základní bod na dráze je poloha perihelu. Poloha vychází jako součet délky vzestupného uzlu a argumentu šířky perihelu (obě jsou v tabulkách).

Dráhy planet můžeme většinou kreslit jako kružnice a ani u Merkuru a Pluta velkou chybu neuděláme.

Dráhy parabolickou a hyperbolickou kreslíme podle vypočítaných bodů polohy tělesa.

Pro výpočet polohy středu elipsy vycházíme z geometrických vlastností elipsy a hyperboly, kteréžto vzorce jsou dosti podobné.

U elipsy je v tabulkách udána doba průchodu perihelem, vzestupný uzel, argument šířky perihelu, velká poloosa a číselná excentricita.

Vypočítáme lineární excentricitu le , malou poloosu b a vzdálenost v perihelu q .

Z daných veličin zakreslíme přímku apsid. Určíme výpočtem polohu středu kružnice dráhy: velká poloosa je součet lineární excentricity le a vzdálenosti v perihelu q .

U hyperboly od lineární excentricity odečteme vzdálenost v perihelu q ; v praxi to ale nepotřebujeme, protože hyperbolu a parabolu sestavujeme z vypočítaných bodů.

U paraboly vypočteme přímku apsid a perihel ve vzdálenosti q .

HALLEYOVA KOMETA :

$a = 17.94374 AU$

$e = 0.967277$

$p = 76.01119$ roků

$b = 4.5526655 AU$

$q = 0.5816416 AU$

$le = 17.356306 AU$

$T = 1986, II, 9.39475 EC$

$w = 111.857 st$

$O = 58.15403 st$

$i = 162.2384 st$

Dráhy nejlépe se kreslí na polární papír. Jinak na pěkný bílý papír, třeba typu rys. Vypočítané délky se v příslušném měřítku přenesou na papír.


```

410>PRINT TAB 8;"ELIPTICKA DRAHA"
420 PRINT
430 INPUT "a = ";a;"e = ";e
435 REM Z velke poloosy a excentricity vypocteme linearni e
xcentricitu "le", malou poloosu "b" a vzdalenost v perihelu
"q":
440 LET le = e*a
450 LET q = a - le
460 LET b = SQR (a*a - le*le)
470 PRINT "a = ";a;" AU""b = ";b;" AU""e = ";e""le = "
;le;" AU""q = ";q;" AU"
480 PRINT
490 PRINT TAB 7;"HYPERBOLICKA DRAHA"
491 PRINT
492 PRINT "Kometa Austin 1989 ci:"
500 PRINT
510 INPUT "q = ";q;"e = ";e
520 LET a = q/(e - 1)
530 LET le = a*e
540 LET b = SQR (le*le - a*a)
550 PRINT "a = ";a;" AU""b = ";b;" AU""e = ";e""le = "
;le;" AU""q = ";q;" AU"
560 STOP : PRINT
570 PRINT TAB 7;"PARABOLICKA DRAHA"
580 PRINT
590 PRINT TAB 5;"Pro parabolickou drahu pouze urcime smer o
sy paraboly a polohu perihelu. Dale pocitame jen body polohy
. Pro hyperbolu vlastne obdobne."
600 PRINT "Kometa Austin 1989 ci:"
610 PRINT
620 PRINT "T = 1990, IV, 9.9761 pro epochu 1990, ET"
622 PRINT "e = 1.000380"
624 PRINT "q = 0.349957 AU"
626 PRINT "U = 61.5546 st"
628 PRINT "w = 75.2132 st"
630 PRINT "i = 58.9569 st"
640 REM "Ø a U jsou znaky vzestupneho uzlu. Znaci se velkou
omegou nebo zvlastnim znakem, ktere Didaktik nema."

```

ELIPTICKÁ DRÁHA

Halleyova kometa:

a = 17.94374 AU
b = 4.552734 AU
e = 0.967277
le = 17.356567 AU
q = 0.587173 AU

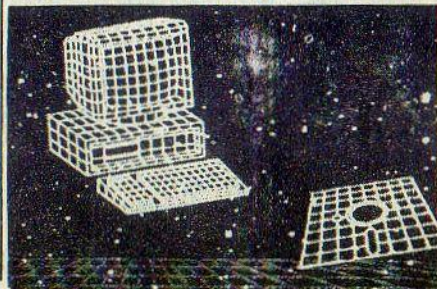
ELIPTICKÁ DRÁHA

Pluto: a = 39.72577 AU
b = 38.428455 AU
e = 0.25347
le = 10.069291 AU
q = 29.656479 AU

HYPERBOLICKÁ DRÁHA

Kometa Austin 1989 ci:

a = 920.93923 AU
b = 25.390957 AU
e = 1.00038
le = 921.28919 AU
q = 0.349957 AU



Čtři anomálie

U eliptické dráhy počítáme se třemi anomáliemi: pravou, střední a excentrickou.

Pravá anomálie "v" je úhel, který svírá průvodič "r" planety s přímkou apsid. Je skutečná a její pomocí můžeme vyjádřit vzdálenost tělesa od Slunce:

$$r = (a \cdot (1 - e \cdot \cos v)) / (1 + e \cdot \cos v)$$

Střední anomálie "M"

je teoretická a je to vlastně pohyb tělesa po kruhové dráze za stejný čas, jako pravá anomálie. Je to úhel, který svírá spojnice myšlené planety se

středem kružnice a s přímkou apsid. Vyjadřuje se jako součin střední anomálie a času.

Excentrická anomálie "E" je opět teoretická. Elipse skutečné dráhy opišeme kružnicí o poloměru "a". Na tuto kružnici promítneme skutečnou polohu tělesa a tento bod spojíme se středem kružnice. Excentrická anomálie je pak úhel této spojnice se středem a s přímkou apsid.

Mezi průvodičem a excentrickou anomálií platí vztah:

$$r = a \cdot (1 - e \cdot \cos E)$$

Mezi pravou a excentrickou anomálií platí vztah:

$$(v/2) = ((1+e)/(1-e)) \cdot \tan(E/2)$$

připravuji do příštího čísla :

Ve chvíli, kdy už jsem *ápětka* tisknul, jsem ještě obdržel další příspěvky od čtenářů. Amaterské programátory, kteří vlastní disketovou jednotku D-40 a již delší dobu si lámou hlavu nad tím, jak ji obsluhovat ze strojáku, jistě potěší podrobný popis jak to udělat, který pro AP připravil Miroslav Mošat, kterého asi znáte jako autora programu "RUN" (redakce ho umístí na diskety SPD) Uveřejním úplný výpis programu v assembleru včetně popisu.

Další příspěvek je od Tibora Bartoše, a řeší pohyb figury (*animace*) pro programátory v Basicu.

```

310>PRINT TAB 10;"TRI ANOMALIE"
320 REM FOR v = 7 TO 255 STEP 10: PLOT v,0: NEXT v
321 PLOT 0,0: DRAW 255,0
330 FOR w = 7 TO 175 STEP 10: PLOT 127,w: NEXT w
340 REM INPUT "a = ";a:"b = ";b
341 LET a = 125: LET b = 70
350 FOR t = 0 TO PI STEP .05
360 PLOT 127 + a * COS t, a * SIN t
370 NEXT t
380 FOR t = 0 TO PI STEP 0.01
390 PLOT 127 + a * COS t, b * SIN t
400 NEXT t
410 LET le = SQR (a * a - b * b)
420 LET e = le / a
430 LET q = a - le
440 LET F = 250 - q
450 PLOT F,0: DRAW -20,55: DRAW 0,38
460 PLOT 209,95: DRAW -82,-95
470 PLOT 245,40: DRAW -120,-40
480 PRINT AT 21,20;"M"
490 PRINT AT 19,19;"E"
500 PRINT AT 21,(F/8);"v"
510 PRINT AT 14,27;"B"
520 PRINT AT 9,26;"B'"
530 PRINT AT 16,29;"B'"

```



Tento program slouží k "monitorování" - zjištění údajů o programech na páse. Takto vzniklé údaje lze vypsat na tiskárně, nahrát na pásek nebo připravit pro kartotéku M-FILE.

TAPE MONITOR umí pracovat pouze s těmi soubory, které jsou nahrané obvyklou rychlostí (1500 Bd). Program zpracovává všechny typy souborů, nemá stanovenou jejich konstantní délku (např. hlavička nemusí mít 17 bytes). Z toho vyplývá, že bezchybná práce programem je podmíněna krátkou prodlevou mezi "tělem" bloku a dalším navigačním signálem. Tuto prodlevu nezachovávají kopírovací programy MICROCOPY a jemu podobné (MISTER COPY, PIRATE). U žádných dalších programů nebyl tento nedostatek zatím zjištěn.

