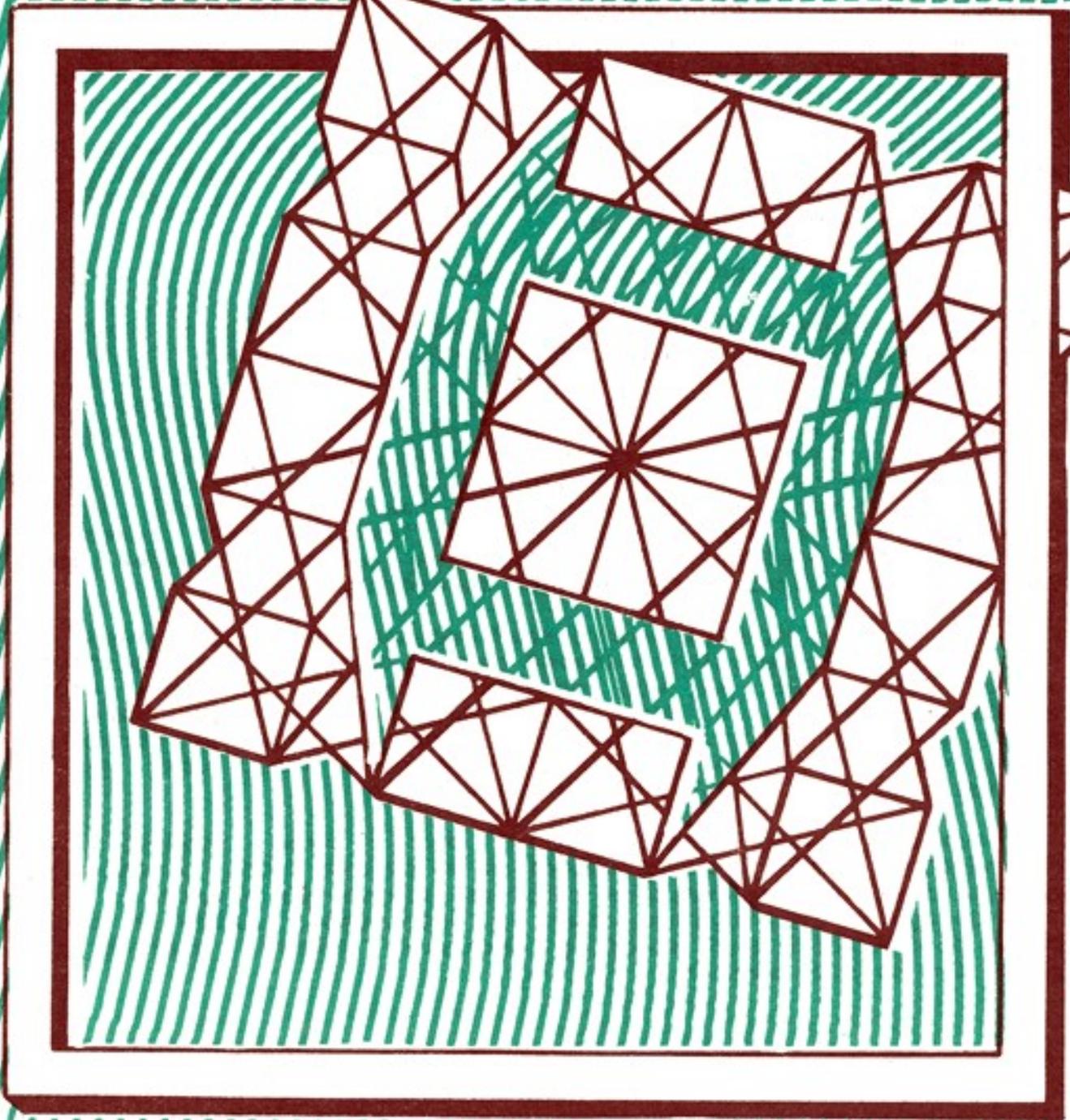




602

ATARI

1
90



ZE ZÁPISNIKU PROGRAMÁTORA

TEXT V GRAFICKÝCH MODECH (3)

Ještě do třetice zůstaneme u grafického modu. č. 8. Podprogramy z minulých částí zobrazovaly znaky v rastru 40 sloupců x 24 řádků, t. j. stejně jako v grafickém modu č. 0. Dále probíraný podprogram může znaky zobrazovat navíc v rastrech:

40 sloupců x 12 řádků,
80 sloupců x 24 řádků,
80 sloupců x 12 řádků.

Máme tedy k dispozici znaky s normální a poloviční šířkou, a s normální a dvojnásobnou výškou. Autorem podprogramu je známý Peter Finzel z NSR.

Způsob použití lze částečně vyčíst z basicového výpisu na str. 4. Na řádcích 30010 až 30220 je instalován do paměti vlastní podprogram, a potom vyvolan mod GRAPHICS 8. Nyní již je možné zobrazovat texty pomocí příkazu:

X=USR(1536,adr,poc,druh) ,

kde "adr" je adresa prvního znaku zobrazovaného textu v paměti. Je-li text v proměnné A\$, pak adr=ADR(A\$). Údaj "poc" je délka textu (počet znaků včetně mezer, např. LEN(A\$)). Specifikace "druh" udává rozměr znaku. Její hodnoty jsou uvedeny v řádcích 79 až 83. Před voláním podprogramu však ještě musíme určit pozici prvního znaku textu na obrazovce pomocí příkazu PLOT x,y.

Ve výpisu na str. 4 řádky 120 až 130 kreslí rámeček. Řádky 140 až 190 provádí vlastní výpis textů tak, že postupně je z data-řádků 1000 až 1060 přečtena pozice textu, rozměr znaku a vlastní text a následně volán podprogram zobrazení. Řádek 190 zamezí zrušení obrazu ihned po jeho dokončení.

