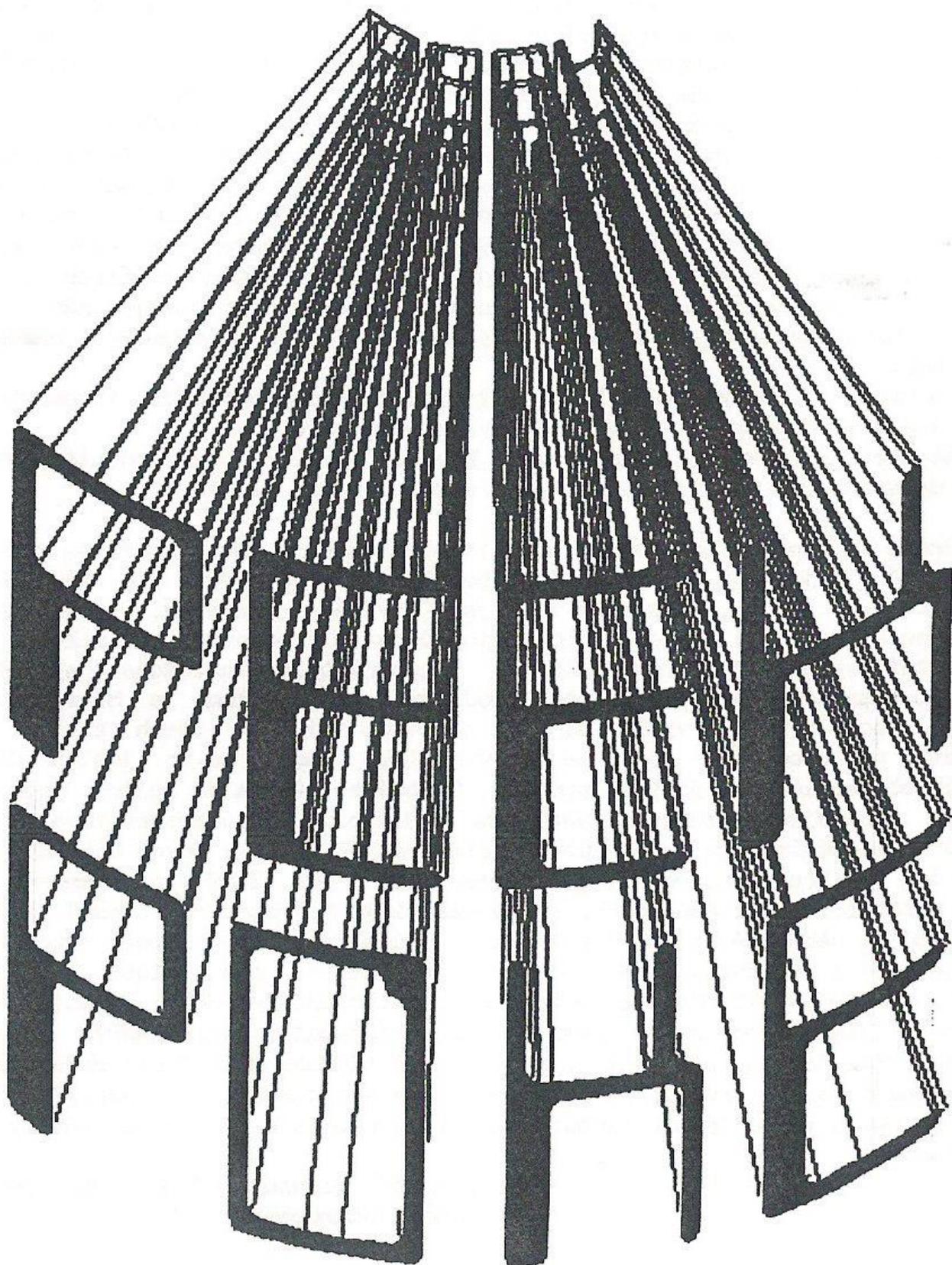


# SHARP 117-800

ZPRAVODAJ SHARP CLUB BRNO

ČÍSLO 3



Vážení přátelé,

toto číslo zpravodaje je zaměřeno na rozdíl od předcházejících poněkud ukecaných čísel více prakticky. Celé je věnováno různým figlům ve strojovém kódu, krátkým podprogramům, úpravám BASICu apod., tj. věcem, počítačovým národem obecně zvaným "píky a pouky". Předem chci upozornit na tři věci.