TAPE MONITOR se skládá ze dvou částí - Basic a strojový kód. Do počítače se nahrává LOAD "", sám se rozběhne a vypíše se menu. Pokud budete chtít TAPE MONITOR nahrát na kazetu, program nejprve zastavíte klávesou BREAK a potom zpustíte pomocí RUN 5010. Po nahrání zadáte RUN a program se zpustí, ale budou smazány všechny záznamy. Pokud tedy chcete pokračovat v přerušené práci, pak raději přerušovaný program spustíte GO TO g.

Pro lepší porozumění textu ještě uvedu, že jeden záznam jsou data



TYP	JMENO	start	délka	delprog
P	T.MONITOR+	L1	00000000	405
B	TM-SCR	15432	00000000	
B	TM-MC 17+	52590	00000000	
P	TM-BASIC6+	L5000	00000000	11963

KONEC VYPISU

Stisknete cokoliv

určující jeden blok na pásku (tzn. navigační signál) + bytes, které ho následují, případně "loading error".

z hlavního menu

můžete volat tyto rutiny: "0-MONITOROVÁNÍ PÁSKU" která pouze čte hlavičky bloků, neboli, čte pásek. V horní části obrazovky je signalizován blikajícím čtverečkem stav počítače: WAIT - počítač je v pohotovosti, LEADER - navigační signál (červenomodré pruhy), COMPUTING BYTES - "tělo" programu (modrozluté pruhy).

Získané údaje z pásky se zapisují do paměti počítače (za předchozí) a zároveň na obrazovku.

"1-VÝPIS ZÁZNAMU NA OBRAZOVKU" jsou vypisovány ve formě tabulky. Jsou podobné jako při monitorování, navíc je zde počáteční byte nebo START LINE program (start), celková délka (délka) a délka programu (delprog - pouze u hlavičky prog).

Provést lze i "2-VÝPIS ZÁZNAMU NA TISKÁRNU" nebo provést "3-INSERT ZÁZNAM", který umožňuje zařadit nové záznamy mezi předchozí.

Dále je v nabídce "4-DELETE" pro mazání záznamů. Pro práci s magnetofonem volíte "5-SAVE/LOAD", kde si takto můžete vytvořit kartotéku záznamů jednotlivých kazet. Pokud chcete soubory používat v programu MFILE, zvolíte "6-PŘEPIS PRO MFILE". Pak ještě pokračuje volba "7." pro pokračování v přepisu pro MFILE, volba "8." pro kontrolní výpis pro MFILE a volba "9." pro přičtení souboru pro MFILE.

rozšířená verze

Koncem roku 1986 vznikla nová verze. Je nazvána TMONITOR, a v hlavním menu se představuje jako rozšířená verze. Byla zde pozměněna částečně grafika, některé rutiny byly přidány a rozšířeny o možnost nahrávání i jiných rychlostí než je standart. Proto je i rozšířeno menu o další nabídky: "X-Změna rychlosti nahrávání", dále "A-Analýza systému nahrávání" a "2-Vyhodnocení všech rychlostí".

Program je zařazen do nabídky sítě

TAPE MONITOR

M. RUKY, Q1965, ROZŠÍŘENÁ VERZE

SYSTEM: SINCLAIR-1500Bd

```

... MONITOROVANI PASKU
... Uprizneni zaznamu na obrazovku
... Uprizneni zaznamu na tiskarnu
... UNDISBERT zaznam
... UNLEETE zaznam
... SAVE / LOAD zaznamy
... prepis zaznamu pro MFILE
... predchozi udaje vymazet
... POKRACOVANI V PREPISU PRO MF
... KONTROLNI VUPIS (MFILE)
... Pricteni souboru z MF souboru
... Zmena rychlosti nahravani
... Smazuza systemu nahravani
... Smazeni prebytecne rutiny
  
```

SECOM PUBLIC DOMAIN. Pokud bude o program dostatečný zájem, zařadím do příštího čísla úplný manuál.



LOCAL proměnná (proměnná) ...

Vytvoří se speciální proměnné, které jsou známé jen dané procedúře. Formální parametry jsou pro procedúru automaticky lokální. V procedúře se může vyskytovat více příkazů LOCAL. Má-li být pole lokální, musí být deklarované postupností příkazů: LOCAL p(): DIM p(. .)

LOOP LOOP WHILE podmínka
LOOP UNTIL podmínka

Příkaz ukončuje strukturu DO-LOOP. Podobně jako DO je možné vložit příkazy pro podmíněné ukončení cyklu.

MERGE (mechanismus) "jméno"

Při Microdrive se BBO3 chová pro samostatovací program vtáhnutý přes MERGE korektně.

MOVE

Může na Microdrive přenášet programy i soubory typu CODE.

GO TO

/ GO SUB ON výraz; řádek1, řádek2, ...
ON výraz; příkaz; příkaz; příkaz; ...

1- Skočí se na řádek, jehož číslo je v pořadí na pozici rovné hodnotě výrazu.

2-Vykoná se příkaz, jehož pořadové číslo je rovné hodnotě výrazu.

Pokud je hodnota výrazu mimo počet řádků nebo příkazů, pokračuje se následujícím příkazem nebo řádkem.

ON ERROR číslo řádku

ON ERROR; příkaz; příkaz; ...

1-V případě chyby (mimo OK a STOP Statement) se zavolá podprogram na uvedeném čísle řádku.

2-Vykonají se příkazy na řádku. Když není chyba, přeskočí se.

Vlastní zpracování je vypnuté po dobu běhu programu a obnoví se při RETURN. CONTINUE vrátí program na řádek, který způsobil chybu, ale nezapne se zpracování chyby, takže se nyní vypíše. (Před CONTINUE je třeba vybrat návrat příkazem POP).

ON ERROR 0 vypne vlastní zpracování chyby.

V podprogramu existují tři proměnné:

line - číslo řádku s chybou
stat - pořadové číslo přík. v řádku
error - číslo chyby

Mohou se v programu používat, ale pokud se aktivuje ON ERROR nebo TRACE, budou přepsány.

OVER číslo

Příkaz akceptuje mimo parametrů 0, 1 a 2 jeho účinek je OR. To znamená, že znaky se nepřepisují jeden druhým, ani se nevykonává XOR po pixlech.

Mimo zobrazení bodu je možné na místě X, Y zobrazit i řetězec. Může se jednat o normální řetězec nebo o řetězec vytvořený pomocí GET. X a Y určují levý horní roh řetězce.

Mimo INK, PAPER a pod. je možné také vložit i CSIZE na zvětšení nebo zmenšení řetězce.

Příklad:

```
PLOT CSIZE 32; INK 2; 100,88;"HI!"
```

Výhoda příkazu je ve stejném souřadnicovém systému pro grafiku a znaky.

POKE adresa, řetězec

Mimo čísla 1-255 je možné vkládat i řetězec. Ekvivalent PEEK nahrazuje funkci MEMORY\$(/slicer).

POP (číselná proměnná)

Příkaz vybírá hodnotu ze stacku GO SUB, DO, PROC. Pokud je udaná proměnná, vloží se do ní číslo řádku. Můžeme nyní vyskočit z podprogramů, cyklů a procedur.

Příklad:

```
100 GO SUB 500
110 STOP
500 POP loc
510 PRINT "podprogram volaný z řádku "; loc
520 GO TO loc+1
```

Kdyby na řádku 520 byl příkaz RETURN, objevilo by se chybové hlášení 7 RETURN without GO SUB.

PROC jméno (parametr),(parametr)...

Příkaz volá proceduru. Je to obdoba GO SUB, ale není nutné znát číslo řádku. Klíčové slovo PROC je možné vynechat.

READ LINE (řetězcová proměnná) ...

Příkaz umožňuje číst do proměnných hodnoty bez toho, aby musely být v příkazu DATA uvedeny v " ".

REF reference

REF proměnná

1- Tímto příkazem se zjistí v programu výskyt dané reference.

Reference je proměnná, číslo, postupnost znaků... Po nalezení znaků se přesune daný řádek do editační oblasti s kurzorem za nalezenou referenci. Pokud řádek nechcete opravovat, stisknete klávesu ENTER. Když stlačíte ENTER ještě jednou, hledá se další výskyt reference. Pokud během hledání vložíte nějaký příkaz, hledání se přeruší.

