

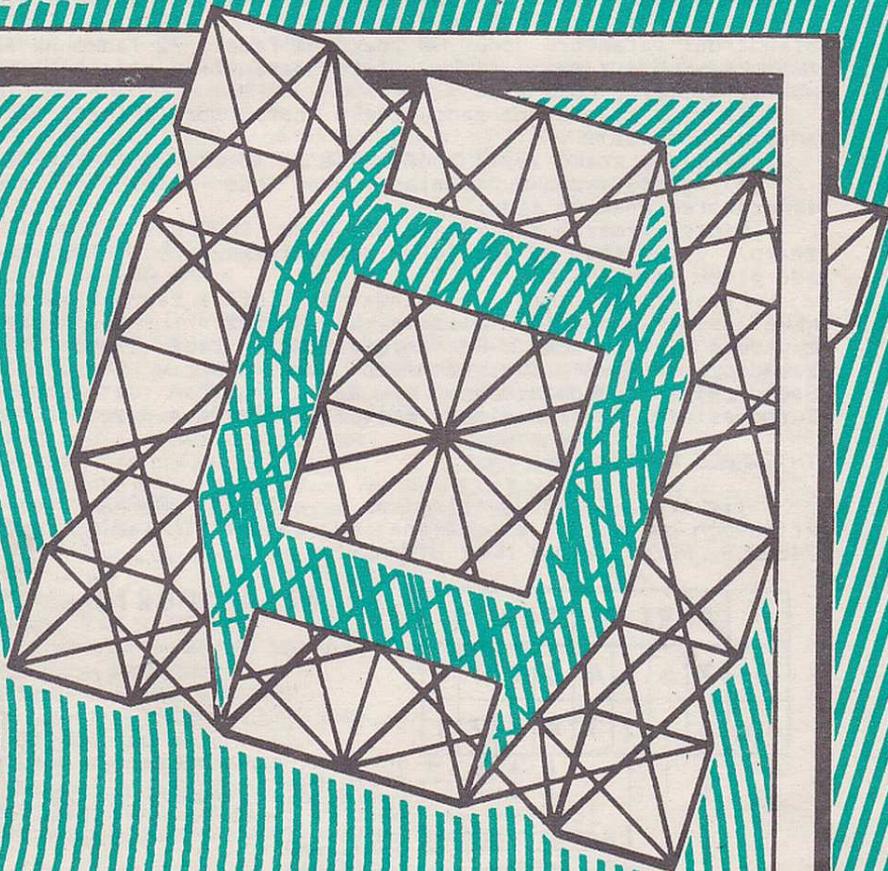


602

ATARI

2

89



## ----- PŘIPOJENÍ TISKÁRNY BT-100 K POČÍTAČI ATARI

BT-100 je připojena přes porty křížových ovladačů. S dále uvedeným software má tyto parametry:

Tisk oběma směry. Po vytištění posledního znaku, který není "space" se vozík okamžitě vrací. Respektuje příkazy pro tiskárnu v Atari basicu. Rozpozná příkazy "line feed", "formfeed". Po vypsání stránky čeká na výměnu papíru a po stisknutí START pokračuje v tisku. Totéž po "formfeed".

Standardní parametry jsou: 60 znaků na řádek, 72 řádků na formát, 1 linka - mezera mezi řádky. Tyto parametry můžeme libovolně měnit. Tiskárna může psát:

1) vždy standardní sadou znaků Atari. (bez ohledů na obsah adresy 756 deci),

2) sadou znaků jejíž počátek je na adrese 256\*PEEK(756),

3) sadou znaků implantovanou do řídicího programu generátorem znakové sady.

Řídicí program zabírá asi 920 Bytu (2000 B s interní sadou znaků). Je umístěn od adr \$0700 nepoužíváme-li DOS (TOS), nebo od adr \$1400 (pro TOS,DOS).

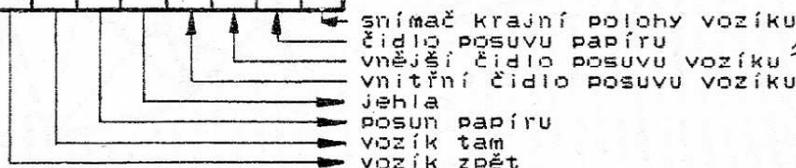
Program lze nahrát ve standardu: START a zapnout počítač. Po RESET se sám znovu inicializuje. Lze ho též přihrát v "turbu" do zavadece Turbo 2000. Po inicializaci předá řízení "turbo" zavadeči, a nyní lze nahrávat programy v turbu (např. Speedscript, Turbobasic apod.). Řídicí program je vyzkoušen s Turbobasicem, Atari basicem, Edit-assemblerem a Speedscriptem.

### INTERFACE BT-100

Tiskárna BT-100 je připojena k počítači ATARI přes porty křížových ovladačů. Je oddělena od počítače hradly 7403ALS a 7408ALS. POZOR OBVODY MUSÍ BÝT ALS!

	PORT 2				PORT 1			
bit	7	6	5	4	3	2	1	0
pin	4	3	2	1	4	3	2	1

### OBSAZENÍ PORTU



## Popis zapojení

Signály od čidel tiskárny přichází na obvod 7403ALS. Tento obvod je neauje a posílá na PORT1 počítače. Signály od počítače přes PORT2 přichází na obvod 7408ALS a jsou posílány do tiskárny. Interface musí být napájen z počítače (pin 7 = +5V, pin 8 = zem). V napájení je zařazen blokovací kondenzátor 47k.

## Stavba

Při stavbě interface se vyskytnou problémy pouze se sháněním konektorů. Konektory doporučuji zapájet do destiček plošných spojů, aby nedošlo k jejich prohození při propojení tiskárny a počítače.

## Oživení interface

Zhotovený a pečlivě zkontrolovaný interface připojíme ke zdroji 5V a změříme klidový odběr. Neměl by být větší než několik mA. Doporučuji před připojením k počítači ověřit funkci hradel. Budete-li postupně přivádět 5V na vstupy 7408ALS, bude se vám měnit úroveň výstupu z los.0 na los.1. Budete-li přivádět 5V na vstupy 7403ALS, pak zaregistrujete mírný pokles napětí na výstupech (nenaměříte los.1, protože 7403 má otevřený kolektor). Přes takto oživený interface můžeme připojit tiskárnu k počítači.