Za prvé - těm z vás, kterým budou připadat předkládané věci příliš složité, doporučuji k prostudování literaturu, které je o počítačové tematice dnes poměrně dostatek. Zejména vyzdvihnu knihu L. Zajíčka Bity do bytu (MF 1988), která je vynikající učebnicí strojového kódu procesoru Z80, a čtyřdílný popis rezidentního systémového softwaru počítače SHARP MZ-800 od Petra Odehnala a Martina Veverky ze SHARP Club Brno (1987). Dále je k dispozici publikace z dnes již zrušeného školicího střediska 141. ZO Svazarmu (tj. dnešního Mikrocentra) "Základy programování mikropočítačů", ve které jsou stručně popsány základy problematiky. Zájemci ji mohou obdržet v klubovních hodinách zdarma (členové klubu po 1 kusu), resp. za 20,- Kčs. O další literatuře týkající se SHARPa vás bude nejlépe informovat Jára Špaček, který sleduje nabídku klubů a firem z celé republiky. Věřím, že pochopíte, že v omezeném rozsahu zpravodaje není možno vše dopodrobna rozebírat. Přesto je celý zpravodaj psán s maximální snahou o přístupnost.

Za druhé - pro ty z vás, kteří již obdobím začátků prošli, vzpomeňte si na ně. Ale i vy najdete ve zpravodaji svoje informace.

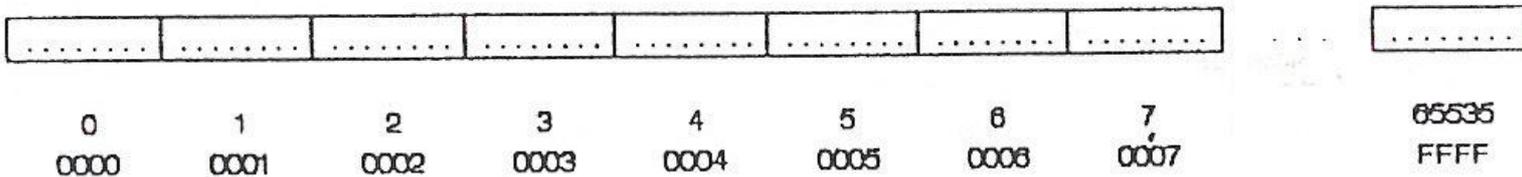
Za třetí - příznivci počítačových her se budou muset poohlédnout po jiných materiálech. Zde žádné "pouky na nesmrtelnost" nenajdou.

Ještě krátce k organizačním záležitostem. Děkuji všem, kteří se podíleli na přípravě a zajištění burzy. Její hospodářský výsledek byl pro klub mírně kladný (cca 200,- Kčs), podstatné však jsou navázané kontakty, spolupráce a prezentace naší (vaší) práce. Akciová společnost Mikrocentrum se nekoná (kdo se zúčastnil výroční konference již ví), místo ní byla založena společnost s ručením omezením MikroComp, jejímž jediným zakladatelem je Mikrocentrum. Tato společnost by měla vydělávat na zájmovou činnost všech klubů. Stav místností na Hybešově - omluvte provizorium, jedná se o jejich dalším využití nebo zajištění jiných prostor. Osobně se stavím za jejich zachování a úpravu vyhovující klubovým potřebám. Možná bude nutné zorganizovat kratší brigádu. V letošním roce bude klubový příspěvek 30,- Kčs a odvod ČSE 20,- Kčs (do 15 let 5,- Kčs), tj. celkem 50,- (resp. 35,-) Kčs. Zvyšování cen se nás zatím nedotklo. Ze zrušeného školicího střediska Mikrocentra získal klub 5 SHARPů, má-li někdo zájem o koupi nebo může sehnat kupce, výbor klubu bude vděčný za informaci. Mikrocentrum zajistilo pro kluby a členy tiskárny CONSUL 2012-04 vč. příslušenství (bohužel v době vydání tohoto zpravodaje již budou s nejvyšší pravděpodobností vyprodány). Stejnou tiskárnu jsme koupili pro klub (je na ní již tištěno toto číslo zpravodaje). Na podmínkách jejího využití se dohodneme na schůzkách. V nejbližší době provedeme novou volbu výboru klubu. Stále přijímám vaše podněty a náměty pro další činnost.

Petr Mynář, předseda klubu a zástupce  
ve výboru Mikrocentra

## PAMĚŤ

Paměť vašeho SHARPa si můžete zjednodušeně představit jako řadu buněk (viz obr.), kterým se říká bajty. Těchto bajtů je za sebou poskládáno v adresním rozsahu procesoru celkem 65536, tj. 64 KB (kilobajt = 1024 bajtů). V každém z bajtů je zapsáno číslo, skládající se z osmi binárních (dvojkových) číslic (tj. jedniček nebo nul), tzv. bitů. Obsah jednoho bajtu tedy může být 00000000 až 11111111 binárně (dvojkově), tj. 00 až FF hexadecimálně (šestnáctkově), což je 0 až 255 decimálně (desítkově). Převody mezi číselnými



soustavami si prostudujte sami např. ve výše uvedených materiálech. Pomoci vám může též tabulka uprostřed tohoto zpravodaje. Přitom mějte na paměti, že čísla jsou v bajtech uložena vždy v binární podobě, hexadecimální tvar slouží pouze k rychlé orientaci (čtou se lépe než dlouhá řada jedniček a nul a přitom jejich převod z a do binárního tvaru je velmi jednoduchý). Decimální soustava potom odpovídá číslům užívaným v běžném životě. Takže je-li v bajtu např. číslo 11001001 binárně, můžeme stejně dobře říci, že je tam číslo C9 hexadecimálně nebo 201 decimálně.