Příklady:

```
REF a$ - hledá se a$
REF sum - hledá se "sum"
REF "SUM" - hledá se SUM
REF 1 - hledá se 1 (včetně neviditelné 5ti bytové formy)
REF "1" - hledá se 1
REF 12*4 - hledá se 12 (včetně...) *4 (včetně...)
REF (a$) - hledá se obsah a$
REF (x) - hledá se obsah x
```

Velké nebo malé znaky nehrají roli.

2- V seznamu formálních parametrů určí proměnné, jejichž hodnota je odevzdaná adresou. Znamená to, že jakákoliv změna obsahu dané proměnné uvnitř procedury bude známá i po jejím ukončení.

RENUM (*) (řádek 1 TO řádek 2) (LINE start) (STEP krok)

Celý, nebo část programu je možné přečíslovat, přesunout úseky, nebo zkopírovat na jiné místo v programu.

a) RENUM - přečíslová program tak, že čísla řádků budou v rozestupech po 10, první číslo je 10.

b) RENUM od TO do - přečíslování úseku.

c) * Starý blok se po přesunutí nesmaže, ale zkopíruje se.

Příklady:

RENUM celý program
RENUM LINE 100 STEP 20 celý program 100, 120, 140 ...

RENUM 100 LINE 300 řádek 100 na 300

RENUM 1540 TO LINE 2000 vše od řádku 1540 se přesune na 2000 a dále.

Příkaz přečísluje všechny reference v GO TO, GO SUB, RESTORE, RUN, ON, ON ERROR, TRACE, LIST, LINE a DELETE. Případná čísla v řádku CLOCK je třeba změnit ručně.

Pokud se stane, že číslo řádku v GO TO, GO SUB je výraz, např.:

100 PRINT : GO TO a

vypíše se zpráva "Failed at..... (1002)

Pozn.: RENUM využívá pro tabulky paměť obrazu. Proto lze zpracovávat i dlouhé programy.

ROLL kód směru (, pixle) (; x , y ; šířka, výška)

Příkaz přesouvá celý nebo jen část obrazu o daný počet pixlů v libovol-

ném směru. To, co z jedné strany vyjde ven, z druhé se objeví (na rozdíl od scroll).

Obraz se posouvá rychleji, pokud se zadá vyšší počet pixlů. Nejrychlejší přesun se provádí ve vodorovném směru o 4 a nebo 8 pixlů, když se využívá přímo funkce v Z80.

kód směru	směr	pohybuje se
1	vlevo	atribut
2	dolu	atribut
3	nahoru	atribut
4	vpravo	atribut
5	vlevo	pixel
6	dolu	pixel
7	nahoru	pixel
8	vpravo	pixel
9	vlevo	současné
10	dolu	současné
11	nahoru	současné
12	vpravo	současné

Pokud se atributy mohou přesouvat jen po osmi pixlech, ignoruje se zadany počet pixlů. V případě přesunu části obrazu se šířka zadává v pozicích jako u znaků (1-32) a výška v pixlech (1-176). Pro přesun v aktuálním okně stačí jednoduchá forma příkazu.

(pokračování příště)



Jasně po uzávěrce jsem obdržel příspěvek od Pavla Raka na téma systémových proměnných. Proto jsem ho ještě zařadil do tohoto čísla za rubriku "Postřehy". Protože jeho příspěvek je ale rozsáhlý, zveřejním dokončení ve třetím čísle AP.

Paměť pro počítač je stejně potřebná jako pro člověka prachy. Čím více, tím lépe. To je nesporné, což mi potvrdí každý z Vás (dovolují si nesouhlasit - pozn. red.). Jak říká jeden z Murphyho zákonů: "Paměti není nikdy dost". Z vlastní zkušenosti to mohou jen dosvědčit. Autor nyní pracuje na Atari Mega STe s 1 MB RAM, což je považováno opravdu za slabé minimum k práci - pozn. autora). Všemi milované SPECTRUM s CPU (Central Processing Unit) Z-80 A vnútri živené 3.5 MHz však může obhospodařovat na jednu 64 Kb paměti. Ať je to RAM (Random Access Memory) či ROM (Read Only Memory), Spectru je to jedno. Ale nám ne, protože my můžeme pro své náročné programy používat pouze RAM. ROMka slouží pouze ke čtení a v případě Spectra je v ní uložen Basic, nutno dodat, že docela dobrý. Jen tak pro zajímavost uvedu, že RAM ve Spectru je paměť dynamická, proto se musí každé 2ms obnovovat, jinak bychom o svá data navždy přišli.

O obnovu dat uvnitř počítače se stará chytrý šváb zvaný ULA. I procesor si na obnovu paměti zřídil jeden registr „R“ (refresh), ale to už bych zacházel do přílišných podrobností. Data do ROMky byly uloženy při výrobě a ty nelze tak jednoduše měnit. Pokud by jste ale měli zájem o změnu ROMky, tak je nutné buď naprogramovat čistou EPROMku nebo tuto svoji smazat ultravioletovým zářením a naprogramovat. To jsou ale věci, které dělají odborníci na slovo vzati. Tuto činnost raději přenechte jim. Obsah ROMky je stálý, tzn., že i po vypnutí proudu tam data zůstanou, což u paměti RAM nehrozí.

Jistě se ptáte, proč procesor může adresovat najednou pouze 65536 bajtů. Je to dáno šífkou jeho adresové sběrnice, která má 16 vývodů a jsou na schématech většinou označovány A0 - A15. Stačí se podívat na obr. 4 v knize „Mikroprocesorové systémy Z80 a I 8085“, na straně 16 a je to hned jasné. Pokud vynásobíte kouzelné číslo 2 na 16 dostanete 65536 což odpovídá našim 64Kb. U Spectra je prvních 16Kb věnováno ROMce a zbylých 48Kb je vyhrazeno RAMce. Podrobnější popis ROMky s příklady, lintami, chybami, voláním . . . uvedu v některém z příštích článků. Majitelé Didaktiků Gama ale namítnou, že oni mají svých 80Kb. Ano mají, ale nesmí zapomenout, že tato paměť není adresována v celku, neboť se zde používá linta s přepínáním paměťových bank po 32Kb. Z toho plyne, že můžou použít buď bank A nebo bank B. Nikoliv současně! Pouze součet dostupné paměti typu RAM dává ono magické číslo 80Kb. Chudák procesor už při této hodnotě jaksi nestíhá, je to nad jeho síly. Výrobce Didaktiků použil na přepínání banků port C0, což poté způsobilo částečnou nekompatibilitu s perifériemi, které na Spectru šlapaly bez problémů, např. tiskárny s připojením přes port C atd.

Dnes se budu věnovat té malé části RAMky od adresy 23552 až po adresu 23733 o které bylo a ještě bude napsáno hodně. Pro začátečníky upřesňuji, že tato oblast je někým (výrobce) a programáto-

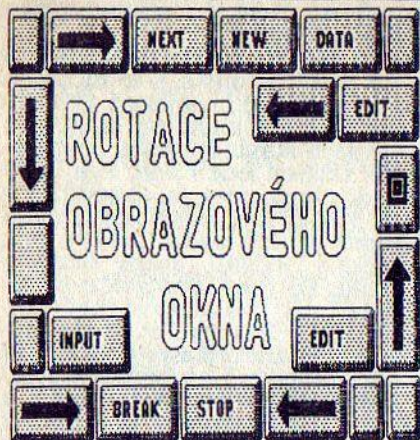
ry) označována jako *systemové proměnné*. Také já zůstanu tomuto označení věrný.

Oblast RAM od adresy 23553 a délky 181 bajtů je místo kde můžeme najít systémové proměnné (SP) počítače. SP obsahují informace o okamžitém stavu systému. Jednotlivé SP zabírají jeden bajt, dva bajty (pro dámy a pány v assembleru *SLOVO* nebo *anglicky "word"* nebo případně i více bajtů. Rekord drží proměnná STRMS, která okupuje 38 bajtů. Po resetu počítače se SP nastaví na standardní hodnoty, které jsou v ROMce. Pokud ale provedete restart příkazem NEW, tak se nám zachovají 3 proměnné; pípní klávesnice - hodnota CLEAR a adresa UDG. Tohle způsobí 3 instrukce na adrese #1bf v ROM. Délka 181 bajtů platí v případě, že nemáte ke svému *gumákovu* připojený Microdrive nebo Interface I. Několik na sobě nezávislých zdrojů tvrdí, že s připojením těchto důmyslných vynálezů se posunuje i adresa konce SP. Dodnes jsem se s těmito zařízeními nesetkal (a snad už *naštěstí* nesetkám - pozn. aut.), proto to nemohu objektivně posoudit. Zjistil jsem, že s připojením Interface I či Microdrive se SP prodlouží o 21 bajtů. Ale asi to tak bude, kdo to ví, ať odpoví . . . nejlépe na adresu redakce, která mu to jistě ráda zveřejní.