Nyní sejmeme spodní víko tiskárny a sejmeme gumové řemínku náhonu. Poté zapneme počítač a tiskárnu připojíme ke zdroji 22V a ověříme správnost funkcí.

### 1.Směr tiskárna-počítač. Napíšeme program:

```
10 POKE 752,1: POS,10,10: ? PEEK(54016): POS,10,10: ? " " :G,10
```

- a) čidlo "Vozík v kraj. poloze": zakryjete-li čidlo, pak bit0=0, odkryjete-li čidlo, pak bit0=1
- b) otočíme takt-kolečkem pro posun vozíku tak, aby nebyla zakrytá čidla, pak bit2 a bit3 jsou "1"
- c) otočíme takt-kolečkem pro posun vozíku tak, aby jedno čidlo bylo zakryto a jedno odkryto, pak bit2=1
- d) otočíme takt-kolečkem pro posun papíru tak, aby bylo čidlo odkryto, bit1 bude "1".

### 2.Směr počítač-tiskárna. Napíšeme program:

```
POKE 54018,56: POKE 54016,240: POKE 54018,60: POKE 54016,240
```

- a) pohon vozíku: POKE 54016,112. Motor vozíku se začne točit (při nasazeném řemínku by se vozík pohyboval směrem od čidla do druhé krajní polohy).

- b) POKE 54016,176. Motor změni směr otáčení.  
 c) posun papíru: POKE 54016,208. Motor posunu papíru se točí.  
 d) jehla: POKE 54016,224:POKE 54016,240. Jehla klepne.

Chodí-li vám takto interface, vypněte tiskárnu, poté počítač. Nasaďte řemínky a můžete se pustit do programové části.

## PROGRAMOVÉ VYBAVENÍ

### Generátor řídicího programu tiskárny

Program je vytvořen v Atari basicu. Nahrává se příkazem CLOAD. Po spuštění program nahrává data do paměti. Po bezchybném nahrání se ohlásí menu programu. Program se zeptá, chcete-li měnit parametry (počet řádků na formát, počet znaků na řádek, počet linek mezi řádky). Chcete-li parametry měnit, napišete R a stisknete RETURN. Nechcete-li měnit, napišete N. Dále se vás program zeptá, kterou verzi chcete vygenerovat:

v.1.0 - sada znaků z adr. 256\*PEEK(756),

v.1.1 - standart Atari sada,

v.2.0 - verze pro generátor znakové sady.

Podle výběru verze stiskneme 1, 2 nebo 3, a Return. Nyní počítač 2x zahouká. Zmáčkní PLAY, REC a RETURN. Řídicí program se vygeneruje. Takto vytvořený program zavádíme do počítače přes tlačítko START (pro verze v.1.0, v.1.1).

### Generátor znakové sady

Program je napsán v TURBOBASICU. Po nahrání Turbobasicu jej nahrajeme obvyklým způsobem. Spustíme příkazem RUN. Zobrazí se menu programu.

L-load externí znakové sady (organizace sady: 1024 B),

S-nahrání vytvořené sady na ms

R-load - nahraje řídicí program tiskárny (RPT). Pokud chceme vytvořit RPT s vlastní sadou, nahrajeme v.2.0. Nahrajeme-li v.1.0 nebo v.1.1, můžeme tiskárnu inicialisovat

G-generování sady znaků. Zmáčknete-li libovolnou klávesu, pak se v levé polovině objeví znak k této klávese příslušející a jeho hexa struktura. Nyní můžeme tento znak předefinovat. Šipky ovládáme klávesami -=>\*, bod zobrazíme ".", prázdné pole ". ". Po vytvoření celého znaku stisknete RETURN a můžete měnit další znak. Takto můžete vytvořit celou vlastní sadu. Chcete-li předefinovat pouze některé znaky, tak před generováním provedte přesun Atari sady do G2S klávesou M.

@-vygenerování RPT s vnitřní znakovou sadou

B-přechod do Turbobasicu. Zde si můžete vytvořenou sadu prohlédnout zadáním POKE 756,112

## Generátor RPT GR.8 BT-100

Program je napsán v Atari basicu. Po spuštění programu se na ma pásku vygeneruje rutina, která umožňuje kopírování obrazovky v grafice 8.

Tato rutina se zavádí do počítače přes tlačítko START. Spouští se příkazem A=USR(1538). Rutina zabírá celou 6. stránku paměti. Ukončení tisku zadáme stiskem HELP. Zpět do Basicu se dostaneme přes RESET. Při tomto RPT tiskárna netiskne v obou směrech.

## POZNÁMKY

1) Chcete-li vytvořit RPT od adresy \$1400, pak použijte v generátoru RPT změny V+. Generátor znakové sady funguje i pro verze V+ mimo inicializaci tiskárny.

2) Tiskárnu lze inicializovat příkazem A=USR(2116), verze V+ A=USR(5444).

3) Vytvoření turbo verze: programem TM2003:

- Cboot
- Turbosave od adr 1914 (5242)  
počet B 800 (800)  
start 1945 (5273)
- RETURN

Takto vytvořený program nahrajeme zavaděčem TURBO 2000. Po inicializaci je předáno řízení opět turbo zavaděči

4) Chcete-li program použít s TOS, musíte TOS změnit takto: po natažení TOS do TM2003 funkcí M změním obsah adresy 4162 z 20 na 24 resp.28 při v.2.0. Takto změněný TOS nahrajeme na maf funkcí R. Při přechodu do basicu musíme tiskárnu zinicilizovat příkazem A=USR(5444). Na verzi TOS, jejíž součástí bude i RPT, se pracuje.

5) Následuje-li v basicu více příkazů LPRINT "....."; pak program ukončíme příkazem LPRINT CHR\$(155). Jinak se nám nevytiskne poslední řádek.

Zavěrem bych chtěl poznamenat, že tiskárna BT-100 je zdařilý výrobek. Jedinou nevýhodou BT-100 je pomalost tisku: 150 bodů za sekundu, což reprezentuje 2,5 znaku.