Každý bajt má svoji adresu, což je číslo, které určuje pořadí bajtu v jejich řadě. Adresa prvního bajtu paměti je 0, posledního bajtu 65535 (decimálně). I adresu můžeme zapsat v binární a hexadecimální podobě. Adresy prvního a posledního bajtu jsou binárně 0000000000000000 a 1111111111111111 (všimněte si, že adresa bajtu se skládá ze šestnácti bitů), hexadecimálně 0000 a FFFF. Hovoříme-li tedy o adrese 19919 decimálně, jde o tutéž adresu jako 4DCF hexadecimálně nebo 0100110111001111 binárně.

Programujete-li v BASICu, jste ušetřeni přepočítávání a pamatování si adres, dokonce ani nemusíte vědět, že nějaké existují. Rovněž tak nemusíte vědět nic o bajtech a uložení čísel v nich, pojem bit se nemusíte vůbec zatěžovat. Napíšete např. příkaz PLAT=4550 a vůbec nemusíte vědět, na které adrese (či adresách) je hodnota 4550 uložena (hloubavější si jistě všimli, že čísla se ukládají do více bajtů- jeden bajt je omezen hodnotou 255 decimálně). Stačí vám, že to ví BASIC, který si sám našel volné adresy na uložení hodnoty proměnné PLAT (dokonce pokud už nemá žádné volné adresy, tak vám to oznámí),

přepočítal si ji do příslušného (binárního) tvaru, hodnotu uložil a ještě si zapsal (kam, to vás vůbec nemusí zajímat) adresu spolu s názvem proměnné tak, aby v případě, že po něm budete chtít hodnotu proměnné PLAT, bez potíží ji našel a předal vám ve vám srozumitelném tvaru. Tedy komfort se vším všudy.

V životě programátora občas nastávají situace, kdy s pouhým BASICem nevystačí. V takovém případě obvykle sáhne po strojovém kódu (uvědomme si, že sedíme u SHARPa, na kterém nejsou implementovány další skutečně prakticky využitelné vyšší programovací jazyky, samozřejmě s výjimkou pracujících pod CP/M), zahodí komfort BASICu, avšak získá mnohé další výhody. Program ve strojovém kódu běží jednak podstatně rychleji než v BASICu, jednak umožňuje řídit přímo hardware vašeho počítače. Kromě toho, BASIC je rovněž programem (zapsaným v paměti RAM ve strojovém kódu) a jako takový není prost chyb (při vší úctě k zaměstnancům firmy SHARP). Zásahem do strojového kódu BASICU je možno tyto chyby opravit, případně přizpůsobit vlastnosti BASICu lépe řešené problematice. Takovéto zásahy však vyžadují skutečného znalce (chyba v jediném bitu může znamenat havárii programu), na druhé straně je však známa celá řada vyzkoušených postupů, které si můžete podle popisu realizovat sami a které jsou v převážné míře obsahem tohoto čísla zpravodaje. Samozřejmě můžete objevit mnohé další, zde nepopsané (čekáme na ně), vyžaduje to však jisté minimální znalosti. Metoda pokusů a omylů zde selhává.

## PEEK, POKE a další

BASIC má několik příkazů pro ovládání strojového kódu. Podrobně si o nich můžete přečíst v manuálu na straně 6-84 a dalších. Parametry příkazů můžete používat buď v decimální soustavě, nebo v hexadecimální soustavě (číslo bezprostředně předchází znak "\$"). Takže číslo 19919 můžete zapsat stejně dobře jako \$4DCF. Tento způsob rozlišení číselných soustav platí v BASICu SHARPa obecně, tj. nikoliv pouze u příkazů pro ovládání strojového kódu. Kromě toho je k dispozici funkce pro převod do hexadecimální soustavy HEX\$(x). Např. povel ? HEX\$(15) vytiskne F (? je zkratka pro PRINT). V tomto zpravodaji se budou používat obě číselné soustavy s dodržováním uvedené konvence.

Stručně k příkazům BASICu pro kontrolu strojového kódu:

PEEK (adr) ... přečte a vrátí hodnotu uloženou v bajtu na adrese adr. Využívá se pro přečtení obsahu bajtu z adresy adr.

POKE adr,data ... uloží na adresu adr hodnotu data. Bajt na adrese adr bude mít tedy po provedení příkazu hodnotu data. Je-li potřeba uložit více hodnot za sebou na postupně se zvyšující adresy, stačí zadat počáteční adresu a hodnoty zapsat za sebou takto: POKE adr,data,data,...,data.

INP@ port,proměnná ... přečte bajt ze zadaného portu a uloží jeho hodnotu do zadané proměnné.

OUT@ port,výraz ... pošle hodnotu výrazu na port.

USR adr ... předá řízení strojovému programu na adrese adr. Umožňuje též předávání parametrů mezi BASICem a strojákem (viz. manuál).