Nemusíte se bát, že s připojením *interfejsu* UR-4 nebo D-40 se v paměti něco radikálně změní. Tato zařízení Vám v podstatě rozšíří možnosti ubohého Spectra. Tak například D-40 má v sobě zabudovanou vlastní statickou RAM (protože *statické paměti jsou velmi rychlé, za to ale pěkně hřejou a "zežerou"* o velikosti 2Kb, takže Vám nezabere ani bit z klasické RAMky. Do své statické RAMky si *děčtyřicítka* ukládá všechny potřebné informace o svém provozu. Namátkou vyberu: číslo následujícího snapu, formát diskety (počet stop, sektorů).

SP je oblast v paměti, proto je lze jednoduše číst z Basicu příkazem PEEK (adresa) a měnit příkazem

(Dokončení na straně 39)



Program vykoná pomocou štyroch základných BOI-scroll rutín otáčanie obrazového okna na pohľad spôsobom veľmi atraktívnym.

Riadiaca časť sa nachodí na začiatku programu (40000 - 40160) a vykonáva nastavenie vstupných BOI-adries a ich rozmerov, respektíve volá v zodpovedajúcom poradí scroll rutiny. Rutiny scroll nasledujú v tomto poradí:

HORE	40161 - 40251
DOLE	40252 - 40344
VLAVO	40345 - 40443
VPRAVO	40444 - 40542

Je len prirodzené, že aj samostatne sú rutiny použiteľné.

Pomocou ADDR označíme každú štartovaciu adresu rutín a na ADDR+01 a ADDR+02 je treba nastaviť:

- obrazovú adresu pravého dolného rohu boxu (v prípade konania scroll smerom dole)
- obrazovú adresu ľavého horného rohu boxu (v prípade konania scroll smerom dole a v pravo)
- obrazovú adresu pravého horného rohu boxu (v prípade konania scroll smerom vľavo)

Na ADDR+04 treba nastaviť:

- šírku boxu v charakteroch (v prípade scroll hore a dole)
- výšku boxu v obrazových bodoch (v prípade scroll vpravo a vľavo)

Na ADDR+05 treba nastaviť:

- šírku boxu v charakteroch (pri scroll vpravo a vľavo)
- hodnotu o jeden obrazový bod menšiu ako je výška boxu (v prípade scroll hore a dole)

Najprv však umiestnime do pamäti rutinu (začiatočníci môžu si pomôcť riadkami DATA, pokročilejší assemblerom). Uložme potom nasledujúci program do počítača a spustme ho; program nám ukáže nové možnosti využitia rutiny:

```
1 BORDER 1: PAPER 1: INK 6: CLS
10 FOR I=1 TO 10: FOR J=11 TO 20: PRINT AT I, J:
CHR$(40+I+J): NEXT J: NEXT I
20 RANDOMIZE USR 40000
```

40000	33, 148, 87	LD	HL, 22420
40003	34, 61, 157	LD	(40253), HL
40006	62, 10	LD	A, 10
40008	50, 64, 157	LD	(40256), A
40011	62, 159	LD	A, 159
40013	50, 65, 157	LD	(40257), A
40016	33, 116, 72	LD	HL, 18348
40019	34, 154, 157	LD	(40346), HL
40022	62, 80	LD	A, 80
40024	50, 157, 157	LD	(40349), A
40027	62, 10	LD	A, 10
40029	50, 158, 157	LD	(40350), A
40032	6, 80	LD	B, 80
40034	197	PUSH	BC
40035	205, 60, 157	CALL	40252

40038	205, 153, 157	CALL	40345	40135	34, 154, 157	LD	(40346), HL
40041	193	POP	BC	40138	62, 80	LD	A, 80
40042	16, 246	DJNZ	40034	40140	50, 157, 157	LD	(40349), A
40044	33, 107, 72	LD	HL, 18539	40143	62, 20	LD	A, 20
40047	34, 253, 157	LD	(40445), HL	40145	50, 158, 157	LD	(40350), A
40050	62, 80	LD	A, 80	40148	6, 80	LD	B, 80
40052	50, 0, 158	LD	(40448), A	40150	197	PUSH	BC
40055	62, 20	LD	A, 20	40151	205, 153, 157	CALL	40345
40057	50, 1, 158	LD	(40449), A	40154	193	POP	BC
40060	33, 158, 67	LD	HL, 22430	40155	16, 249	DJNZ	40150
40063	34, 61, 157	LD	(40253), HL	40157	201	RET	
40066	62, 10	LD	A, 10	40158	0	NOP	
40068	50, 64, 157	LD	(40256), A	40159	0	NOP	
40071	62, 79	LD	A, 79	40160	0	NOP	
40073	50, 65, 157	LD	(40257), A	40161	33, 53, 64	LD	HL, 16437
40076	6, 80	LD	B, 80	40164	1, 10, 160	LD	BC, 40970
40078	197	PUSH	BC	40167	197	PUSH	BC
40079	205, 252, 157	CALL	40444	40168	229	PUSH	HL
40082	205, 60, 157	CALL	40252	40169	221, 33, 4, 91	LD	IX, 23300
40085	183	POP	BC	40173	126	LD	A, (HL)
40086	16, 246	DJNZ	40076	40174	221, 119, 0	LD	(IX+0), A
40088	33, 53, 64	LD	HL, 16437	40177	35	INC	HL
40091	34, 226, 156	LD	(40162), HL	40178	221, 35	INC	IX
40094	62, 10	LD	A, 10	40180	13	DEC	C
40096	50, 229, 156	LD	(40165), A	40181	32, 246	JB	NZ, 40173
40099	62, 159	LD	A, 159	40183	225	POP	HL
40101	50, 230, 156	LD	(40166), A	40184	193	POP	BC
40104	33, 53, 64	LD	HL, 16437	40185	197	PUSH	BC
40107	34, 253, 157	LD	(40445), HL	40186	229	PUSH	HL
40110	62, 80	LD	A, 80	40187	124	LD	A, H
40112	50, 0, 158	LD	(40448), A	40188	230, 7	AND	7
40115	62, 10	LD	A, 10	40190	254, 7	CP	7
40117	50, 1, 158	LD	(40449), A	40192	40, 6	JB	Z, 40200
40120	6, 80	LD	B, 80	40194	17, 0, 1	LD	DE, 256
40123	197	PUSH	BC	40197	25	ADD	HL, DE
40126	205, 225, 156	CALL	40161	40198	24, 19	JB	40219
40129	193	CALL	40444	40200	125	LD	A, L
40130	16, 246	POP	BC	40201	230, 224	AND	224
40132	33, 62, 64	DJNZ	40122	40203	254, 224	CP	224
		LD	HL, 16445	40205	32, 6	JB	NZ, 40213