Bohumil JIRSIK

Pozn. red.: Z technických a kapacitních důvodů bohužel nemůžeme uveřejnit výpisy všech popisovaných programů. Výpis základního RPT naleznete na dalších stránkách. Ostatní popisované programy budou k dispozici na pravidelných burzách programů v našem klubu.

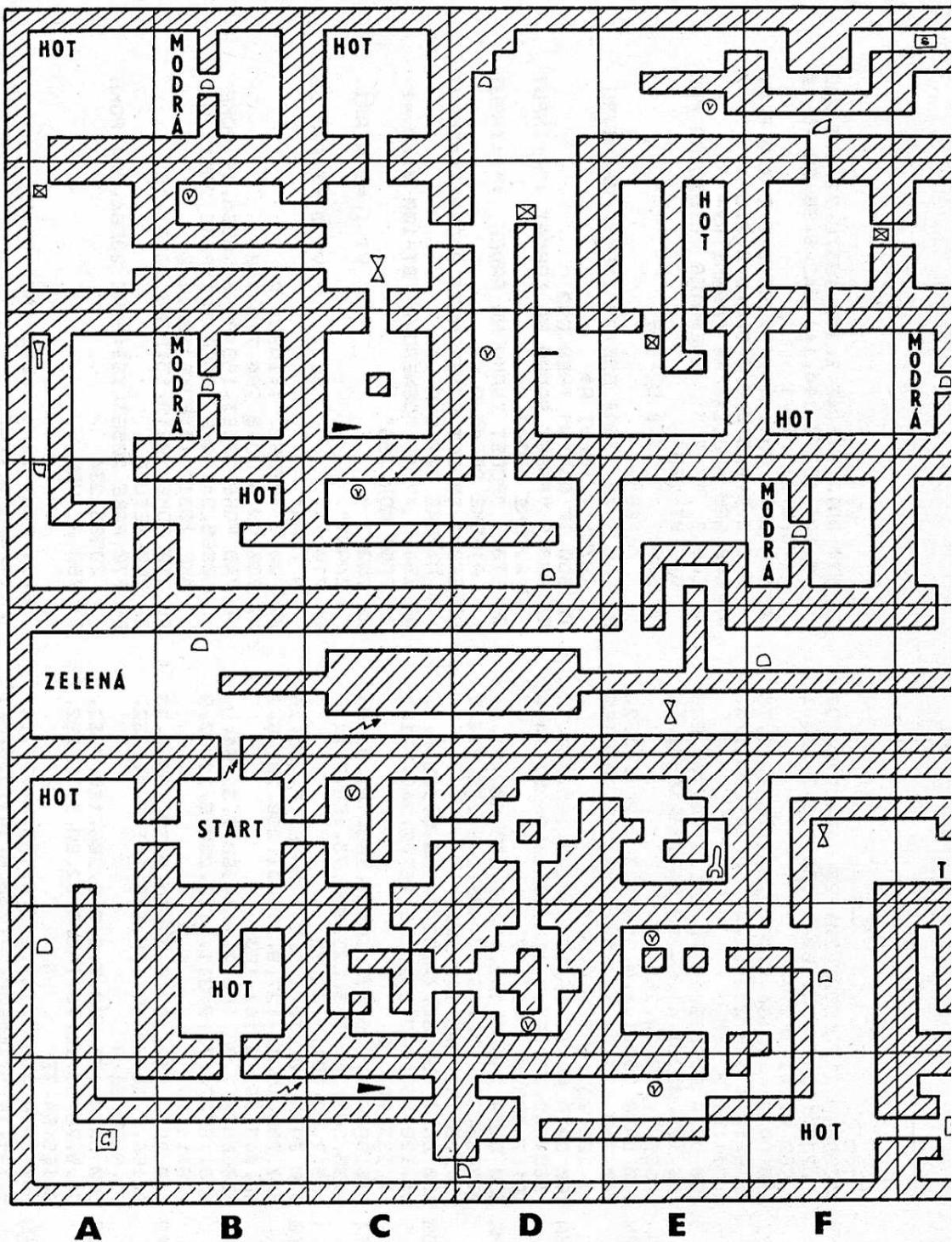
DA 20 REM GENER. RIDICIHO. PROGR. BT-100  
 DM 30 REM  
 BB 40 REM  
 LF 50 REM \* BASIC/STROJOVY JAZYK \*  
 ED 60 REM  
 IE 70 REM \*\*\*\*\*  
 ZX 80 REM \* JBC-soft (c)89 \*  
 AA 91 REM \*  
 AG 92 REM \* BOHUMIL JIRSIK PRAHA \*  
 AE 93 REM \*  
 IO 94 REM \*\*\*\*\*  
 SX 150 REM  
 QX 160 ? :? "K":? "JBC-soft  
 (c)89"  
 XJ 170 POKE 752,1:POSITION 2,10:?"CTENE  
 DATA":DIM A\$(1)  
 XB 180 FOR X=0 TO 800:READ A:SUMA=SUMA+A:  
 POKE 20000+X,A:POSITION 25,10:?"X:NEXT  
 X  
 MJ 190 IF SUMA<>80776 THEN 240  
 TO 200 GOSUB 620  
 HY 210 OPEN #4:8,128:"C":POKE 900,32:POKE  
 E 901,78:POKE 904,52:POKE 905,3:POKE 8  
 98,11  
 GM 220 A=USR(ADR("H#L#LV#"),64):CLOSE #4  
 NX 230 END  
 GH 240 ? "CHYBA DAT":END  
 LS 260 REM \*RIDICI PROGRAM TISK BT-100\*  
 HN 280 DATA 0,7,122,7,69,8,169,60,141,0,2  
 11,169,139,141,231,2,169,10,141,232,2,  
 169,69,133,10,169  
 UL 290 DATA 8,133,11,24,96,32,69,8,76,197  
 ,5,0,0,253,0,0,0,72,8,60,1,0,240,186,7  
 ,206  
 WR 300 DATA 7,206,7,209,7,206,7,206,7,76,  
 187,7,0,169,56,141,2,211,169,240,141,0  
 ,211,169,60,141