LIMIT adr ... vyhradí oblast paměti, do které nemá BASIC přístup. Od adresy adr směrem k vyšším adresám nemůže BASIC zapisovat svoje údaje (text programu, hodnoty proměnných apod.) a nemůže tak poškodit obsah paměti zde uložený. Samozřejmě se příslušně zmenší volná paměť dostupná BASICu. Maximální oblast pro BASIC se vyhradí příkazem LIMIT MAX.

Zde uvedené činnosti se dají samozřejmě provést i ve strojovém kódu nebo příkazy monitoru. Postup může být různý, podstatný je výsledek.

## A MŮŽEME ZAČÍT

Basic používaný u počítače SHARP MZ-800 je velmi vyspělým programovým produktem, který je možno považovat za jeden z vrcholů osmibitové éry domácích počítačů. Citelně mu chybí pouze možnost předávání parametrů při volání podprogramů (přiřazení hodnot proměnným před voláním podprogramu a po návratu z něj je poněkud těžkopádné a koneckonců taky odporující zásadám strukturovaného programování). Volná paměť pro BASICké programy není též nijak velká (ale narazil někdo z vás na tuto hranici?).

Vzhledem k tomu, že BASIC není uložen v počítači rezidentně (musí se "natahovat"), je přístupný úpravám různých softwarových kutilů. V důsledku toho existuje velké množství verzí, lišících se často jen v několika bajtech. Obecně v tomto zpravodaji budeme hovořit pouze o třech základních:

BASIC MZ-1Z016 ... základní BASIC, dodávaný na kazetě spolu s počítačem. Je nejrozšířenější ze všech, obsahuje značné množství chyb. Vyskytuje se ve značném počtu verzí, upravujících některé jeho vlastnosti.

BASIC MZ-5Z009 ... novější verze BASICu MZ-1Z016, vyvinutá firmou SHARP původně pro počítače vybavené quick-diskem. Další úpravy byly provedeny u nás (Pilsoft) a rozšiřuje se pod názvem SHARP Basic 800. Má opravěnu většinu chyb a zná množství nových příkazů (pro zájemce je k dispozici vě. manuál). Doporučuje se k všeobecnému používání.

BASIC MZ-1Z013 ... původně pro počítač SHARP MZ-700, vzhledem ke kompatibilitě směrem nahoru funguje i na počítačích MZ-800. Neobsahuje jemnou grafiku, zato poskytuje podstatně více volné paměti.

Tyto verze se od sebe liší jednak svými vlastnostmi, jednak (a to je pro nás v tuto chvíli podstatné) svým kódem. Při jejich úpravách je k tomu nutno přihlížet. Proto budou úpravy popsány zvlášť pro každou verzi (pokud jsou známy) a to, co platí pro jednu z nich obecně nemusí (ale může) platit pro druhou. Sbíрка je poskládána z různých dostupných materiálů, mnohdy bez ověření funkce. Vše zde uvedené je tedy bez záruky a bez nároku na úplnost. Doporučuji též před zápisem do paměti (POKE) zjistit původní obsah (PEEK).

Volba druhé znakové sady:

POKE \$5D4,1 (PRINT)

POKE \$4DCF,1 (SYMBOL)

Rychlé SAVE/LOAD:

POKE 15249,35

POKE 15255,11

POKE 15259,49

Nahrávání jen jednoho bloku dat:

POKE \$3919,1

POKE \$3951,1

POKE \$399A,1

Zkrácený zaváděcí tón při SAVE:

POKE \$3AFF,\$30,\$11

POKE \$3B0A,\$98,\$8

Doplnění příkazu BOR pro změnu barvy borderu (zruší OFF !):

POKE 23017,66,79,210

POKE 21936,205,218,132,123,1,207,6,237,121,201

POKE 23709,176,85

Podprogram pro změnu barvy borderu:

POKE \$F100,\$3E,xx,11,\$CF,6,\$ED,\$79,\$C9

použití: .USR(\$F100) xx je kód barvy 0 až 15

podprogram může být zapsán i na jiných adresách

Vylepšený TRON - vypisuje se pouze vlevo nahoře:

POKE 22400,42,134,16,229,42,130,16,229,33

POKE 22409,0,0,34,130,16,33,0,128,34,134,16,42

POKE 22421,218,157,205,33

POKE 22425,121,223,11,6,5,62,32,205,18,0,16,249,225,34

POKE 22439,130,16,225,34,134,16,195,6,98

použití: nový TRON : POKE 25078,195,128,87

původní TRON: POKE 25078,62,91,223

Doplnění příkazů SCROLL, BORDER, SYNC:

POKE \$5D1D,\$C5,\$55,\$B0,\$55,\$B9,\$55

POKE \$5B17,\$53,\$59,\$4E,\$C3,\$42,\$4F,\$52,\$44,\$45,\$D2,\$53,\$43,\$52,\$4F,  
\$4C,\$CC

POKE \$55B0,\$CD,\$C6,\$84,1,\$CF,6,\$ED,\$59,\$C9,\$CD,\$DA,\$84,1,\$CF,1,\$ED

POKE \$5500,\$59,4,\$ED,\$51,\$C9,\$DB,\$CE,\$E6,\$40,\$28,\$FA,\$DB,\$CE,\$E6,\$40,  
\$20,\$FA,\$C9

Odstranění chyby v příkazu SYMBOL:

POKE \$55D2,\$CD,\$EA,\$84,\$78,\$B7,\$C0,\$11,\$DE,\$55,6,1,\$C9,\$20

POKE \$78A9,\$CD,\$D2,\$55

Doplnění příkazu SPEED pro CMT:

POKE \$55E0,\$CD,\$C6,\$84,\$7B,\$FE,2,\$D2,\$E,\$4D,\$E5,\$21,1,\$56,\$87,\$83,\$85,  
\$6F,\$7E,\$32,\$91,\$3B,\$23,\$7E,\$32,\$97,\$3B,\$23,\$7E,\$32,\$9B,\$3B,  
\$E1,\$C9

POKE \$5601,76,24,105,35,11,49,22,6,31

POKE \$5AAA,\$53,\$50,\$45,\$45,\$C4

POKE \$5CFD,\$E0,\$55

Použití přidavné VRAM jako RAM-disku:

AW=\$5780:AR=\$57C0

POKE AW,\$E5,\$F5,\$CB,\$FC,\$3A,\$7A,\$13,\$3D,\$20,13,\$37,\$CB,\$1C,\$CB,\$1D,\$CB,  
\$B4,\$3E,\$18,\$CB,\$AC,\$38,2,\$3E,\$14,\$D3,\$CC,\$DB,\$E0,\$F1,\$77,\$DB,  
\$E1,\$B7,\$E1,\$C9

POKE AR,\$E5,\$F5,\$CB,\$FC,\$3A,\$7A,\$13,\$3D,\$20,13,\$37,\$CB,\$1C,\$CB,\$1D,\$CB,  
\$B4,\$3E,\$18,\$CB,\$AC,\$38,2,\$3E,\$14,\$D3,\$CD,\$DB,\$E0,\$F1,\$6E,\$DB,  
\$E1,\$7D,\$B7,\$E1,\$C9

POKE \$4094,\$C3,AW MOD 256,AW/256

POKE \$409F,\$C3,AR MOD 256,AR/256

POKE \$F6B,0:POKE \$1099,0:POKE \$13C3,17

Pruhy při LOAD:

BD=\$5740:BA=BD+21:BO=BA+1

POKE BD,\$E6,\$20,\$C2,\$3C,\$3A,\$C5,1,BA AND 255,BA/256,\$A,\$2F,2,1,\$CF,6,  
\$ED,\$79,\$C1,\$C3,\$4A,\$3A

POKE \$3A46,\$C3,BD AND 255,BD/256

POKE BO,1,\$CF,6,\$AF,\$ED,\$79,\$C1,\$F1,\$C9

POKE \$3B79,\$C3,BO AND 255,BO/256

POKE BA,3

Klávesa DEL maže znak pod kurzorem:

POKE \$6C4,AD MOD 256,AD/256

POKE AD,\$CD,\$D4,9,\$C3,\$CC,6

Původní význam klávesy DEL klávese BLANK:

POKE \$D49,\$1E

POKE \$97,\$BE,6

CAPS LOCK místo SHIFT LOCK:

POKE \$CC4,\$C3,AD MOD 256,AD/256,0

POKE AD,\$FE,1,\$C2,\$CC,\$C,\$85,\$FE,8,\$DA,\$CC,\$C,\$FE,\$41,\$D2,\$CC,\$C,\$C3,  
\$C8,\$C

CTRL Z smaže zbytek řádku:

POKE AD,\$C5,6,80,\$C5,\$CD,\$BE,6,\$C1,\$10,\$F9,\$C1,\$C9

POKE \$8F,AD MOD 256,AD/256

Zrušení tiskárny:

POKE \$9F6E,\$C9:USR(\$9F58)

CONSOLE nemaže obrazovku:

POKE 1257,0,0,0

PRINT nemaže podklad:

POKE \$05DA,\$00

Zrušení příkazu NEW:

POKE \$9F58,\$C9

LOAD adresa bez změny při NEW ON:

POKE \$9F6E,\$C9

Zrušení příkazu SAVE:

POKE 29503,201

Zrušení příkazu HIST:

POKE 27293,201

Zrušení příkazu PEEK:

POKE 39011,201



Zrušení příkazu BYE:  
 POKE 27228,201

Zákaz zastavení ESC:  
 POKE 3368,201

Zákaz přerušení SHIFT BREAK:  
 POKE 22754,62

Úpravy příkazu INPUT:  
 POKE 28213,0,0  
 POKE 28213,63,32 ... zruší otazník  
 POKE 28213,32,58 ... vypisuje dvojtečku