40207	17, 32, 0	LD	DE, 32	40272	32, 246	JR	NZ, 40264
40210	25	ADD	HL, DE	40274	225	POP	HL
40211	24, 6	JR	40219	40275	193	POP	BC
40213	17, 224, 6	LD	DE, 1760	40276	197	PUSH	BC
40216	167	AND	A	40277	229	PUSH	HL
40217	237, 82	SBC	HL, DE	40278	124	LD	A, H
40219	209	POP	DE	40279	230, 7	AND	7
40220	229	PUSH	HL	40281	246, 0	OR	0
40221	126	LD	A, (HL)	40283	40, 6	JR	Z, 40293
40222	18	LD	(DE), A	40285	17, 0, 1	LD	DE, 256
40223	35	INC	HL	40288	167	AND	A
40224	19	INC	DE	40289	237, 82	SBC	HL, DE
40225	13	DEC	C	40291	24, 19	JR	40312
40226	32, 249	JR	NZ, 40221	40293	125	LD	A, L
40228	225	POP	HL	40294	230, 224	AND	224
40229	193	POP	BC	40296	246, 0	OR	0
40230	5	DEC	B	40298	32, 8	JR	NZ, 40308
40231	197	PUSH	BC	40300	17, 32, 0	LD	DE, 32
40232	329	PUSH	HL	40303	167	AND	A
40233	32, 208	JR	NZ, 40187	40304	237, 82	SBC	HL, DE
40235	235	POP	HL	40306	24, 4	JR	40312
40236	193	POP	BC	40308	17, 224, 6	LD	DE, 1760
40237	221, 33, 4, 91	LD	IX, 23300	40311	25	ADD	HL, DE
40241	221, 126, 0	LD	A, (IX+0)	40312	209	POP	DE
40244	119	LD	(HL), A	40313	229	PUSH	HL
40245	35	INC	HL	40314	126	LD	A, (HL)
40246	221, 35	INC	IX	40315	18	LD	(DE), A
40248	13	DEC	C	40316	43	DEC	HL
40249	32, 246	JR	NZ, 40241	40317	27	DEC	C
40251	201	BET		40318	13	DEC	C
40252	33, 158, 87	LD	HL, 22430	40319	32, 249	JR	NZ, 40314
40255	1, 10, 60	LD	BC, 20490	40321	225	POP	HL
40258	197	PUSH	BC	40322	193	POP	BC
40259	229	PUSH	HL	40323	5	DEC	B
40260	221, 33, 4, 91	LD	IX, 23300	40324	197	PUSH	BC
40264	126	LD	A, (HL)	40325	229	PUSH	HL
40265	221, 119, 0	LD	(IX+0), A	40326	32, 206	JR	NZ, 40278
40268	43	DEC	HL	40328	225	POP	HL
40269	221, 35	INC	IX	40329	193	POP	BC
40271	13	DEC	C	40330	221, 33, 4, 91	LD	IX, 23300

40334	221, 126, 0	LD	A, (IX+0)	40413	40, 3	JR	Z, 40418
40337	119	LD	(HL), A	40415	35	INC	H
40338	43	DEC	HL	40416	24, 231	JR	40393
40339	221, 35	INC	IX	40418	125	LD	A, L
40341	13	DEC	C	40419	230, 224	LD	224
40342	32, 246	JR	NZ, 40334	40421	254, 224	CP	224
40344	201	RET		40423	40, 8	JR	Z, 40433
40345	33, 62, 64	LD	HL, 16446	40425	17, 224, 6	LD	DE, 1760
40346	1, 80, 20	LD	BC, 5200	40426	167	AND	A
40351	34, 0, 91	LD	(23296), HL	40428	237, 62	SBC	HL, DE
40354	237, 67, 2, 91	LD	(23298), BC	40431	24, 216	JR	40393
40356	229	PUSH	HL	40433	124	LD	A, H
40359	33, 1, 1	LD	HL, 257	40434	254, 87	CP	87
40362	88	LD	E, B	40436	40, 211	JR	Z, 40393
40363	22, 0	LD	D, 0	40438	17, 32, 0	LD	DE, 32
40365	167, 82	AND	A	40441	25	ADD	HL, DE
40366	237, 82	SBC	HL, DE	40442	24, 205	JR	40393
40368	125	LD	A, L	40444	33, 53, 64	LD	HL, 16437
40369	50, 186, 157	LD	(40378), A	40447	1, 80, 10	LD	BC, 2640
40372	225	POP	HL	40450	34, 0, 91	LD	(23296), HL
40373	229	PUSH	HL	40453	237, 67, 2, 91	LD	(23298), BC
40374	221, 225	POP	IX	40457	0	NOP	
40376	221, 126, 237	LD	A, (IX+237)	40458	0	NOP	
40379	7	RLCA		40459	0	NOP	
40380	0	NOP		40460	0	NOP	
40381	0	NOP		40461	88	LD	E, B
40382	203, 22	RL	(HL)	40462	0	NOP	
40384	43	DEC	HL	40463	0	NOP	
40385	5	DEC	E	40464	0	NOP	
40386	32, 250	JR	NZ, 40362	40465	0	NOP	
40388	42, 0, 91	LD	HL, (23296)	40466	29	DEC	E
40391	24, 15	JR	40406	40467	123	LD	A, E
40393	34, 0, 91	LD	(23296), HL	40468	50, 29, 158	LD	(40477), A
40396	237, 75, 2, 91	LD	BC, (23298)	40471	0	NOP	
40400	13	DEC	C	40472	229	PUSH	HL
40401	200	RET	Z	40473	221, 225	POP	IX
40402	237, 67, 2, 91	LD	(23298), BC	40475	221, 126, 9	LD	A, (IX+9)
40406	24, 221	JR	40373	40478	15	RRCA	
40408	124	LD	A, H	40479	0	NOP	
40409	230, 7	AND	7	40480	0	NOP	
40411	254, 7	CP	7	40481	203, 30	RR	(HL)

40483	35	INC	HL	50011	219, 254	IN	A, (254)
40484	5	DEC	B	50013	230, 31	AND	31
40485	32, 250	JR	NZ, 40481	50015	204, 15	CP	15
40487	42, 0, 91	LD	HL, (23296)	50017	204, 122, 195	CALL	Z, 50042
40490	24, 15	JR	40507	50020	62, 239	LD	A, 239
40492	34, 0, 91	LD	(23296), HL	50022	219, 254	IN	A, (254)
40493	237, 75, 2, 91	LD	BC, (23298)	50024	230, 31	AND	31
40498	13	DEC	C	50026	254, 27	CP	27
40500	200	RET	Z	50028	204, 176, 195	CALL	Z, 50096
40501	237, 67, 2, 91	LD	(23298), BC	50031	62, 127	LD	A, 127
40505	24, 221	JR	40472	50033	219, 254	IN	A, (254)
40507	124	LD	A, H	50035	230, 31	AND	31
40508	230, 7	AND	7	50037	254, 30	CP	30
40510	254, 7	CP	7	50039	200	RET	Z
40512	40, 3	JR	Z, 40517	50040	24, 223	JR	50009
40514	36	JR	H	50042	6, 192	LD	B, 192
40515	24, 231	JR	40492	50044	17, 0, 64	LD	DE, 16384
40517	125	LD	A, L	50047	213	PUSH	DE
40518	230, 224	AND	224	50048	225	POP	HL
40520	254, 224	CP	224	50049	35	INC	HL
40522	40, 8	JR	Z, 40532	50050	197	PUSH	BC
40524	17, 224, 6	LD	DE, 1760	50051	1, 31, 0	LD	BC, 31
40527	167	AND	A	50054	26	LD	A, (DE)
40528	237, 82	SBC	HL, DE	50055	50, 0, 91	LD	(23296), A
40530	24, 216	JR	40492	50056	237, 176	LDIR	
40532	124	LD	A, H	50060	43	DEC	HL
40533	254, 67	CP	87	50061	229	PUSH	HL
40535	40, 211	JR	Z, 40492	50062	17, 33, 92	LD	DE, 23585
40537	17, 32, 0	LD	DE, 32	50065	25	ADD	HL, DE
40540	25	ADD	HL, DE	50066	126	LD	A, (HL)
40541	24, 205	JR	40492	50067	225	POP	HL
				50068	119	LD	(HL), A
				50069	229	PUSH	HL
				50070	17, 33, 92	LD	DE, 23585
				50073	25	ADD	HL, DE
				50074	229	PUSH	HL
				50075	209	POP	HL
				50076	35	INC	HL
				50077	1, 31, 0	LD	BC, 31
				50080	237, 176	LDIR	

Rotácia viacerých obrazoviek

49998	33, 0, 128	LD	HL, 32768
50001	17, 0, 64	LD	DE, 16384
50004	1, 0, 24	LD	BC, 6144
50007	237, 176	LDIR	
50009	62, 247	LD	A, 247

50082	43	DEC	HL
50083	98, 0, 91	LD	A, (23296)
50086	119	LD	(HL), A
50087	225	POP	HL
50088	35	INC	HL
50089	229	PUSH	HL
50090	209	POP	DE
50091	35	INC	HL
50092	193	POP	BC
50093	16, 211	DJNZ	50050
50095	201	RET	
50096	6, 192	LD	B, 192
50098	17, 255, 87	LD	DE, 22527
50101	213	PUSH	DE
50102	225	POP	HL
50103	43	DEC	HL
50104	197	PUSH	BC
50105	1, 31, 0	LD	BC, 31
50108	26	LD	A, (DE)
50109	50, 0, 91	LD	(23296), A
50112	237, 184	LDDR	
50114	35	INC	HL
50115	229	PUSH	HL
50116	17, 95, 92	LD	DE, 23647
50119	25	ADD	HL, DE
50120	126	LD	A, (HL)
50121	225	POP	HL
50122	119	LD	(HL), A
50123	229	PUSH	HL
50124	17, 95, 92	LD	DE, 23647
50127	25	ADD	HL, DE
50128	229	PUSH	HL
50129	209	POP	DE
50130	43	DEC	HL
50131	1, 31, 0	LD	BC, 31
50134	237, 184	LDDR	
50136	35	INC	HL
50137	98, 0, 91	LD	A, (23296)
50140	119	LD	(HL), A
50141	225	POP	HL
50142	43	DEC	HL
50143	229	PUSH	HL
50144	209	POP	DE
50145	43	DEC	HL
50146	193	POP	BC
50147	16, 211	DJNZ	50104
50149	201	RET	

Rutina plynule rotuje viacero spojených obrazoviek. Rutinu je možné použiť k tvorbe programov hier, že pri užití vhodnej grafiky vyvolá taký dojem akoby sme monitorom stáby oknom prehládali pred sebou uloženú krajinu.