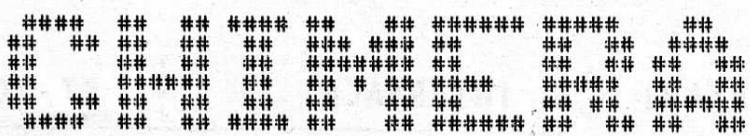
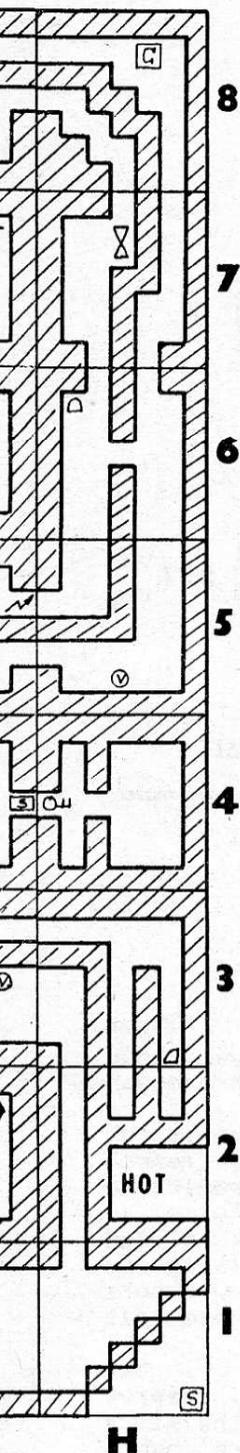
SB 310 DATA 2,211,169,240,141,0,211,160,1  
 ,96,201,12,240,15,201,10,240,20,201,13  
 ,240,91,201,155,208,39  
 WC 320 DATA 76,57,8,32,0,8,32,246,9,76,20  
 7,7,173,44,8,201,0,240,3,32,0,8,169,8,  
 32,226  
 MA 330 DATA 9,76,207,7,169,0,141,44,8,32,  
 116,8,96,72,41,96,201,96,240,26,201,32  
 ,240,10,201,64  
 KX 340 DATA 240,12,104,9,64,76,43,8,104,4  
 1,223,76,43,8,104,73,96,76,43,8,104,14  
 1,0,7,238,44  
 BQ 350 DATA 8,173,44,8,205,167,7,208,150,  
 169,0,141,44,8,32,116,8,76,207,7,104,1  
 69,171,141,27,3  
 XB 360 DATA 169,7,141,28,3,169,139,141,23  
 1,2,169,10,141,232,2,169,0,141,162,7,1  
 41,164,7,141,163,7  
 BW 370 DATA 162,60,157,255,6,202,208,250,  
 169,253,141,161,7,76,187,7,173,14,9,20  
 1,200,240,3,32,79,10  
 LK 380 DATA 173,0,211,41,1,208,21,169,112  
 ,141,0,211,32,72,9,32,88,9,238,159,7,1  
 73,159,7,201,3  
 XR 390 DATA 208,240,169,176,141,0,211,173  
 ,0,211,41,1,208,249,32,96,9,169,0,141,  
 159,7,32,133,9,160  
 KB 400 DATA 0,162,0,169,112,141,0,211,32,  
 72,9,32,88,9,32,80,9,185,0,7,41,128,24  
 0,5,169,176  
 CA 410 DATA 76,209,8,169,144,141,219,8,18  
 5,60,7,42,153,60,7,144,5,169,224,76,22  
 8,8,169,240,141,170  
 HH 420 DATA 7,173,0,211,41,8,240,3,238,15  
 9,7,32,64,9,173,170,7,45,0,211,141,0,2  
 11,9,16,141  
 FV 430 DATA 0,211,232,224,8,208,26,162,0,  
 204,164,7,240,22,200,192,255,240,17,17  
 3,0,211,41,1,240,10

MY 440 DATA 173,159,7,201,24,240,3,76,192  
 ,8,238,162,7,173,162,7,205,166,7,208,1  
 1,169,0,141,162,7  
 JI 450 DATA 141,164,7,76,223,9,76,6,10,76  
 ,126,10,173,0,211,41,4,240,249,96,173,  
 0,211,41,8,240  
 KO 460 DATA 249,96,173,0,211,41,4,208,249  
 ,96,173,0,211,41,8,208,249,96,169,1,14  
 1,169,7,240,24,169  
 BN 470 DATA 208,141,0,211,173,0,211,41,2,  
 240,249,173,0,211,41,2,208,249,206,169  
 ,7,208,237,169,240,141  
 SA 480 DATA 0,211,96,172,162,7,165,203,72  
 ,165,202,72,162,0,134,203,169,253,141,  
 161,7,169,0,7,41,127  
 RA 490 DATA 240,3,142,164,7,133,202,24,6,  
 202,38,203,238,161,7,208,246,173,244,2  
 ,5,203,133,203,177,202  
 DS 500 DATA 157,60,7,192,7,208,8,189,0,7,  
 41,128,157,0,7,232,236,167,7,240,7,169  
 ,0,133,203,76  
 TZ 510 DATA 146,9,173,164,7,168,162,0,104  
 ,133,202,104,133,203,96,173,168,7,141,  
 169,7,32,101,9,32,79  
 IS 520 DATA 10,238,163,7,173,163,7,205,16  
 5,7,208,15,32,127,9,173,31,208,201,6,2  
 08,249,169,0,141,163  
 RF 530 DATA 7,96,169,0,141,160,7,32,64,9,  
 238,160,7,173,0,211,41,8,208,6,32,80,9  
 ,76,11,10  
 PR 540 DATA 206,160,7,32,96,9,32,133,9,23  
 8,164,7,32,79,10,169,176,141,0,211,32,  
 72,9,32,64,9  
 IF 550 DATA 173,160,7,240,9,206,160,7,32,  
 80,9,76,53,10,76,192,8,32,80,9,76,192,  
 8,169,64,77  
 FQ 560 DATA 215,8,141,215,8,169,64,77,14,  
 9,141,14,9,169,32,77,238,8,141,238,8,1  
 73,46,9,73,5