Jak pracuje vlastní podprogram? Podobně jako podprogramy z předešlých částí seriálu! Postupně bere znaky textu, přepočítává jejich ASCII kódy na kód interní, pomocí interního kódu vypočítává adresu i. z osmi bajtů definujících tvar znaku v paměti a těchto osm bajtů přenáší na určené místo do obrazové paměti (VRAM). Před umístěním znaku v obrazové paměti však navíc provádí případnou úpravu rozměru znaku.

Úprava rozměru znaku spočívá buď v jeho zvýšení na dvojnásobnou výšku, nebo ve zúžení na poloviční šířku. U znaku o poloviční šířce a dvojnásobné výšce se uplatní oba případy.

Úprava na dvojnásobnou výšku je jednodušší. Každý z osmi bajtů definujících znak stačí na obrazovku přenést dvakrát za sebou (ve svislém směru).

Zúžení znaku je složitější. Tvar znaku je definován 8 bajty, tedy v matici 8x8 bitů. Upravený znak bude mít matici 8x4 bity, tj. 8 bajtů. 4. až 7. bit každého bajtu bude prázdný (nulový). Tvar znaku budou definovat 0. až 3. bit. Z osmi bitů u normálního znaku tedy musíme vybrat 4 bity tak, aby znak byl čitelný. Tento princip znázorňuje obrázek na str. 5. Autor podprogramu zvolil jako rozhodující bity 0, 2, 4 a 5. Hodnota těchto bitů je přenášena do bitů 0 až 3 upraveného znaku. Tím je znak zúžen. Další krok je závislý na textové souřadnici znaku na obrazovce. Jeden řádek obrazovky tvoří 40 bajtů, což odpovídá 40 normálním znakům. My potřebujeme 80 znaků na řádek. Musíme tedy umístit na místo jednoho normálního znaku dva zúžené. Pokud bude vodorovná souřadnice sudá, přesuneme zpracováváný znak do pravé poloviny normálního znaku, tedy z bitů 0 až 3 do bitů 4 až 7. Pokud je souřadnice lichá, bity 0 až 3 zůstávají. Sloučení obou zúžených znaků do jednoho prováděno přímo ve VRAM. Podprogram postupně vyzvedne z příslušného sloupce obrazovky její obsah a pomocí logického součtu jej sloučí s bajty zpracováváného znaku. Znak se sudou souřadnicí je v bitech 4 až 7, znak s lichou souřadnicí je v bitech 0 až 3.

Pokud se zobrazuje znak s poloviční šířkou a dvojnásobnou výškou, zobrazí se každý bajt zúženého znaku 2x za sebou.

Nyní v krátkosti k výpisu strojového podprogramu na str. 5 až 7. Řádky 1080 až 1260 definují pomocné proměnné a adresy proměnných operačního systému. Na řádcích 1320 až 1490 jsou převzaty adresa a délka textu a rozměr znaků, dále jsou vynulovány některé pomocné adresy. Textová souřadnice počátku textu se určuje na řádcích 1540 až 1610. Pro každý znak textu je na řádcích 1660 až 1740, zjišťováno, zda nebude mimo obrazovku (ve vodorovném směru). Interní kód znaku určují řádky 1780 až 1900. Na řádcích 1940 až 2030 se počítá adresa prvního z osmi bajtů definujících tvar znaku v paměti. Od 2070. do 2160 řádku probíhá vlastní úprava a zobrazení znaku za pomoci dvou podprogramů: UPRAV a UL02. Příprava na zpracování dalšího znaku nebo návrat do basicu po zobrazení celého textu je na řádcích 2170 až 2310. Vlastní úprava znaku, pokud ji požadujeme, je prováděna na řádcích 2350 až 2540, zobrazení probíhá na řádcích 2690 až 2810.

Poznámky:

- zúžené znaky jsou čitelnější, jsou-li tmavé na světlém podkladě;
 - inverzní znaky jsou zobrazovány jako znaky normální;
 - je zachováván původní obsah obrazovky pod zobrazovanými znaky;
 - podprogram není přemístitelný bez přepočtu některých adres.
- js-

Příště: Text ve čtyřbarevných grafických modech - program TEXTPLOT.

```

FW 79 REM KODOVA CISLA
HD 80 REM 0=40 NORMALNICH ZNAKU
FY 81 REM 64=40 ZNAKU DVOJITE VYSKY
EF 82 REM 128=80 ZNAKU NORMALNI VYSKY
LC 83 REM 192=80 ZNAKU DVOJITE VYSKY
BL 84 REM V datech jsou prva dve cisla
TS 85 REM pro posici pisma a treti cislo
CS 86 REM pro typ pisma
JS 87 REM Posice se udava pro GRAFIKU 8
TL 88 REM *** vzor zadani: ***
DV 89 REM COLOR 0:PLOT 120,100
IT 90 REM X=USR(ADR("Text")), pocet
      znaku, kodove cislo znaku)
BI 91 REM
BK 92 REM
BJ 100 DIM A$(127):GOSUB 30000
ZX 110 GRAPHICS 8+16:SETCOLOR 2,8,10:SETC
      OLOR 1,0,0
MV 120 COLOR 1:PLOT 5,5:DRAWTO 314,5:DRAW
      TO 314,186:DRAWTO 5,186:DRAWTO 5,5
YF 130 PLOT 5,150:DRAWTO 314,150
EJ 140 RESTORE 1000:FOR I=0 TO 7
GK 150 READ X,Y,ATT,A$
UV 160 COLOR 0:PLOT X,Y
XH 170 A=USR(1536,ADR(A$),LEN(A$),ATT)
GH 180 NEXT I
PT 190 GOTO 190
EW 1000 DATA 80,20,64,80-ZEICHENKONVERTER
RP 1010 DATA 40,40,128,fuer alle 8-Bit AT
      ARI-Computer 400/800/600XL/800XL/130XE
YY 1015 DATA 64,70,128,ERZEUGT 80-ZEICHEN
      DISPLAY IN DER GRAPHIKSTIFE 8
EL 1020 DATA 60,100, 192,und zusaetzlich
      recht flexibel in den Schriftarten
FA 1030 DATA 16,120,0,PETER'S
RZ 1050 DATA 80,160,64,ASSEMBLERECKE
SC 1060 DATA 192,160,128,IN COMPUTER KONT
      AKT