Rutinu štartujeme príkazom RANDOMIZEUSR 49998, najprv však je potrebné predom pripravené SCREEN file nahráť z magnetofonu alebo microdrive:

LOAD "CODE 32768 a LOAD "CODE 40000

Na stlačenie klávesy »5« sa dva nahrané obrazy spoja a rotujú vľavo, pri použití klávesy »8« vpravo. Pomocou klávesy »SPACE« je možné z rutiny vystúpiť. Podstata rotácie vľavo je na obrázku A. Pri rotácii vpravo prirodzene opačne.

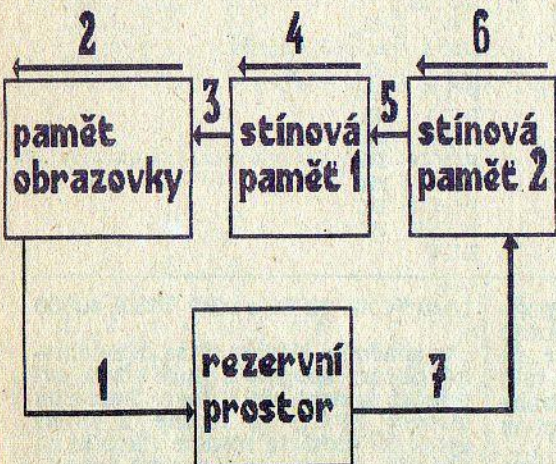
Je možné tiež spojiť viac obrazov, tak keď napríklad využijeme hornú

treťinu obrazovky, môžeme tento počet strojnásobiť. Mechanizmus tohoto pre prípad troch obrazov (viď obr. B). Tu predvedená rutina pohybuje len display file, možno ju ale upraviť aj na pohyb atribútov. Rutina je postavená nasledovne: priestor 49998 - 50040 je pre kontrolu klávesnice, priestor 50042-50095 vykoná doľava a priestor 50096-50149 doprava rotáciu.

(Pre AP z maďarského časopisu "SPECTRUM VILÁG" preložil Eugen H.Becz).



Obr. A: Rotace vlevo



Obr. B: Mechanizmus rotace u třech obrázků

♦ MAJITELÉ SPECTRA A DIDAKTŮ POZOR! ♦

Nabízím program, pomocí kterého budete moci ve svých programech a hrách nabírat obrázky do počítače různými elektrickými způsoby. Podobně jako ve hrách, např.: Transform, Podraz 3, ATV, IKARI a mnohé další. CENA 33,- Kč (v této ceně je započítána kromě samotného programu ještě kazeta a podrobný manuál). Program zašlu postou na dobírku.

♦♦♦ PIŠTE JEŠTĚ DRES! ♦♦♦

První objednavatel obdrží program zdarma.

.....
 : J. Kobližka, Dukelská 662 :
 : 391 02 SEZIMOVŮVŮSTÍ :

TM FOS

to je knihovna
fontů a semigrafiky
pro uživatele
programu
text machine

ve verzi
pro
kazetu i
disketu

obdržíte
dobírkou
u firmy

SECOM

Staré Křečany - poč 407 61

cena 60.- Kčs

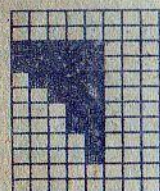


O přednostech a výhodách programu Text Machine jsem již na stránkách AEP psal několikrát, protože s radostí konstatuji skutečnost, že se konečně našel distributor jeho disketové verze. Sám tuto verzi sice ještě nevlastním, ale přesto vám

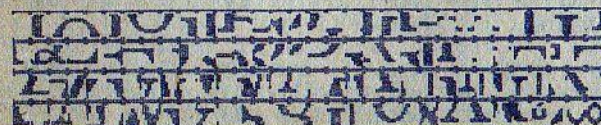
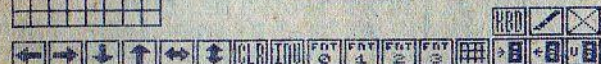
již dnes mohu nabídnout něco, co z tohoto programu udělá skutečný **DESK TOP PUBLISHING**.

Program Text Machine má totiž jednu obrovskou výhodu, kterou prozatím nemá ani tolik propagovaný a opěvovaný (evidentně bezdůvodně) **DESKTOP**. Text Machine totiž umí pracovat se semigrafikou. Ta přináší uživateli mnoho výhod, a tak lze bez jakékoli nadsázky program řadit mezi špičkový a skutečný DTP, tak jak tyto programy známe na počítačích řady PC.

Program je vybaven nejen fontovým editorem, v kterém lze vytvářet vlastní znaky, ale součástí je také semigrafický editor. V něm si uživatel může vytvářet různé obrazce. Nejhlavnější použití semigrafického editoru je ve tvorbě znaků pro psaní titulků, velkých nápisů a pod. Největší znak tak



ABCDEF G H
I J K L M N O P
Q R S T U V W X
Y Z 1 2 3 4 5



může být sestaven ze 72 (D) bodů na výšku a 40 bodů na šířku. Uvažme-li že klasický znak tohoto editoru je velikosti 12 bodů na výšku a 8 bodů na šířku, je to velikost úctyhodně semigrafické znaky se píše běžným způsobem, tj. klávesou příslušného znaku. Pokud si sem. editoru přivlastníte ještě dal

Ší obrazce nebo dokonce i obrázky, můžete je umísťovat na libovolné místo v textu a ten pak nechat kolem něho obtékat, jako zde kolem obrázku křičníku a klávesy ENTER, kterou najdete také ve zmíněné knihovně fontů a semigrafiky TM FOS.

Pokud si obrazec předem umístíte na určité místo a pak zjistíte, že by bylo dobré aby byl o kousek dál nebo bliž, můžete s ním pohybovat stejně jako s běžným fontem, tak jak to vidíte na obrázci vedle šípky vpravo.

Se semigrafikou si můžete doslova a do písmene pohrát jako to vidíte zde na tomto obrázci. Vytvořen byl tak, že obrazec byl po každé přepsání sám sebou vždy posunutim kurzoru na jinou pozici, což působí dojemem, že se jedná o jeden obrazec (celek).

Editor umožňuje použít najednou celkem 4 fonty, z nichž, pokud budeme chtít použít i běžných znaků, si tři můžeme vyhradit pro semigrafiku.

Nejprve si tedy do programu nahrajeme všechny fonty, které budeme chtít používat a pak postupně, či střídavě (podle potřeby) si přihráváme jen rutiny pro zobrazení semigrafiky.

Jestliže tedy máme nyní v programu rutinu pro semigrafiku číslo 2 nemůžeme "psát" semigrafikou číslo 3, 4 nebo 1. Jak by to dopadlo se můžete podívat na ukázce ve vedlejším sloupci.

Font (soubor znakové sady) ve kterém je vytvořena podoba klávesy ENTER je v programu umístěn jako font číslo 2 a "vypíše" se stiskem kláves Q a W. Potom jsem potřeboval do textu vložit obrazec vykřičníku. Nahrál jsem si proto rutinu pro font číslo 3, ve kterém

tento obrazec je vytvořen. Potom jsem ale chtěl znovu "vypsat" obrazec klávesy ENTER.

Bohužel jsem (jako) zapoměl na to, že mám nyní v programu rutinu pro font číslo 3, přepnul jsem do semigrafiky a stiskl klávesy Q a W...nu a výsledek uvidíte na obrázku(?) vpravo.

Vhodnou kombinací několika semigrafik můžete docílit osobité grafické úpravy při tvorbě celé stránky. Ale to už bych vlastně popisoval funkce programu.