JG 570 DATA 141,46,9,173,193,8,72,173,242  
 ,8,141,193,8,104,141,242,8,96,32,79,10  
 ,173,0,211,41,1  
 UD 580 DATA 208,249,76,166,8,0,0,0,0,0,  
 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0  
 FH 600 REM \*VDLITELNE PARAMETRY\*  
 AL 620 ? "K":? " GENERATOR RIDICHO PRO  
 GR. BT-100"  
 NG 630 ? :? "J B C -Soft  
 (C)89"  
 RD 640 ? :? :TRAP 620:?" ZMENY PARAMETRU  
 (A/N)":? INPUT A\$  
 TL 650 IF A\$="N" THEN 690  
 GS 660 ? "POCET RADKU NA FORMAT ":?:INPUT  
 A:POKE 20043,A  
 DH 670 ? "POCET ZNAKU NA RADEK ":?:INPUT  
 A:POKE 20045,A  
 NG 680 ? "POCET LINEK MEZI RADKY":?:INPUT  
 A:POKE 20046,A  
 GG 690 ? :? : "GENEROVAT BT-100 V1.0 =1  
 SADA ADR.756"  
 QL 700 ? "  
 ZNAMY"  
 WM 710 ? "  
 N.ZNAK"  
 ZZ 720 ? "VERSE ":?:INPUT A  
 YZ 730 ON A GOTO 780,770,740  
 UB 740 POKE 20563,169:POKE 20564,11:POKE  
 20565,24:POKE 20566,101:POKE 20012,0:P  
 OKE 20017,15:POKE 20214,0  
 PE 750 POKE 20219,15:POKE 20001,15  
 ZP 760 RETURN  
 BF 770 POKE 20563,169:POKE 20564,224:POKE  
 20565,234  
 ZT 780 RETURN

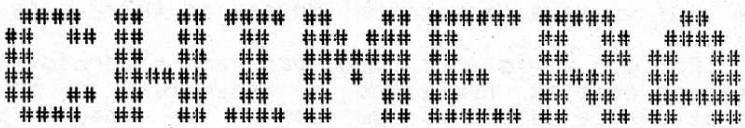
Pozn.: Podtržene znaky zadavat  
 i n v e r z i e !

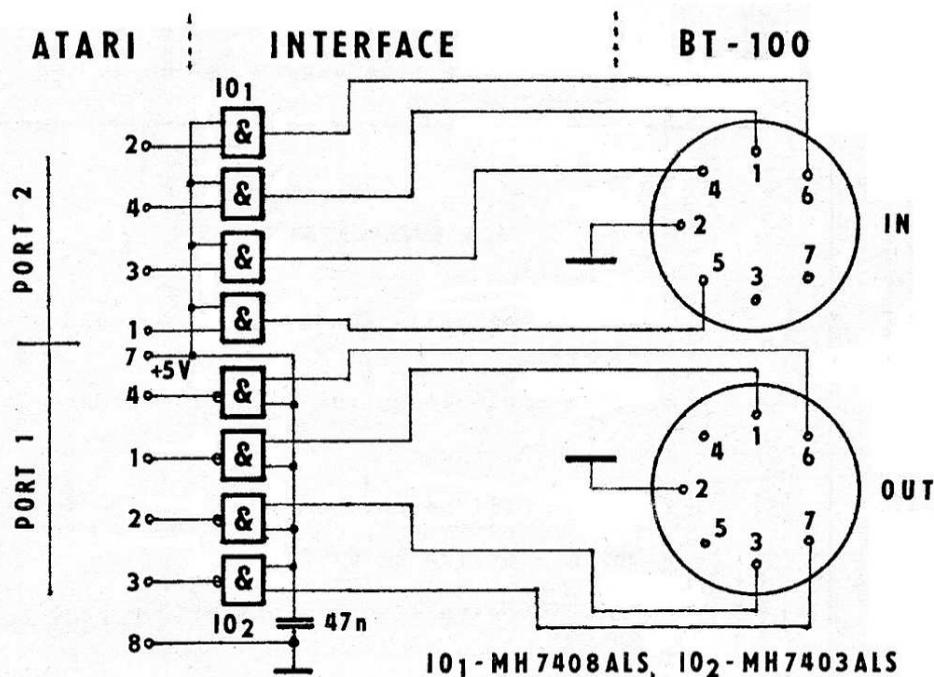




**LEGENDA :**

-  - hlavice (raketa) - odnáší se do modrých místností
-  - chléb - doplňuje energii a ničí sporáky
-  - voda - ochlazuje
-  - sporák - ničí se chlebem
-  - krabice - otvírá se klíčem
-  - dveře - otevírají se stejným klíčem jako krabice
-  - CML (přesýpací hodiny) - ničí se Jehlanem
-  - energetická bariéra - likviduje se francouzským klíčem
-  - počítač
-  - klíč - otevírá dveře a krabice
-  - francouzský klíč - likviduje energetické bariéry
-  - baterka - bude potřeba v temné místnosti
-  - Jehlan - likviduje CML
- HOT** - radiátor





## U Ž I V A T E L S K E   P R O G R A M Y

### TEXGEN 1.0

Často potřebujeme již hotové programy ve strojovém kódu opravit, upravit nebo doplnit určitou sekvencí strojových instrukcí či daty. V takovém případě máme dvě možnosti:

- Editovat strojový program pomocí vhodného monitoru. Vhodným monitorem je monitor disponující mimo jiné též funkcí přemístění obsahu paměti na jiné místo zároveň s přepočítáním operandů (adres) u tříbajtových instrukcí. Takovým monitorem je například DEBUG MONITOR 2. Editace programu je však i tak zdlouhavá, náročná, je nutné dávat pozor, aby se nezměnily adresy HW registrů, apod. Navíc jsme omezeni pouze na adresový prostor, který není využíván monitorem! Adresy od \$A000 výše jsou pro nás "tabu".

- Pořídít výpis strojového programu - zdrojový text -, ten přenést do editoru některého z assemblerů a zde jej teprve editovat. Práce se zdrojovým textem je značně pohodlnější a rychlejší, navíc můžeme editovat i takové programy, které budou

umístěny v jinak nedostupných oblastech paměti. Jedinou nevýhodou tohoto způsobu je pracné pořizování zdrojového textu. Tuto zdlouhavou práci nám výrazně ulehčí program TEXTGEN.