```

4

```

IA 30000 REM * 80-ZEICHEN ML-PROGRAMM
EV 30010 S=0:RESTORE 30100
EG 30020 FOR A=1536 TO 1746:READ D:POKE A
      ,D:S=S+D:NEXT A
PB 30030 IF S(>25816 THEN ? "DATEN-FEHLER
      !":STOP
EJ 30090 RETURN
RN 30100 DATA 104,104,133,209,104,133,208
      ,104,104,133,210,104,104,133
AJ 30110 DATA 214,169,0,133,211,133,205,1
      33,213,165,92,133,216,165,91
VQ 30120 DATA 102,216,106,102,216,106,133
      ,212,165,212,201,80,176,103
AR 30130 DATA 106,102,213,165,94,133,203,
      165,95,133,204,160,0,132,207
IQ 30140 DATA 164,211,177,208,41,127,201,
      96,176,11,201,32,176,4,9,64
CC 30150 DATA 208,3,56,233,32,10,10,38,20
      7,10,38,207,133,206,24,165,207
IJ 30160 DATA 109,244,2,133,207,162,8,160
      ,0,177,206,36,214,16,3,32,147
JU 30170 DATA 6,32,191,6,36,214,80,3,32,1
      91,6,230,206,202,208,230,230
RR 30180 DATA 212,36,214,16,6,36,213,48,4
      ,16,4,230,212,230,205,230,211
TZ 30190 DATA 198,210,208,147,96,133,215,
      169,0,133,216,160,3,185,183
WD 30200 DATA 6,36,215,240,7,185,187,6,5,
      216,133,216,136,16,239,165,216
VQ 30210 DATA 36,213,48,4,10,10,10,10,96,
      1,4,16,32,1,2,4,8,72,164,205
QI 30220 DATA 17,203,145,203,24,165,203,1
      05,40,133,203,144,2,230,204,104,96

```

1000	:80	ZNAKU V MODU GR.8			
1010	:				
1020	:	Volani z BASICu:			
1030	:	X=USR(1536,ADR(TEXT),LEN(TEXT),DLJH)			
1040	:	Inde-ADR(TEXT) je adresa textu			
1050	:	-LEN(TEXT) je delka textu			
1060	:	-DLJH je specifikace rozmeru znaku.			
1070	:				
1080	:	Promenne OS:			
1090	:				
1100	:	OLDCOL = \$B	isloupec st. kurzoru		
1110	:	OLDADR = \$E	iadr. st. kurzoru ve VRAM		
1120	:	CHBAS = \$ZF4	ukazatel znakové sady v ROM		
1130	:				
1140	:	Pomocne promenne:			
1150	:				
1160	:	TEXT = \$08	iadresa textu		
1170	:	LEN = \$02	idelika textu		
1180	:	DLJH = \$06	irozmer znaku		
1190	:	NOTVAR = \$08	iuprav. tvar znaku		
1200	:	TCOL = \$04	ipomocny textovy kurzor		
1210	:	POMKUR = \$CB	iadr. POMKUR ve VRAM		
1220	:	TVAR = \$CE	iadresa tvaru znaku v ROM		
1230	:	POM2 = \$05	ipomocna adresa		
1240	:	POM3 = \$03	;		
1250	:	POM4 = \$CD	;		
1260	:	POM5 = \$D7	;		
1270	:				
1280	:				
1290	:				
1300	:	Start programu, prevzeti parametru			
1310	:				
1320	:	ZAC PLA	ipocet parametru		
1330	:	PLA	iadresa textu		
1340	:	STA TEXT+1	ido TEXT		
1350	:	PLA			
1360	:	STA TEXT			
1370	:	PLA			
0600					
0600	68				
0601	68				
0602	8501				
0604	68				
0605	8500				
0607	68				
0608	68				
0609	8502				
0608	68				
0609	8502				
060C	68				
060D	8506				
061F	6A				
0620	6508				
0622	6A				
0623	8504				
0625	8504				
0627	C350				
0629	8657				
062B	6A				
062C	6505				
062E	655E				
0630	850B				
0632	655F				
0634	850C				
1380	68	PLA	idelika textu		
1390	8502	STA LEN	ido LEN		
1400	68	PLA			
1410	68	PLA	idruh znaku		
1420	8506	STA DLJH	ido DLJH		
1430	:				
1440	:	Priprava pomocnych promennych			
1450	:				
1460	8500	LDA #0	inuluj		
1470	8503	STA POM3			
1480	850D	STA POM4			
1490	8505	STA POM2			
1500	:				
1510	:	Vypocet polohy text, kurzoru z polohy			
1520	:	igrafickeho kurzoru			
1530	:				
1540	855C	LDA OLDCL+1	ipolohu graf, kurzoru		
1550	8508	STA NOTVAR			
1560	8558	LDA OLDCL			
1570	6508	ROR NOTVAR	ivydel cislem '4'		
1580	6A	ROR A			
1590	6508	ROR NOTVAR			
1600	6A	ROR A			
1610	8504	STA TCOL	ituloz do TCOL		
1620	:				
1630	:	Overeni pripustne polohy text, kurzoru			
1640	:	180 znaku)TCOL... (8;79)			
1650	:				
1660	8504	TEST LDA TCOL	iporovnej TCOL		
1670	C350	CHP #80	is cislem 80		
1680	8657	BCS NVRAT	TCOL)88-navrat do BASICu		
1690	6A	RJR A	TCOL v poradku		
1700	6505	P/R POM2			
1710	8508	LDA OLDADR	ituloz polohu adr. st. kurzoru		
1720	850B	STA POMKUR	ive VRAM		
1730	655F	LDA OLDADR+1			
1740	850C	STA POMKUR+1	ido POMKUR		
1750	:				
1760	:	Prevod ATASCII->SCREEN kod			