Pípnutí při READY:  
 POKE 21953,205,62,0,17,240,99,201  
 POKE 22650,205,193,85

Tvar kurzoru (je dán bitovou mapou na uvedených adresách):  
 POKE \$1391,\$7E,\$BD,\$DB,\$E7,\$E7,\$DB,\$BD,\$7E (normální ALPHA kurzor)  
 POKE \$1399,\$E7,\$E7,\$E7,0,0,\$E7,\$E7,\$E7 (SHIFT ALPHA kurzor)  
 POKE \$13A1,0,0,0,0,0,0,0,\$FF (GRAPH kurzor)

Výpisy při SEARCH a LIST s vloženým prázdným řádkem:  
 POKE 21946,58,131,107,205,6,0,201  
 POKE 27424,205,146,85  
 POKE 21909,205 ... zapnutí prázdného řádku  
 POKE 21909,201 ... vypnutí prázdného řádku

Test instalace přídavné VRAM:  
 PEEK(\$1099) 0 ... není instalována  
 1 ... je instalována

Zjištění aktuálního COLOR:  
 PEEK(\$109B)

Zjištění nastaveného počtu znaků na řádku:  
 PEEK(\$136B)

Čtení stisklé klávesy:  
 USR(\$BA1) ... ukládá kód stisklé klávesy na adresu \$1365

Test stisknutí klávesy SHIFT:  
 INP@(\$D3),SH SH=255 ... SHIFT není stisknut  
 SH=254 ... SHIFT je stisknut

Příkaz GET čeká na stisk klávesy:  
 POKE \$6879,1 ... při GET <řetězec> čeká na stisk klávesy a bliká kurzorem

Startovací adresa po zhroucení BASICu (z monitoru):  
 G005B (nebo CTRL+RESET)

Uložení programu v BASICu:  
 od \$A3FA (po NEW ON od \$9F9E)  
 adr ... délka řádku +5  
 adr+2,adr+3 ... číslo řádku (\*1,\*256)  
 adr+4 až adr+PEEK(adr) ... obsah řádku (příkazy jsou uloženy ve zkrácené formě ve tvaru tzv. tokenů)

Uložení DEFKEY:  
 od \$12B2 do \$1352

CONSOLE nemaže obrazovku:

POKE \$04E9,\$21

Volba druhé znakové sady:

POKE \$5D4,1 (PRINT)

POKE \$4DD0,1 (SYMBOL)

PATTERN nemaže pozadí:

POKE \$5031,0

CTRL+RESET maže paměť:

POKE \$004D,0 (lépe přidat POKE \$5808,\$2A)

Zákaz BREAK, ESC zůstává:

POKE \$0D2B,0

Zákaz BREAK i ESC (kromě LIST):

POKE \$0D27,\$3E

Zákaz BREAK při INPUT:

POKE \$0A90,0

BREAK při INPUT jako chyba:

POKE \$00E9,kód chyby (nutno zakázat BREAK při běhu, lze  
použít ON ERROR GOTO)

Tvar kurzoru:

Je uložen na stejných adresách jako u BASICu MZ-1Z016

Tisk PLAY, RECORD, Loading ... při běhu programu:

POKE \$108E,0 (platí jen do konce řádky)

POKE \$5910,\$3A (předcházející úprava platí stále)

Předdefinování znaků do vstupu:

POKE \$1352,0,n,z1,z2,...,zn (n<=15) způsobí, že se znaky z1 až zn  
při dalším testu klávesnice  
jakoby stisknou.

př.: POKE \$1352,0,4,\$43,\$4C,\$53,\$0D  
C L S <CR>

Volné místo pro krátké strojáky:

\$5720 až \$57FF

Kód klávesy, stisknuté při posledním testu klávesnice:

PEEK(\$1365)

Test MUSIC PLAY:

PEEK(\$40FA) je-li >0, hraje hudba

Tón při stisknutí klávesy (písmena, čísla):

POKE 399,150

Přerušení programu pouze BREAK (bez SHIFT):

POKE 1203,201

Tón při stisknutí CR:

POKE 292,1

Blokování PSG:

POKE 57347,0

Tisk PLAY, RECORD, LOADING ... při ROPEN/WOPEN:

POKE 18242,205:POKE 18252,175

LIST s vloženými prázdnými řádky:

POKE 12410,205,249,23,205,249,23,201:POKE 16781,122,48

Změna tvaru kurzoru na znak daný kódem:

POKE 96,kód znaku

Přerušení dokud není stisknuta jakákoli klávesa:

USR(\$25B)

Pípnutí:

USR(62)

Nastavení výšky tónu pro předcházející USR:

POKE 2618,0 až 255

Souvislý tón o výšce dané předcházejícím POKE:

USR(68)

Zastavení předcházejícího USR:

USR(71)

Využití druhé znakové sady:

POKE \$5D,PEEK(\$5D)+128 (má vliv pouze na bezprostředně následující PRINT, další tisk je již z první sady)

Máš-li co lepšího, ven s tím, jinak se podděj!

Quintus Flaccus Horatius (65 - 8 př.n.l.)