## Popis programu

Program TEXTGEN 1.0 pořizuje zdrojový text strojových programů. Text je generován ve formátu vhodném k načtení do textového editoru assembleru ATMAS II a ATMAS III. Je nutné jej uložit na kazetu či disketu, při používání TOSu můžeme ukládat i do RAMdisku (ATMAS II).

Aby mohl program TEXTGEN pracovat, je nutné nahrát do paměti počítače zpracovávaný strojový program. Proto ihned po svém spuštění TEXTGEN aktivuje funkci L - "load file" a očekává zadání počáteční adresy programu (adresy, od níž je program normálně umístěn) a specifikaci periferního zařízení (D:,T:FNAME.EXT,D:FNAME.EXT, apod.). Po nahrání programu se vypíše počáteční a koncová adresa, dále pak délka souboru, a zobrazí se otazník. Program očekává další příkazy.

Nyní máme k dispozici tyto funkce: D, M, L, SHIFT CLEAR(mazání obrazovky), U, a T.

### D - disassembler

Tato funkce je velmi důležitá. Umožní nám prohlédnout si nahraný strojový program a vypsat si jeho zpětný překlad na obrazovku. Před vlastním generováním zdrojového textu funkcí T musíme totiž zadat programovou oblast, ve které se nacházejí data nebo strojové instrukce, o tom však později. Po vyvolání disassembleru zadáme adresu, od níž chceme výpis na obrazovku. Adresa musí být v rozsahu adres bufferu (nahraného strojového programu). Provádění výpisu můžeme kdykoli přerušit klávesou "ESC", stejně jako můžeme tímto způsobem přerušit běh i všech ostatních funkcí. Objeví-li se ve výpisu strojového programu místo názvu strojové instrukce tři otazníky - ??? - , znamená to, že na této adrese není strojová instrukce, ale data. Všechny tyto adresy si musíme poznamenat pro pozdější použití.

### M - memory dump (monitor)

Funkce M vypisuje na obrazovku obsah bufferu od zadané adresy ve formátu HEX ... ASCII. Pro její použití platí stejné zásady, jako u funkce D.

### U - user table - nastavení dekompileční tabulky

Jak již bylo uvedeno, před vlastním generováním zdrojového textu funkcí T musíme zadat adresy, na kterých se nacházejí strojové instrukce a adresy, kde se vyskytují data. To nám umožní funkce U - definice dekompileční tabulky.

Při definování vlastní tabulky máme k dispozici tyto funkce:

I - instructions - pomocí této funkce zpětnému překladači sdělujeme, ve kterých oblastech paměťového bufferu má hledat strojové instrukce.

B - byte - tato funkce udává, která oblast paměťového bufferu má být chápána jako oblast jednobajtových dat. Ve výsledném zdrojovém textu budou tato data parametrem pseudoinstrukce "DFB". Funkce B se použije i pro vymezení textu.

W - word - funkce má podobný význam jako funkce B, avšak data budou ve výsledném zdrojovém textu parametrem pseudoinstrukce "DFW".

S - show table - zobrazí definovanou dekompilační tabulku.

N - new table - dekompilační tabulku vymaže.

F - free memory - zobrazí okamžitý počet bajtů využitelných pro definici tabulky.

Při definování dekompilační tabulky je nutné dodržet několik zásad:

- Oblasti definované funkcí I nesmějí obsahovat překladači neznámé strojové instrukce. V opačném případě nebude zpětný překlad do zdrojového textu dokončen a bude signalizována chyba. (Takovou instrukcí je např. instrukce ADC IND(Y) s oper. kódem \$71. Tuto instrukci ale nazná ani ATMAS.)

- V oblasti strojových instrukcí nesmějí být skoky do oblastí definovaných funkcí B nebo W. Pokud program takové skoky obsahuje, lze předpokládat, že data obsahují strojovou rutinu, kterou jsme přehlédli. V opačném případě můžeme celý program rozdělit, provést zpětný překlad u každé části zvlášť a potom je všechny v ATMASU spojit v jeden celek.

- Adresy definovaných oblastí se nesmějí překrývat. Udávají se vždy jako uzavřený interval, tj. -od-, -do-. Rozsahy definované funkcí W musí obsahovat sudý počet bajtů, rozsahy strojových instrukcí končí posledním bajtem poslední instrukce.

- Jednotlivých oblastí I, nebo B, či W může být libovolný počet, musí však zůstat prostor pro tabulku, generovanou při vlastním překladu.

T - text generator

Použitím funkce T vrcholí práce na zpětném překladu do zdrojového textu. Jejím vyvoláním spustíme zpětný překlad. Ten

probíhá ve třech fázích. V první fázi si překladač vytváří tabulku odkazů, kterou později využije při tvorbě návěstí. Zároveň ošetří chyby (neznámé strojové instrukce, nepostačující paměť apod.). Narazí-li na chybu, vypíše příslušné hlášení a přeruší práci. V tom případě je nutné ověřit a případně předefinovat dekompilací tabulku nebo jiným způsobem chyby odstranit. Proběhne-li první fáze bezchybně, spustí se fáze druhá. Před tím však program požádá o specifikaci výstupního zařízení, na které chceme zdrojový text uložit. Vzhledem k nedostatku paměti jej nelze uložit přímo do paměti počítače a proto je rovnou při tvorbě ukládán na výstupní zařízení. Jako výstupní zařízení však není vhodné používat tiskárnu. Zdrojový text je generován ve formátu pro editor ATMASU, a proto by byl při použití tiskárny vtištěn velmi nepřehledně. Během druhé fáze se na specifikované zařízení vypíše deklarace návěstí, která nelze z nějakých důvodů nadefinovat přímo v textu. Třetí fáze na specifikované zařízení vypisuje vlastní zdrojový text.