0635 0000	LDY #0	inulu; vyssi bajt adresy	2150 BVC MOVMS	:6.bit=0...norm. vyska
0638 84CF	STY TVAR+1	iadresy tvaru	2160 JSR UL0Z	:6.bit=1...dvojita vyska
063A 0AD3	LDY POKS		2170 MOVMS INC TVAR	i dalsi bajt tvaru znaku
063C B100	LDA (TEXT),Y	iASCII kod znaku textu	2180 DEX	
063E 237F	AND #47F	i:invertzni znak-)norm. znak	2190 BNE ZNKK	i:jeste nezprac. cely znak
0640 C360	CHP #56	i:male pismeno?	2200 INC TOOL	i:posun textovy kurzor
0642 800B	BCS POCITEJ	i:ano, kod 0.K.	2210 BIT DRUH	i:jaka sirka?
0644 C320	CHP #32	i:velke pismeno?	2220 BPL NOSIRI	i:normalni
0646 8084	BCS VELKE	i:ano!	2230 BIT POK2	i:lilcha ci suda poloha kurzoru?
0648 8340	ORA #64	i:graf.znak, pricti '5A'	2240 BHI LIDHA	i:7.bit POK2=1...lilcha
064A 00B3	BNE POCITEJ		2250 BPL SUDA	i:7.bit POK2=0...suda
064C 38	VELKE SEC	i:velke pismeno =)	2260 NOSIRI INC TOOL	i:jeste posun kurzor
064D E320	SBC #32	i:odecti '32'	2270 LIDHA INC POKA	
			2280 SUDA INC POKS	
			2290 DEC LEN	i:delka textu -1
			2300 BNE TEST	i:nezprac. cely text
			2310 MVRAT RTS	i:text zpracovan;navrat do BBS.
			2320 ;	
			2330 iPodprogram zuzeni znaku	
			2340 ;	
064F 0A	POCITEJ ASL A	i:SCREEN kod nasob '8'	2350 UPRAV STR POKS	i:tvar znaku do POKS
0650 0A	ASL A	i:(8 bajtu...1 znak)	2360 LDA #0	i:nulu;
0651 25DF	ROL TVAR+1		2370 STA NOTVAR	i:NOTVAR
0653 0A	ASL A		2380 LDY #3	
0654 25DF	ROL TVAR+1		2390 UPRAVZ LDA MASK1,Y	i:masku; tvar znaku
0656 85DE	STA TVAR	i:vysedek do TVAR	2400 BIT POKS	
0658 18	CLC	i:prirava na scitani	2410 BEQ UPRAV3	i:bit znaku nulovy
0659 A5DF	LDA TVAR+1	i:pricti adresu znakove	2420 LDA MASK2,Y	i:testovany bit nastaven.
065B 50F402	ADC D-BMS	i:sady v ROM, uloz do TVAR	2430 ORA NOTVAR	i:nastav prisl. bit zuzeneho
065E 85DF	STA TVAR+1		2440 STA NOTVAR	i:tvaru znaku
			2450 UPRAV3 DEY	i:dalsi bit
			2460 BPL UPRAV2	i:tvaru znaku
0660 A208	LJX #8	i:8x proved.(1 znak ... 8 bajtu)	2470 LDA NOTVAR	
0662 0000	ZNKK LDY #0		2480 BIT POK2	i:lilcha ci suda pozice?
0664 B1DE	LDA (TVAR),Y	i:bajt tvaru znaku	2490 BHI UPRAV4	i:lilcha (prava)
0666 24D6	BIT DRUH	i:jaky druh znaku?	2500 ASL A	i:suda (leva), posun
0668 10B3	BPL NOSIR	i:7.bit=0...norm. sirka	2510 ASL A	i:oprava
066A 203306	JSR UPRAV	i:7.bit=1...polovici sirka	2520 ASL A	
066D 20BF06	NOSIR JSR UL0Z	i:zobraz bajt		
0670 24D6	BIT DRUH	i:jaka vyska znaku?		
0672 50B3				
0674 20BF06				
0677 E5DE				
0679 0A				
067A 00E5				
067C E5D4				
067E 24D6				
0680 1005				
0682 24D5				
0684 3004				
0686 1004				
0688 E5D4				
068A E5CD				
068C E5D3				
068E D5D2				
0690 D093				
0692 60				
0693 8D07				
0695 0900				
0697 8D08				
0699 A003				
069B 89B706				
069E 24D7				
06A0 F007				
06A2 89B006				
06A5 8D08				
06A7 8D08				
06A9 88				
06AB 10EF				
06AC 0D08				
06AE 24D5				
06B0 3004				
06B2 0A				
06B3 0A				
06B4 0A				