Pokud máte nějaké své zajímavé zkušenosti či poznatky s tímto programem, pošlete je ke zveřejnění. Sám jsem totiž měl možnost zjistit u několika uživatelů, že dosud ještě neznají všechny možnosti, které tento kvalitní DTP nabízí. Možná by bylo i prospěšné založit jakýsi klub uživatelů TEXT MACHINE/DTP.

Na zahájení jeho činnosti nabízím tuto první knihovnu fontů a semigrafiky.

Pokud si budete o TMFOS psát, nezapomeňte si poslat vlastní kazetu nebo disketu. Zpět ji obdržíte poštou na dobírku za uvedenou cenu + poštovné.

Přehled fontů a semigrafiky

Knihovna fontů a semigrafiky TM-FOS obsahuje celkem 17 semigrafických fontů, 6 fontů titulkových písmen, které se píší bez semigrafiky pomocí dvou kláves a 7 fontů klasické velikosti.

označení fontů a jejich význam:



příklad : "roboF" - písmeno F označuje font
"roboS" - písmeno S označuje semigrafickou rutinu

pro použití musíme nahrát font (F) a také i semigrafickou rutinu (S). Font je dlouhý 1536.

příklad : "lomsecALL" - označení ALL znamená, že se nahrává více fontů jako jeden blok. V tomto případě o délce 6144, tedy jako čtyři fonty. Proto se volí volba při LOAD FONT ALL



"lomsecS" - písmeno S označuje semigrafiku, v tomto případě pro všechny fonty společnou.

příklad : "patkaCS" písmena CS označují fonty, které se píší klávesou příslušného znaku přímo (levá polovina) a současným stiskem klávesy CAPS SHIFT (pravá polovina)



příklad : "rene" font bez dalšího označení vyjadřuje běžný font (jako např. tento).

Příklad použití fontu označeného jako CS :



stiskem klávesy "a" se napíše a
stiskem klávesy "a" současně s CAPS SHIFT se napíše A a tak vznikne písmeno "A".

U semigrafiky jsou znaky umístěny pod příslušnou klávesou a také se tak píší. Háčky, čárky a kroužky nad písmenem se doplňují dodatečně a bývají u všech těchto fontů umístěny převážně takto:



klávesa "1" = ^ čárka nad "i" nebo "i"
klávesa "2" = ˇ čárka nad ostatními znaky
klávesa "3" = ˆ háček nad znakem
klávesa "4" = ˆ kroužek nad "ó" / "o".
klávesa "5" někdy bývá tečka nebo pomlčka.

- "roboF" a "roboS"

ABCDEF GHIJKL M

NO PQRSTU VWXY

Z / / / / 5
1 2 3 4

- "lomsecALL" a "lomsecS"

font číslo 1:

abcdefghijklmnopqrstu vwxyz

1 2 3 4 5
.....

font číslo 2:

ABCDEFGHIJKLMN OPQRSTUVWXYZ

1 2 3 4 5
.....

font číslo 3:

klávesa "j" = 1 klávesa "d" = 2 klávesa "t" = 3

klávesa "c" = 4 klávesa "p" = 5 klávesa "s" = 6

klávesa "a" = 7 klávesa "o" = 8 klávesa "e" = 9

klávesa "n" = 0
.....

- "RUPZALL" a "RUPZS"

font číslo 1:

abcdefghijklmnopqrstu vwxyz

1= / 2= / 3= / 4= / 5= / (spodní část písmene "j")
.....

font číslo 2:

ABCDEFGHIJKLMN OPQRSTUVWXYZ


1= / 2= / 3= / 4= / 5= / (spodní část písmene "j")
.....

- "inzertF" a "inzertS"

abcdefghijklmnopqr

stuvwxyz

1=  čárka nad "i"

2=  čárka nad "ó, é,"

4=  čárka nad "y"

3=  háček nad "ě, š, č"

5=  čárka nad "ú"

- "knižníALL" a "knižníS"

font číslo 1:

abcdefghijklmnopqrstuvwxy

1 2 3 4 5
.....

font číslo 2:

ABCDEFGHIJKLMN OPQRSTUVWXYZ

1 2 3 4 5
.....

1234567890 psaní číslic je stejné jako u "lomsecALL"

- "patka1CS" (číslo 1 označuje základní výšku)
abcdefghijklmnopqrstuvwxy
ěščřzyáíúůň

- "bigtypeF" a "bigtypeS"

ABCDEFGHIJKLM

NOPQRSTUVWXYZ

Z 1 2 3 4
.....

- "fshadeF" a "fshadeS"

ABCDEFGHIJKLMN OPQR

STUVWXYZ 1 3 4
.....

- "lukeer" a "lukees"

abcdefghijklmnopqr
stuvwxyz 1 2 3 4 5

- "Falstaff" a "Falstaffs"

ABCDEFGHIJKLMNO P
RSTUVWXYZ 1 2 3 4 5

- "doublecs"

abcdefghijklmnopqrstuvwxy z
őščřžáíéóúťň

SS+Y= [] SS+U= [] SS+H= [] SS+O= [] SS+D= [] SS+S= []

- "plastikr" a "plastiks"

ABCDEFGHIJKLMNO P
RSTUVWXYZ 2 3 4

- "classicr" a "classics"

ABCDEFGHIJKLMNO P
RSTUVWXYZ 1 2 3 4

- "blinc3r" a "blinc3s"

ABCDEFGHIJKLMNO P
RSTUVWXYZ 1 2 3 4

• "contraCS"

ABCDEFGHIJKLMN**OP**QRSTU**VW**XYZ

1 2 3 4



• "poster1CS"

ABCDEFGHIJKLMN**OP**QRSTU**VW**XYZ

123456789

• "classicCS"

ABCDEFGHIJKLMN**OP**QRSTU**VW**XYZ

123456789

• "janeALL" a "janes"

font číslo 1:

abcdefghijklmnopqr**stuv**wxyz

1 2 3 4 5

font číslo 2:

ABCDEFGHIJKLMN**OP**QRSTU**VW**XYZ

1 2 3 4 5

font číslo 3:

1234567890 psaní číslic je stejné
jako u "lomseCALL"

• "MINUSKAaF" a "MINUSKAaS"

font číslo: 1

abcdefghijklmnopqr**stuv**wxyz

1 2 3 4

font číslo: 2

a.abcdefghijklmnopqr**stuv**wxyz

1 2 3 4

• "atari"

abcdefghijklmnopqr**stuv**wxyz 01234567890!@#\$%&'()*+,-./:;<=>? ABCDEFGHIJKLMN**OP**QRSTU**VW**XYZ
01234567890!@#\$%&'()*+,-./:;<=>?

• "bombastic"

abcdefghijklmnopqrstuvwxyz 0éščřžýáíéúťňď 1234567
 890 !@#%&'()* _ [] - + = : ; ? / © ABCDEFGHIJKLMNOP
 QRSTUVWXYZ 0ÉŠČŘŽÝÁÍÉÚŤŇĎ

• "bigbold"

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 0 É Š Č Ř Ž Ý Á Í É Ú Ť Ň Ď 1 2 3 4 5 6 7 8 9 0 ! @ # \$ % & ' () _
 < > [] ^ { } | ~ - + = : ; ? / *

• "nova"

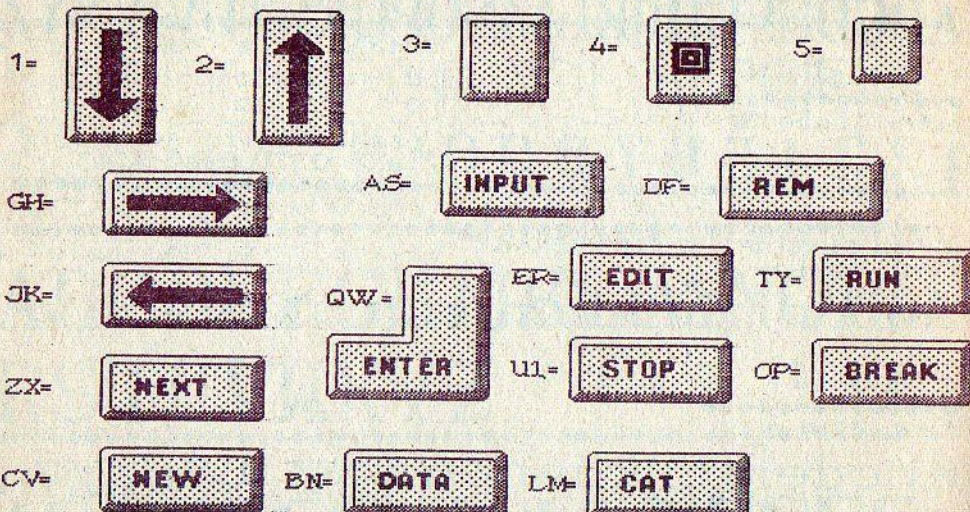
abcdefghijklmnopqrstuvwxyz 0éščřžýáíéúťňď 123456
 7890 !@#%&'()* _ > [] < [] - + = : ; ? / *