#### Poznámky k programu:

- Narazí-li funkce M a D na konec bufferu, zastaví výpis a čekají na stisk klávesy "ESC".
- KTEROUKOLI funkci lze KDYKOLI přerušit stiskem klávesy "ESC". Je-li již nějaký strojový program nahrán v paměti, lze funkci L opustit ještě v průběhu zadávání nahrávací adresy. Později již NE!
- Nelze očekávat, že okamžitě po načtení zdrojového textu z TEXTGENU do ATMASU můžeme program relokovat, a že po překladu bude "chodit". Například jsou adresy často předávány v registrech. TEXTGEN nemůže poznat nejedná-li se o data, a proto ještě před zpětným překladem v TEXTGENU vytipovat a poznamenat si je. Nejčastěji se jedná o adresy tabulek dat.
- Práci s programem TEXTGEN ukončíme stisknutím RESET, klávesa BREAK je blokována. Chceme-li zpracovat další strojový program, vyvoláme funkci L.

#### Mapa paměti

TOS/TEXTGEN/BUFFER/US.TABLE/Tab.odkazu.../

Poznámka na závěr: TEXTGEN 1.0 vznikl z okamžité nutnosti mít v co nejkratším čase k dispozici program, urychlující a usnadňující práci se strojovým kódem. Není proto dokonalý, chybí řada funkcí, které by takový program měl obsahovat. Je napsán v Basicu, což je příčinou relativní pomalosti zpětného překladu. Basic však byl v dané chvíli nevhodnějším jazykem pro rychlé odladění a experimentování. I přes zmíněné "nedostatky" se však domnívám, že může být řadě uživatelů počítačů ATARI XL/XE užitečný.

Martin Flechšmid

V minulém čísle našeho zpravodaje jsme vám nabídli návod, jak pomocí počítače ATARI měřit elektrické napětí, případně další veličiny, které jsou přetransformovány na jeho odpovídající hodnotu. Dnes vám nabízáme další návod z oblasti měření elektrických veličin pomocí počítače ATARI. Jedná se o způsob měření frekvence elektrických signálů.

#### CITAC

-----

Program, jehož výpis naleznete na další stránce, pracuje jako čítač. Počítá impulsy na vstupu (vývod č.1 joysticku 1) za určený čas. Zpracovávané pulsy musí mít úroveň běžnou u obvodů TTL. Pro reálná měření je vhodné použít stejný tvarovač, jaký používá TURBO 2000 (integrovaný obvod MAR741, MAC... apod.); pro měření vyšších frekvencí lze připojit předděličku.

Platí zásada: čím je měřená frekvence vyšší, tím se více uplatní vliv zdržení činnosti procesoru zobrazováním a dalšími operacemi, probíhajícími v přerušení. Proto je pro vyšší kmitočty vhodné používat delší dobu měření. Pro nízké kmitočty srovnatelné s dobou čítání (jednotky Hz při  $T=1s$ ) se uplatní vliv nezávislosti startu čítání na měřených pulsech - chybí synchronizace.

Z hlediska praktického použití lze programu mnohé vytknout. Nebyl ovšem navržen jako náhrada drahých přístrojů, ale jako ukázka schopností našeho počítače. I tak však program pro informativní měření frekvencí postačuje.

Základem programu je strojová rutina, která testuje stav vstupního portu. Pro zachycení obou logických úrovní jsou připraveny čekací smyčky. Po ukončení každého pulsu se zvyšuje stav čítače frekvence o 1. Doba čítání je určena systémovým čítačem CDTMV3 a je tudíž odvozena z krystalu počítače. Doba stanoví počáteční obsah čítače (adresy \$21C,\$21D). Každou  $1/50$  s je tento čítač dekrementován. Při dosažení nuly se během VBI přepíše obsah CDTMF3 (\$22A) z \$FF na \$00. Poté rutina předá řízení BASICu. Interpret převezme počet impulsů automaticky do proměnné A funkce  $A=USR(adr)$ , protože jsou pro čítač použity buňky \$D4,\$D5. Program v BASICu zajišťuje již jen komfort obsluhy a snadnou možnost změny parametrů čítání.