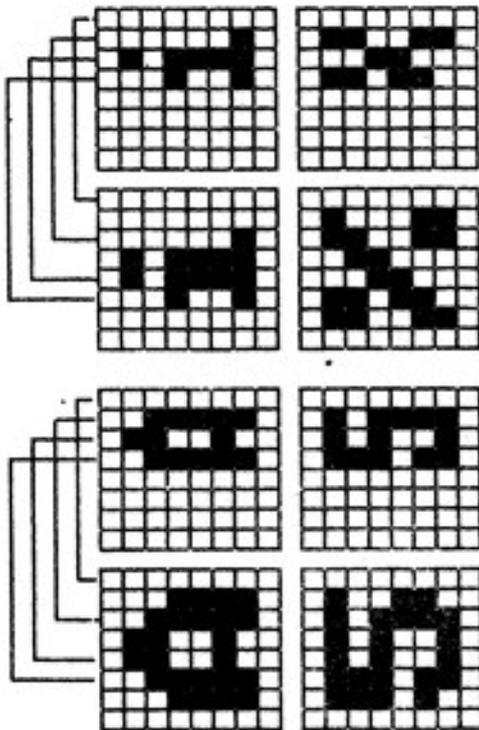
```

06B5 0A   ASL A
06B6 60
2538     ASL A
2548 UPRRAV RTS
2558 ;
2568 ;Pomočne masky
2578 ;
2588 MASK1 .BYTE 1      ;bin:00000001
2598     .BYTE 4        ;bin:00000100
2608     .BYTE 16       ;bin:00010000
2618     .BYTE 32       ;bin:00100000
2628 MASK2 .BYTE 1      ;bin:00000001
2638     .BYTE 2        ;bin:00000010
2648     .BYTE 4        ;bin:00000100
2658     .BYTE 8        ;bin:00001000
2668 ;
2678 ;Podprogram zobrazení znaku
2688 ;
2698 ULOZ PHA           ;ulož tvar znaku do zásobníku
2700 LDY POMA
2710 DBA (POMAJ),Y     ;tlač tvar znaku s pu-
2720 STA (POMAJ),Y     ;vodním obsahem obrazu
2730 ;                 ; a ulož na obrazovku
2740 CLC
2750 LDA POMAJ
2760 ADC #48            ;další graf. řádek obrazovky
2770 STA POMAJ
2780 SOC ULOZ1
2790 INC POMAJ+1
2800 ULOZ1 PLA        ;vyzvedni tvar znaku
2810 RTS              ;a vrat se
2820 ;

```

*** ASSEMBLY ERRORS: 0 21761 BYTES FREE

Princip zuzovani znaku:



80-ZEICHENKONVERTER

für alle 8-Bit AT&T-Computer 486/888/888HL/888HL/138ME

ERZEUGT 88-ZEICHEN DISPLAY IN DER GRAPHIKSTUFE 8

und zusätzlich recht flexibel in den Schriftarten

PETER'S

ASSEMBLERECHE IN COMPUTER KONTAKT

JAK NA TO?

TOMAHAWK

BY
DAVE MARSHALL

(Simulace vrtulníku ARMY AM-64 APACHE)

Máte jedinečnou příležitost stát se pilotem bojového vrtulníku. Nahrajte program TOMAHAWK do Vašeho počítače, zvolte si jazyk, který je Vám blízký (angličtina -francouzština-němčina) a můžete začít.

Simulátor se ovládá buď pomocí dvou joysticků nebo jednoho joysticku a klávesnice nebo pouze pomocí klávesnice. Nejdříve tlačítka SELECT a OPTION zvolte v hlavním menu podmínky pro let

- MISE 1 - seznámení s terénem a bojem (bez opětování nepřátelské palby)
- MISE 2 - boj, zničení nepřítele a návrat na základnu
- MISE 3 - osvobození se z nepřátelského obklíčení
- MISE 4 - bitva o území

Ve čtvrté variantě jde o obsazení celého území, které je na mapě označeno jako nepřátelské. Každý protivníkův sektor se stane spojeneckým, jsou-li v něm zničeny všechny cíle.

Dále je možno volit mezi denní a noční variantou, mezi

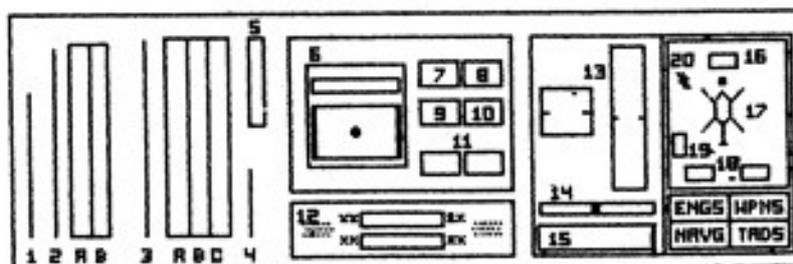
jasným nebo oblačným počasím. U oblačného počasí pak volíme výšku mraků od 50 do 1000 stop.

Z posledního řádku menu přejdeme k vlastní simulaci letu:

V dolní třetině obrazovky je vidět palubní deska s přístroji, v horních dvou třetinách pak podle zvoleného režimu buď mapa nebo pohled z kokpitu. Budete-li ovládat simulátor pomocí klávesnice, použijte těchto tlačítek:

- Q...otvírání klapky karburátoru
- R...uzavírání klapky karburátoru
- W...páka kolektivní nahoru <zvyšuje tah>
- S...páka kolektivní dolů <sníží tah>
- C...volba objektu, na který bude vrtulník navisován
 - H - základna
 - B - radiomaják
 - T - pozemní cíl
 - vrtulník
- N ..upřesnění objektu, na který bude vrtulník navisován
 - H <0-3>
 - B <0-7>
 - T <0-7>
- P ..volba zbraňového systému
 - kanon - dostřel 2000 stop
 - rakety - dostřel 4000 stop
 - samonaváděcí rakety - dostřel 3.1 míle
- M...mapa <vyvoláte-li mapu při přistání, je možné pomocí tlačítek 5,6,7,8 přemisťovat vrtulník do jiného sektoru>
- Z,X.....ovládá vrtulník směrově při nulové dopředné rychlosti
- 4,5,8,9...ovládá výkrut
- 6,7.....ovládá náklon.