• "profit"

abcdefghijklmnopqrstuvwxyz 0éščřžýáíéúťňď 123456
 7890 !@#%&'()* _ [] - + = : ; ? / © ABCDEFGHIJKLMNOPQRSTU
 VWXYZ 0ÉŠČŘŽÝÁÍÉÚŤŇĎ • "

• "klávesyF" a "klávesyS" (obrazce)

rozložení na klávesnici:



OBRAZCE SE ZDE PÍŠÍ VŽDY DVĚMA KLÁVESAMI,

příklad : B= N= VZNIKNE

• "rene"

abcdefghijklmnopqrstuvwxyz oěščřžýáíéúůň' 1234567
890 !@#%&()_<>~| \{}+ - * = : ; ? / * ABCDEFGHIJKLMNOP
QRSTUVWXYZ OĚŠČŘŽÝÁÍÉÚŮŇ'

• "era"

abcdefghijklmnopqrstuvwxyz oěščřžýáíéúůň' 12345678
90 ABCDEFGHIJKLMNOPQRSTUVWXYZ OĚŠČŘŽÝÁÍÉÚŮŇ'
!@#%&()_<>~| \{}+ - * = : ; ? / *

• "kurziva"

abcdefghijklmnopqrstuvwxyz oěščřžýáíéúůň' 123456
7890 !@#%&()_<>~| \{}+ - * = : ; ? / *

• "šipky2F" a "šipky2S" (obrazce)

klávesy abecedy zde nejsou vůbec použity a
obrazce jsou uloženy pod klávesami 1-5.



(Dokončení ze strany 23)

POKE adresa_hodnota. Hodnota je v rozmezí 0-255. Je možné vložit třeba i -1. Počítač je chytrý a převede si to na 255. Lze samozřejmě dát i -2, Spectrum to přelouská. Hodnoty větší než 255 se ukládají pomocí drobné tinty uvedené později. Jediné co asi nejde, je volat nějakou SP příkazem USA adresa. (Ha, Ha, Ha - smích v okně) Pozor! Neuvážené změny SP mohou způsobit (někdy) havárii systému, proto měním jen ty SP u kterých vím co se stane když to změním. Experimenty se doporučují jen v případě, že v paměti není žádný strategický program.

Pokud pracujete ve strojáku a nemíníte se už vůbec odkazovat na Basic ani nepoužívat rutiny z ROMky, tak můžete oblast SP použít

jako normální paměť, třeba jako bafr pro text. Většinou je ale dobré, když už programujete stylem „stroják/basic, když budete SP používat a vesele se na ně odkazovat a využívat jejich služeb. Je to výhodné, uspoří Vám to někdy dost námahy a času.

Důležitá technická poznámka pro pracovníky ve strojáku: při ne-destruktivním návratu ze svého programu ve strojáku do Basicu musí (zdůrazňuji musí) být registr IY nastaven na hodnotu #5c3a, protože ROMka se na většinu SP odkazuje !ld (iy+xx),a(kde >xx) je příslušný pevně zvolený odkok (displacement). Kdyby IY mělo jinou hodnotu tak by interpret měnil jiné SP a systém by to asi psychicky nepřenesl přes srdce a kolaboval by. Dejte si na to velký pozor.

AHA...



amatérský programátor

Soukromý a zcela nezávislý časopis pro amatérské programátory na počítačích ZX Spectrum, Didaktik a kompatibilních. Vychází každý druhý měsíc v rozsahu 40 stran. Cena jednoho čísla je 15,- Kč. Předplatné ve výši 132,- Kč (včetně poštovného) lze zajistit na adrese redakce:

AP-SECOM - PETR ČERNÝ - 407 61 STARÉ KŘEČANY

Do tohoto čísla přispěli: P.Macek, J.Drexler, V.Vojta, MUDr.Fr.Slůger, E.H.Becz, P.Rak.

Tiskne soukromá maloobětová tiskárna SECOM St.Křečany.
Toto číslo vyšlo v březnu 1993.



haló uživatelé ZX Spectra, 128K, +2, +2A, +3, Timex, Didaktik Gama, M, SAM Coupé.

nechcete lépe využít svůj počítač

Netoužíte získat v zahraničí oblíbené doplňky k němu při přepočtu 1DM za 2 až 4,- Kč? Neláká Vás dohrát každou i sebeobtěžnější hru do zdárného konce? Zde je stručný přehled některých možností:

- zvuk přes TVP. Libovolná hlasitost zvuku z reproduktoru televize.
- úprava interface pro SOUČASNÝ provoz dvou joysticků. • zpomalovač her. Dohrajete libovolnou akční hru do konce. • připojení lib. tiskáren Star, Seikosha, Epson, SP120/180VC atd. • ovladače těchto tiskáren pro DWRITER, DTEXT, Tasword 2CS, Masterfile CS, Datalog, ART-STUDIO, Omnicalc CS přes lib. IF (LPRINT 3, Kempston) • úprava tuzemské IF DIPRA pro provoz u všech her. • úprava IF Kovodružstva Náchod na kompatibilní s 2x Sinclair joy. • úprava sériové tiskárny či zapisovače na paralelní styk Centronics. • sériové rozhraní pro ZXS, DG, SAM. • opravy diskových řadičů DISCIPLE, SWIFT, PLUS D, připojení a ovladače tiskáren pro ně, kopirovací programy. • připojení disketové jednotky D40 k ZXS 128K, +2, +2A, +3. • opravy různých interface pro ZXS, např. videodigitizeru. • barvený pásek pro tiskárny, resp. si můžete obarvit vlastnoručně. • pro staré pásky do tiskáren spec. barva k osvětlení (vydrží 2 až 3 roky) • připojení libovolných periférií, tiskáren, zapisovačů, atd. • poradenská služba (hardware, software). • opravy a úpravy ZX Spectrum a kompatibilních. • programování EPROM, PROM, PAL, EEPROM, GAL a jednočipových /uP. • MIDI. Obousměrná komunikace s hudebním nástrojem - profesionální. • přenos dat na a z jiných počítačů, např. IBM PC. • komunikace normou RS-232, GP-IB, IEC-625, HP-IL, Centronics atd. • testy periférií, zjištění schematu, analýza funkce HW i SW. • hudební el.: Flanger, echo, vocoder, harmonizer atd., pomocí ZXS. • úprava IF Dktronics pro 2 joysticky současně (Sinclair IF2). • barevné nálepky na vnější klávesnici v lib. velikosti. • nálepka s vícebarevným popisem sběrnic ZXS na konektor. • nálepky na IO s popisem vývodů (výborné pro identifikaci pinů). • barevný a černobílý videovýstup ze ZXS, připojení lib. monitoru. • úprava odstraňující vytahování jacku z EAR při SAVE.
- Didaktik Gama: odstranění nekompatibility, magické tlačítko, bezproblémové POKE, BREAK, LIST, SAVE, COPY u spuštěných programů. • Timex: úprava pro připojení UR-4 a dalších periférií. • úprava IF UR-4 či Soldan na Teletex a na Sinclair interface 2. • kvalitní bicí (jako YAMAHA za 5000,- DM) + sady bubnů (590+200,-). • připojení zvukového IO AY-3-8910 a hudební software. • výměna programů, manuálů, schémat a dalších zajímavostí. • návody a dokumentace ke stavbě: připojení ATARI1029, AT1050, hudební generátor + SW, schemata Issue 1-6, +2, +2A, +3, vývojový systém MCS 48/51 (pro mikroprocesory řad 8048 - 8051), tester a identifikátor obvodu TTL/CMOS/OZ atd., univ EPROM modul, vnější paměť 256kB se simulací IF1 + /uD, Multiface 1, M128, Muliface 3, programátor EPROM, GAL a pod.

▶ případné dotazy plus další seznamy za zpětnou frankovanou obálku na adrese: Jan Drexler, Jahodová 2889, 106 00 PRAHA 10 4