Tato sbírka samozřejmě ani zdaleka není úplná. Má-li kdokoliv jakékoliv připomínky, změny a další úpravy (a má potřebu je sdílet s dalšími uživateli), jsou mu tyto stránky k dispozici.

## MÁTE RÁDI EXPERIMENTY ?

Tento se bude týkat časovače (I8253), který v počítači generuje akustický kmitočet a především realizuje hodiny reálného času. Nejdříve stručný popis I8253, ale ne tak stručný jako v manuálu na str. 7-5.

Obvod má tři na sobě nezávislé 16-bitové čítače s frekvencí čítání od 0 do 2 MHz. Čítání může být binární i dekadické. Obvod má dále jeden řídicí registr, který je určen pro naprogramování režimu, zavedení hodnoty předvolby a čtení stavu čítačů. Čítače 0 až 2 jsou funkčně identické a tak stačí popsat pouze jeden z nich. Každý čítač má jeden vstup (CLK), jeden výstup (OUT) a jeden vstup GATE, který může zastavit nebo povolit čítání v závislosti na zvoleném režimu. Obsah čítače je možno číst i při chodu, aniž by bylo čítání narušeno. Čítání probíhá vždy směrem dolů.

## Režimy čítání:

- MOD 0 - přerušení na konci čítání. Výstup čítače je po operaci volby na úrovni L. Po zapsání předvolby zahájí čítání a po dočítání na 0 přejde výstup na úroveň H a zůstane na ní, dokud není čítač znovu předvolen.
- MOD 1 - programovatelný monostabilní generátor. Výstup přejde na úroveň L po zahájení čítání vzestupnou hranou na vstupu GATE. Po dočítání přejde výstup na úroveň H.
- MOD 2 - dělení frekvence číslem N. Výstup bude na úrovni L po dobu jedné periody na vstupu. Rozteč mezi výstupními impulsy se rovná počtu N period na vstupu.
- MOD 3 - generování obdélníkových impulsů. Výstup bude na úrovních L i H po dobu rovnající se N/2 period na vstupu. Je-li N liché číslo, bude se úroveň L na výstupu rovnat (N-1)/2 a úroveň H (N+1)/2.
- MOD 4 - programem ovládané strobování. Po zavedení režimu bude výstup na úrovni H. Po zavedení předvolby začne čítání. Po dočítání bude na výstupu úroveň L po dobu jedné periody na vstupu a potom se vrací na úroveň H.
- MOD 5 - obvodově ovládané strobování. Čítání se zahájí po vzestupné hraně na vstupu GATE a po dočítání bude na výstupu úroveň L po dobu jedné periody na vstupu.

## Nastavení režimu, předvolby a čtení čítačů:

Adresy a porty pro I8253 viz manuál str. 7-5.

Řídicí slovo:        bit    7   - 6   - 5   - 4   - 3   - 2   - 1   - 0  
                   symbol SC1 - SC0 - RL1 - RL0 - M2   - M1   - M0   - BCD

Volba čítače    : SC1 - SC0

                  0    0   - čítač č. 0

                  0    1   - čítač č. 1

                  1    0   - čítač č. 2

Čtení/předvolba: RL1 - RL0

                  0    1   - pouze nižší slabika

                  1    0   - pouze vyšší slabika

                  1    1   - nejdřív nižší a pak vyšší slabika

Nastavení modu : M2   - M1   - M0

                  0    0    0   - mod 0

                  0    0    1   - mod 1

                  -    1    0   - mod 2

                  -    1    1   - mod 3

                  1    0    0   - mod 4

                  1    0    1   - mod 5

Režim čítání    : BCD

                  0   - binární čítání (16-bitový čítač)

                  1   - dekadické čítání (4 dekády)

Čtení za chodu : SC1 a SC0 = volba čítače

                  RL1 a RL0 = 0 , 0

                  M2, M1, M0 a BCD jsou bezvýznamné

V počítači má I8253 jeden čítač určen ke generování akustického kmitočtu. Čítač pracuje v modu 3 a na jeho vstup je přiveden kmitočet asi 1.1 MHz. Výstup je připojen na hradlo NAND (druhý vstup hradla je připojen na výstup PO0 I8255) a dále přes zesilovač na reproduktor. Chceme-li zvuk spustit, musíme výstup PO0 nastavit na hodnotu 1. V režimu MZ-700 můžeme zavedením hodnoty 1 nebo 0 na adr. E008h změnit úroveň na GATE 0 a tím zastavit nebo povolit čítání.

Na vstup CLK čítače 1 je přiveden řádkový kmitočet 15611 Hz. Čítač pracuje v modu 2 a předvolba je nastavena na hodnotu 15611. Na výstupu je potom kmitočet 1 Hz a ten je zároveň připojen na vstup čítače 2. Čítač 2 pracuje v modu 0 a jeho předvolba je po resetu počítače nastavena na hodnotu 43200, což je počet sekund za 12 hodin. Čtením hodnoty čítače dostaneme hodnotu předvolby zmenšenou o počet sekund, které uběhly od poslední předvolby. Po dočítání na 0 se na výstupu čítače objeví úroveň H, která si vyžádá přerušeni. Pokud není přerušeni zakázáno, provede se restart na adr. 38h. Obslužný program změni příznak dopoledne/odpoledne (adr. 119Bh) a do čítače bude opět zapsána předvolba 43200.