SCHÉMA ROZNÍSTĚNÍ PŘÍSTROJŮ:



1. páka kolektivní
2. točivý moment
 - A-motor č.1
 - B-motor č.2

3. otáčky
 - A-motor č.1
 - B-rotor
 - C-motor č.2

10

4. otevření škrtící klapky karburátoru
5. množství paliva
6. TADS - systém pro vyhledání a určení cíle
7. rychlost
8. výška
9. čas potřebný k dosažení zvoleného cíle
10. rychlost stoupání /klesání/
11. vzdálenost od zvoleného cíle
12. množství munice
 - 1200 nábojů do 30mm kanonu
 - 38 raket
 - 8 samonaváděcích raket
13. ukazatel výkrutu a klopení
14. zatačkoměr
15. ukazatel skóre
16. kompas
17. indikace polohy zvoleného cíle vůči vrtulníku
18. VOR
19. indikace druhu zvoleného cíle
20. indikace vzdušného napadení

Kontrolní světla signalizace poruchy:

ENGINES-motorů

WEAPONS-zbraňového systému

NAVIGATIONS-navisačních přístrojů

TADS-systému pro vyhledávání a určování cílů

Rozsáhlá poškození jsou indikována tak, že symbol vrtulníku bliká. Třetí vážný zásah je poslední.

Mezní rychlost vrtulníku je asi 160 uzlů/hod., maximální rychlost stoupání je 1450 stop/min., přistát lze rychlostí menší než 60 uzlů/hod. při rychlosti klesání menší než 12 stop/sec. S vrtulníkem je možno pojíždět po zemi. Tankování, doplnění munice a opravy zajistíte přistáním na základně ve spojeneckém sektoru. Zničení spojence má za následek vynulování skóre.

Hodně štěstí!

Svatopluk Škuta

Poznámka redakce: Tento příspěvek si neklade za cíl poskytnout detailní návod jak hrát. Chce pouze alespoň trochu ulehčit start těm, kteří o hře vůbec nic neví. Jedním z předpokladů úspěšného ovládní je také znalost principů letu a řízení vrtulníku.

S L A B I K Á R A T A R I S T Y

OPERAČNÍ SYSTÉM TURBO 2000

=====

Téměř každý atarista v krátké době po zakoupení počítače začne používat systém zrychleného přenosu dat TURBO 2000 a oceňovat jeho výhody. Poprávu, neboť zkrácení nahrávky z 15 na 1,5 min. je jeho velkou výhodou. Nevýhodou je, že klasické TURBO 2000 či SUPERTURBO neumí stejnou rychlostí ukládat data. "Hráčům" to nevadí, avšak ti, kteří počítač chtějí vážně používat, se budou ohlížet po něčem šikovnějším. A to, po čem s největší pravděpodobností sáhnou, bude některý z Turbo operačních systémů, zkráceně nazývaných TOS.

POZN.: Dříve, než začneme probírat TOS, vyjasníme si některé termíny:

- klasické TURBO - systém ukládání programů v jednom bloku, libovolně dlouhém (tímto způsobem bývají nahrány programy hry, i uživatelské)
- zařízení "T:" - systém ukládání dat; data jsou ukládána také systémem Turbo, avšak ve dvou či více dávkách o délce 1024 b.
- ramdisk - oblast paměti RAM v počítači, se kterou se pracuje jako s disketou.

TOS je program, který v sobě sdružuje řadu užitečných funkcí: zavaděč, generátor a "kopírák" nahrávek TURBO 2000. Dále nám umožní inicializovat kartridž nebo BASIC ROM, kopírovat datové soubory, spouštět programy ve strojovém kódu a využívat zařízení "T:" a ramdisk. Tedy téměř totéž, co majitelům disket umožňuje DOS (diskový operační systém).

TOSů existuje celá řada, avšak nejpoužívanějším z nich je bezesporu TOS 4.1 Milana Říhy z Brna. Tento program nám také poslouží jako praktický příklad, na kterém si vysvětlíme používání TOSů.

Používání TOS 4.1

Program TOS 4.1 dává uživateli k dispozici všechny dříve uvedené možnosti použití. Pro práci s magnetofonem definuje zařízení "D:", "D2:" a zůstává nám zařízení "C:". "D:" je vlastně přejmenované zařízení "T:", "D2:" pracuje standardní rychlostí 600 Bd s krátkými meziblokovými mezerami. Zařízení "C:" zůstává beze změn. TOS 4.1 definuje ramdisk jako zařízení "D8:". Umístění ramdisku a jeho velikost je možné měnit. Výhodou tohoto TOSu je, že nám umožní i práci s programy, které jsou určeny pouze pro

12

spolupráci s disketou.