#### Použitá literatura :

- 1) Zabka A., RNDr. / Meranie frekvencie počítačom ZX Spectrum, Sdělovací technika č.3/88, s.110
- 2) Kodera J., inž. / Komentovaný výpis ROM

```

EM *****
REM.*          CITAC          max.do 20 kHz          *
REM.*-----*
REM.* Max.frekvence zavisi na dobe citani T.          *
5 REM.* VSTUP (pin1 joysticku) zpracovava signaly o urovni TTL. Je vhodna delicka a tvarovac *
6 REM.*          *
7 REM.*          ) made by OK1UXX -c- 1989 (          *
15 REM *****
17 ? CHR$(125):? :? :? :? :? " Inicializace ..."
18 FOR I=16377 TO 16446:READ A:POKE I,A:NEXT I:POKE 556,0
19 T=1:H=0:REM nastaveni pocatecnich parametru. T=doba mereni, H=doba mezi dvema cykly mereni.
20 GRAPHICS 1:POKE 752,1:POKE 710,0:POSITION 2,9:? #6;"<";POSITION 8,11:? #6;"HERTZ"
21 POSITION 2,2:? #6;"CITAC          0-20KHZ"
22 POSITION 2,8:? #6;"-----":POSITION 2,10:? #6;"-----"
23 ? CHR$(125);" Zmeny parametru ... HELP"? :? "          * made by OK1UXX *":POSITION 14,2:? #6;"20"
24 X=T/0.02:HB=INT(X/256):LB=X-HB*256:POKE 16390,HB:POKE 16395,LB:IF T<0.6 THEN POSITION 14,2:? #6;"10"
25 X=H/0.02:HBH=INT(X/256):LBH=X-HBH*256
26 F=INT(256*256/T/1000):IF T>1 AND F(20 THEN POSITION 14,2:? #6;" ":POSITION 14,2:? #6:F
27 POSITION 1,14:? #6;"T=";T;"SEC":POSITION 13,14:? #6;"H=";H;"SEC"
28 POKE 732,0:A=USR(16377):X=5:A=A/T:REM vlastni mereni a prepocet dle doby mereni T
29 IF A(10 THEN X=9
30 IF 10<=A AND A(100 THEN X=8
31 IF 100<=A AND A(1000 THEN X=7
32 IF 1000<=A AND A(10000 THEN X=6
33 IF T>1 THEN X=X-1
34 POSITION 3,9:? #6;"          ":REM Konec vypoctu pozice tisku hodnoty frekvence (A).
35 POSITION X,9:? #6:A:IF H<0 THEN POKE 556,255:POKE 543,HBH:POKE 542,LBH
36 POSITION 13,15:? #6;"hold":IF PEEK(732)=0 THEN IF PEEK(556)=255 THEN 38:REM cekani na HOLD ci HELP
37 POSITION 13,15:? #6;" ":IF PEEK(732)=0 THEN 30:REM neni-li HELP, novy cyklus citani.Jinak zmeny
38 POSITION 4,9:? #6;"stop! ":? CHR$(125):"OPTION (+/-) ... cas mereni"
39 ? "SELECT (+/-) ... cas mezi merenim"? "START          ... spusti mereni"
40 P=PEEK(53279):IF P=6 THEN POSITION 3,9:? #6;"          ":GOTO 25:REM start mereni - nove par.
41 IF P=3 THEN 50:REM OPTION - time => nastaveni doby mereni T (od 0.1s po 0.1s od 1s po 1s)
42 IF P=5 THEN 60:REM SELECT - hold => nastaveni doby mezi cykly mereni H (po 1s od 0)
43 POSITION 1,14:? #6;"T=";T;"SEC          ":POSITION 13,14:? #6;"H=";H;"SEC          "
44 GOTO 45:REM znovu na test konzoly
45 REM TIME - nastav T
46 IF PEEK(764)=6 AND T=1 THEN T=T+1
47 IF PEEK(764)=6 AND T(1 THEN T=T+0.1
48 IF PEEK(764)=14 AND T(=1 AND T)0.1 THEN T=T-0.1
49 IF PEEK(764)=14 AND T)1 THEN T=T-1
50 POKE 764,255:IF PEEK(764)()255 THEN 55
51 SOUND 1,10,10,8:I=2*2:SOUND 1,0,0,0:GOTO 48:REM skok na tisk novych parametru
52 REM HOLD - nastav dobu mezi cykly
53 IF PEEK(764)=6 AND H(20 THEN H=H+1
54 IF PEEK(764)=14 AND H)0 THEN H=H-1
55 POKE 764,255:IF PEEK(764)()255 THEN 63
56 SOUND 1,30,10,8:I=2*2:SOUND 1,0,0,0:GOTO 48:REM skok na tisk novych parametru
57 END
100 DATA 104,169,255,141,42,2,169,0,133,212,133,213,169,0,141,29,2,169,50,141,28,2
101 DATA 173,0,211,41,1,201,1,240,6,173,42,2,208,242,96,230,212,208,2,230,213
102 DATA 173,42,2,208,1,96,173,0,211,41,1,201,1,240,6,173,42,2,208,215,96
103 DATA 173,42,2,208,236,96

```

## J A K N A T O ?

### CHIMERA

---

Okolo naší planety Země obíhá podivná kosmická loď. Nemá posádku, avšak je vybavena množstvím různých zajímavých zařízení. Tvým úkolem je: dostat se na palubu lodi a zmocnit se čtyř hlavic, které ti pomohou rozšifrovat zakódovanou zprávu, určenou pro obyvatele Země.

Než budeš moci rozluštit zprávu pro obyvatele Země, budeš muset překonat mnoho obtíží. Jediné, co ti autoři této hry ulehčili je vstup na palubu lodi.

Hra začíná v místnosti B3. Musíš se postupně dostat do 4 modrých místností (Blue room), ve kterých přečteš zakódované zprávy. Nejdříve musíš získat klíč k odstraňování energetických bariér, ukrytý v místnosti E3. Hledané hlavičky jsou v místnostech C1, C7, C6 a G3. Po zlikvidování energetických bariér vezmi první hlavičku z C1 a odnes ji do místnosti F5 (v levém dolním rohu obrazovky je zobrazen předmět, který právě neseš; předměty zvedni, polož nebo použij stiskem FIRE; chléb a vodu raději nepřinášej; při pokládání hlavic se nepřibližuj ke chlebu a vodě).

Nyní zlikviduj radiátory v místnostech G8 a G4. Nejdříve v G8. Napij se vody a zruš radiátor v G4. Za ním je ukrytý klíč. Tento klíč otevírá dveře v místnostech D1, F8, H3 a A5, a krabice v místnostech E7, E6, D7 a A7. Po otevření dveří a krabic můžeš vzít další hlavičky z místnosti G7 a zanést do některé z volných modrých místností. Také můžeš vzít lampičku z A6 a zanést ji do místnosti G3 (temná místnost). Zde lampičku rozsviř. Najdeš další hlavičku. Můžeš zde vypít také vodu, ale ještě před rozsvícením lampičky.

Další tvoje cesta by měla vést do místnosti G2, kde je ukryt Jehlan. Tento Jehlan likviduje CML v místnostech F3, E4, H7 a C7. Po likvidaci CML v C7 se dostaneš do C6, kde je ukryta poslední hlavička. Odnes ji do poslední volné modré místnosti. Abyš dokončil hru, zbývá ti již jen dojít do zelené místnosti A4.

Při cestování po kosmické lodi musíš při vhodných příležitostech pít vodu a jíst chléb. V horkých místnostech se zdrž co nejméně. Ne všechny předměty, které po cestě najdeš, musíš sebrat, nebo použít. Vypracuj si podle plánu kosmické lodě nejoptimálnější cestu.

(podle Bajtku připravil -js-)

=====

ATARI 502, technický zpravodaj pro mikroelektroniku a výpočetní techniku. Vydává 602. ZO Svazarmu pro potřeby vlastního aktivu. Zodpovědný redaktor: J. Skála. Adresa redakce: 602, ZO Svazarmu, Wintrova 8, 160 41 Praha 6. Telefon: 341 409. Povoleno ÚVTEI pod ev. číslem 87006. Cena 9 Kčs dle ČCÚ č. 1030/202/86. Náklad 500 výtisků. Praha, únor 1989

**Igi/2024**