A experimenty ? Ty už nechám na vás. Ale na zkoušku si můžete vyzkoušet tyto dva:

Program demonstruje zvuk v rozmezí 20 až 15000 Hz.

```

100 INIT"CRT:M1":F=20          `F=kmitocet
110 OUT@$D7,$36                `citac 0 - mod 3
120 CURSOR 10,10:PRINT"frekvence =      Hz"
130 A=1108404                  `kmitocet na vstupu
140 H=INT((A/F)/256)
150 L=INT((A/F)-(H*256))
160 OUT@$D4,L:OUT@$D4,H        `nastaveni predvolby
170 OUT@$D3,1                  `PO0 na 1 (zapni zvuk)
180 CURSOR 22,10:PRINTUSING"#####";F
190 F=F+10:IF F<15001 THEN 140
200 OUT@$D3,0:END              `PO0 na 0 (vypni zvuk)

```

Program počítá setiny sekundy (čas není zcela přesný, číslo 15611 se totiž setsakra blbě dělí, aniž by něco zbylo).

```

100 INIT"CRT:M1"
110 CURSOR 10,10:PRINT"cas :          sec"
120 OUT@$D7,$74                `citac 1 - mod 2
130 OUT@$D7,$B0                `citac 2 - mod 0
140 OUT@$D7,$80                `cteni citace 2 za chodu
150 OUT@$D5,156:OUT@$D5,0      `citac 1 - N=156
160 OUT@$D6,$60:OUT@$D6,$EA    `citac 2 - N=60000
170 INP@$D6,A :INP@$D6,B      `cte N z citace 2
180 C=(60000-A-B*256)/100
190 CURSOR 16,10:PRINTUSING"###.##";C
200 IF C<599 THEN 170 ELSE END

```

Lubomír Srnský

## MÁTE ZÁJEM ?

V průběhu času vykrytalizovalo v naší republice několik firem, zabývajících se vývojem a distribucí software pro SHARP MZ-800. Pro vaši informaci uvádíme adresy dodavatelů, sdružených do Asociace distributorů software SHARP MZ-800:

BBS corporation  
Milan Bendl  
Kosá 274/8  
405 02 DĚČÍN VII

hry, převážně upravené  
Petrem Pátíkem ze Sinclaira

RTMVsoft  
Rudolf Jaroměřský  
Jungmannova 1438  
500 05 HRADEC KRÁLOVÉ II

hry, převážně z dílny Profi

mZx Soft - O.Zemek jr.  
P.O. box 93  
381 01 ČESKÝ KRUMLOV

převážně užitkové programy a původní tvorba,  
v autorském zázemí mj. PROTON Ústí nad Labem

Computer Service JOPSON  
P.O. box 12  
756 05 KAROLINKA

převážně užitkové programy, literatura

Firmy zašlou na vyžádání nabídkové katalogy tištěné, případně ve FETu.

## PŘIPOJENÍ TISKÁRNY CONSUL 2012

Tiskárna CONSUL 2012, která se objevila na výprodejovém trhu, poskytuje za svoji cenu (do 4000,- Kčs) kvalitu vysoce nadprůměrnou. Umožňuje tisknout na jednotlivé listy i na perforovaný papír v několika kvalitách tisku, běžných u výkonnějších (a podstatně dražších) zahraničních tiskáren, tj. draft (běžná datová kvalita) a NLQ (typ písma Roman a SansSerif) s běžnými roztečemi znaků (PICA, ELITE, PROPORTIONAL). Umožňuje tisk písmem zúženým, rozšířeným, tučným, dvojitým, kurzívou, horními a dolními indexy, vše s možností automatického podtrhávání. Ovládání je kompatibilní se standardem ESC/P. Navíc obsahuje několik kódových tabulek (KOI-8, EPSON, LATIN 2, Kameničtí) a možnost download. Připojení této tiskárny k SHARP MZ-800 je zcela shodné s připojením tiskárny SEP 510, které bylo popsáno v minulém čísle zpravodaje. Vzhledem ke stejnému standardu ovládání fungují i tamtéž uvedené programy pro HARDCOPY.

Jedna rada nakonec:

Originální ploché konektory pro výstup CENTRONICS SHARPa jsou v prodeji za sníženou cenu (74,- Kčs) v prodejně ELTOS na Radnické. Původně sloužily k připojení tiskárny GAMACENTRUM k SHARPu a stály 350,- Kčs.

---

Zpravodaj vydává SHARP MZ Club, Mikrocentrum, Hlinky 164, 603 00 Brno, tel. 331 666. Číslo sestavil ing. Petr Mynář. Náklad 80 výtisků. Brno, únor 1992