Program nahrajeme obvyklým způsobem, tj. pomocí zavaděče TURBO 2000, do počítače. Po nahrání se zobrazí následující nabídka:

```
TURBO 2000 Operating System V4.1
[ C ] Milan Riha 1988 AK BRNO

Load T 2000 file
Goto at adress
Run cartridge
Save T 2000 program
Duplicate T 2000 program
Copy data file
Basic/Ram ;now RAM
Modify ramdisk ;now $C000-$FFFF
Last run adr: $0B00
LOMEM : $1771

SELECT ITEM
█
```

Věta "SELECT ITEM" nás vybízí k volbě některé z funkcí. Nabízené funkce volíme stiskem příslušné klávesy a <RETURN>. Většina funkcí po volbě vyžaduje ještě odpověď na několik otázek. Jednotlivé funkce si nyní probereme detailně.

Load T 2000 file - <L>

Tato funkce slouží jako zavaděč nahrávek TURBO 2000. Zvolí se klávesou <L> a stiskem <RETURN>. Po volbě se vypíše otázka "Spec:". Program po nás chce zadat jméno souboru. Název souboru můžeme zadat, ale můžeme také stisknout pouze <RETURN>. V prvním případě bude program prohledávat kazetu tak dlouho, dokud nenarazí na soubor zadaného jména. Při hledání vypisuje jméno každého souboru, který našel, např.:

```
Found:Dmaster
Found:Koala,
a pod.
```

V okamžiku, kdy je nalezen soubor zadaného názvu, se ozve zvukové znamení a soubor je nahrán do paměti. Zadáme-li na místo jména souboru pouze <RETURN>, bude nahrán první nalezený soubor.

Budeme-li se pokoušet nahrát soubor, který by přemazal část programu TOS 4.1, vypíše zavaděč hlášení "Adr. Error" a přeruší nahrávání. Takový soubor nebude nahrán.

V době, kdy zavaděč prohledává kazetu, můžeme pomocí stisku některé z kláves <OPTION>, <SELECT> a <START> zastavit a znovu spustit motor magnetofonu.

Po nahrání souboru zavaděč vypíše hlášení, např.:

```
Found: Cizek 3.1
      $1DF1-$4D02, Start:$1DF1
Run (Y/N)?
```

Druhý řádek oznamuje umístění programu v paměti (\$1DF1-\$4D02) a startovací adresu (Start:\$1DF1). Na třetím řádku se nás TOS ptá, zda má nahraný program spustit, či ne. Po stisku <Y> bude program spuštěn, po <N> TOS přejde do režimu volby funkce. Nespustíme-li program po nahrání klávesou <Y>, můžeme jej spustit později pomocí funkce

Goto at adress - <G>

Tato funkce spouští strojové programy. Po volbě <G> vypíše otázku na startovací adresu "Adress:\$". Odpovíme např.:

```
Adress:$1DF1
```

a program bude spuštěn.

Run cartridge - <R>

Jsme-li šťastní majitelé nějakého zásuvného modulu - kartridže, můžeme ji pomocí této funkce inicializovat. Zasuňme kartridž do počítače a stiskneme <R>. Program se z modulu přenesení do počítače, umístí na pracovní adresu a spustí. V případě, že vyvoláme funkci "Run cartridge" bez zasunutého modulu, bude vypsané chybové hlášení "No cartridge".

Tuto funkci můžeme použít také k přechodu z TOSu do Basicu. O tom však později.

Save T 2000 program - <S>

Funkce představuje opak "Load T 2000 file". Po volbě <S> musíme zadat: počáteční adresu programu, koncovou adresu programu, startovací adresu a nakonec název programu. Např.:

```
From:$5000
To:$5AFF
Run:$5900
Name:jmeno
```

Po stisku <RETURN> počítač 2x zahouká a po dalším stisku klávesy bude program uložen na kazetu.

Duplicate T 2000 program - <D>

Pomocí této funkce kopírujeme nahrávky "klasickým Turbem". Po

14

stisku <D> postupujeme stejně, jako u funkce "Load T 2000 file". Jakmile je program nahrán, počítač vypíše hlášení "Save.." a 2x zahouká. Stiskem <RETURN> uložíme program na kazetu.

Copy data file - <C>

Jedna z nejužitečnějších funkcí TOSu. Slouží ke kopírování datových souborů z jednoho periferního zařízení na druhé. Volí se klávesou <C>. Po volbě se zeptá otázkou "From-" na zdrojové zařízení (zařízení, ze kterého budeme kopírovat). Zadáme název zařízení, případně i s názvem souboru a stiskneme <RETURN>. Soubor se nahraje do paměti. Nyní se počítač otázkou "To-" zeptá na druh cílového zařízení (zařízení, na které se bude datový soubor ukládat). Zadáme název cílového zařízení a stiskneme <RETURN>. Datový soubor se uloží na zvolené zařízení. Zdrojová a cílová zařízení můžeme libovolně kombinovat, případně mohou být (pokud to jejich druh dovolí) stejná. Dále jsou uvedeny některé z nejpoužívanějších kombinací: D:-D:, D2:-D:, D:-D8:, D8:-P:, D8:-E:. Možné jsou i kombinace např.: K:-D8, K:-P: a pod. V případě nemožné kombinace, např. D8:-K:, bude vypsáno chybové hlášení.

Při používání funkce "Copy data file" si musíme uvědomit, že máme-li v paměti nějaký program, či soubor dat, budou pravděpodobně přemazány kopírovaným souborem.

Basic/Ram -

Touto funkcí přecházíme z TOSu do Basicu, nebo její pomocí přepínáme paměť v oblasti \$A000-\$BFFF z ROM na RAM a naopak.

Po stisku je vypsána kontrolní otázka "Init (Y/N)". V případě "Y" bude inicializován Basic, tj. po krátké chvilce přejdeme do režimu Basic. V případě stisku "N" záleží na předchozím stavu paměti. Byla-li zapnuta ROM (Basic), přepne počítač na paměť RAM, tj. vypne Basic. Byla-li zapnuta RAM, přejde počítač do Basicu, avšak neprovede celkovou inicializaci

V Basicu máme k dispozici všechna definovaná zařízení. Největší výhodou jsou zařízení D: a D8:, která nám umožní efektivně používat příkazy LOAD, SAVE, RUN "D:", LIST, ENTER a pod., tj. příkazy, které při použití zařízení "C:" zabírají příliš mnoho času.

Z Basicu se do TOSu dostaneme příkazem "DOS". Po jeho zadání se normálně zobrazí menu TOSu. Program přitom zůstává zachován. Pro návrat do Basicu nyní raději použijeme funkci "Run cartridge". Stiskem <R> se dostaneme opět do Basicu. Kdybychom použili funkci "Basic/Ram", mohli bychom přijít o basický program v paměti.

Za názvem funkce "Basic/Ram" je vypsáno hlášení o aktuální konfiguraci paměti. Např. hlášení "now Basic" znamená, že je zapnut Basic, "now RAM" znamená, že je Basic vypnut a zapnuta RAM.

Poznámka: Někteří uživatelé doporučují ihned po spuštění TOSu inicializovat BASIC, opustit jej příkazem "DOS" a přepnout opět na RAM. V některých případech může tento postup zamezit pozdějšímu "zaseknutí" počítače.

Modifiky ramdisk - <M>

Tato funkce umožňuje měnit umístění a velikost ramdisku. Ramdisk je standardně umístěn v oblasti adres \$C000 až \$FFFF. Někdy můžeme potřebovat začátek ramdisku posunout trochu dále. V tom případě stiskneme <M> a na otázku "Range:\$" odpovíme např. \$D000. Ramdisk se posune do oblasti \$D000 až \$FFFF.

<RETURN>

Stiskneme-li na výzvu "SELECT ITEM" pouze <RETURN>, vymaže se obrazovka a znovu je zobrazeno menu TOSu. To je výhodné zejména tehdy, je-li toto menu již "odrolováno".

<RESET>

Stisk klávesy <RESET> nemaže TOS, ani ramdisk. Pouze bude přerušena běh programu a provedena částečná inicializace. Po inicializaci proběhne: návrat do TOSu, nebo je-li zapnut BASIC, tak do BASICu.

Poznámka: Některé verze TOSu vyžadují stisk <RESET> ihned po nahrání, aby proběhla dokonalá inicializace.

Pomocná hlášení

Mimo nabídky funkcí vypisuje TOS průběžně ještě pomocná hlášení. Dvě z nich jsme již probrali u funkcí "Basic/Ram" a "Modifiky ramdisk". Dalším je hlášení o poslední použité startovací adrese, např.:

Last run adr: \$1DF1

Dalším hlášením je:

LOMEM : \$1DE0.

Toto hlášení nám oznamuje nejnižší použitelnou adresu pro zavádění programů.

Mimo těchto hlášení vypisuje TOS ještě hlášení chybová.

Názvy souborů

U zařízení "D:" a "DB:" můžeme ještě za dvojtečkou uvádět

jména souborů. Jméno souboru může mít maximálně 12 znaků. Můžeme je rozdělit na dvě části. První z nich je vlastní název souboru. Druhá část se uvádí za tečkou a znamená vlastně poznámku o druhu souboru. Název souboru tedy může vypadat např. takto:

"D:KRUH.BAS",

kde tzv. extender BAS vyjadřuje, že se jedná o program v Basicu

Jména souboru uvádíme jak při ukládání, tak při nahrávání do počítače. V tom případě bude nahrán pouze soubor se zadaným jménem. Pokud neznáme celé jméno souboru, máme k dispozici tzv. divoké karty. Jedná se o znak "?" a "*". Otazník v názvu souboru nahrazuje 1 znak a to v místě jeho umístění. Hvězdička nahrazuje všechny znaky názvu, které jsou umístěny za ní, včetně její pozice. Jména souboru můžeme zadat např. takto:

"D:DO?IS".....("D:DOPIS")
"D8:*".

Ve druhém případě bude nahrán první datový soubor, na jehož počátek počítač narazí. Divoké karty nesmíme použít v názvu souboru při ukládání.

-js-

=====

Vážení čtenáři!

Jistě jste si všimli, že příspěvky z jednotlivých čísel zpravodaje pocházejí od velmi úzkého okruhu autorů. Není to tím, že by tito autoři chtěli stánky zpravodaje okupovat pouze pro sebe. Nemají k tomu žádný důvod, už proto, že doposud nebyl žádný z příspěvků honorován. Jako jedni z mála byli ochotni dát ataristům k dispozici své zkušenosti, či výsledky své práce, za což jim patří dík. Avšak i přes toto jejich úsilí má redakce stále potíže s náplní jednotlivých čísel. A to má náš zpravodaj pouhých 15 stránek!

Chci vás tedy touto cestou požádat, abyste nám když ne přímo příspěvky, zasílali alespoň náměty k náplni zpravodaje, svoje názory na jeho obsah, své připomínky. I to redakci pomůže v tom, aby vám byl náš zpravodaj užitečný.

Jiří Skála

=====

ATARI 602, technický zpravodaj pro mikroelektroniku a výpočetní techniku. Vydává 602. ZO Svazarmu pro potřeby vlastního aktivu. Redaktor: J. Skála. Adresa redakce: 602. ZO Svazarmu, Wintrova 8, 160 41 Praha 6. Telefon: 341 409. Povoleno ÚVTEI pod ev. číslem 87006. Cena .22 Kčs dle ČCÚ č. 1030/202/86. Náklad: 2.000 výtisků. Praha, prosinec 1989.