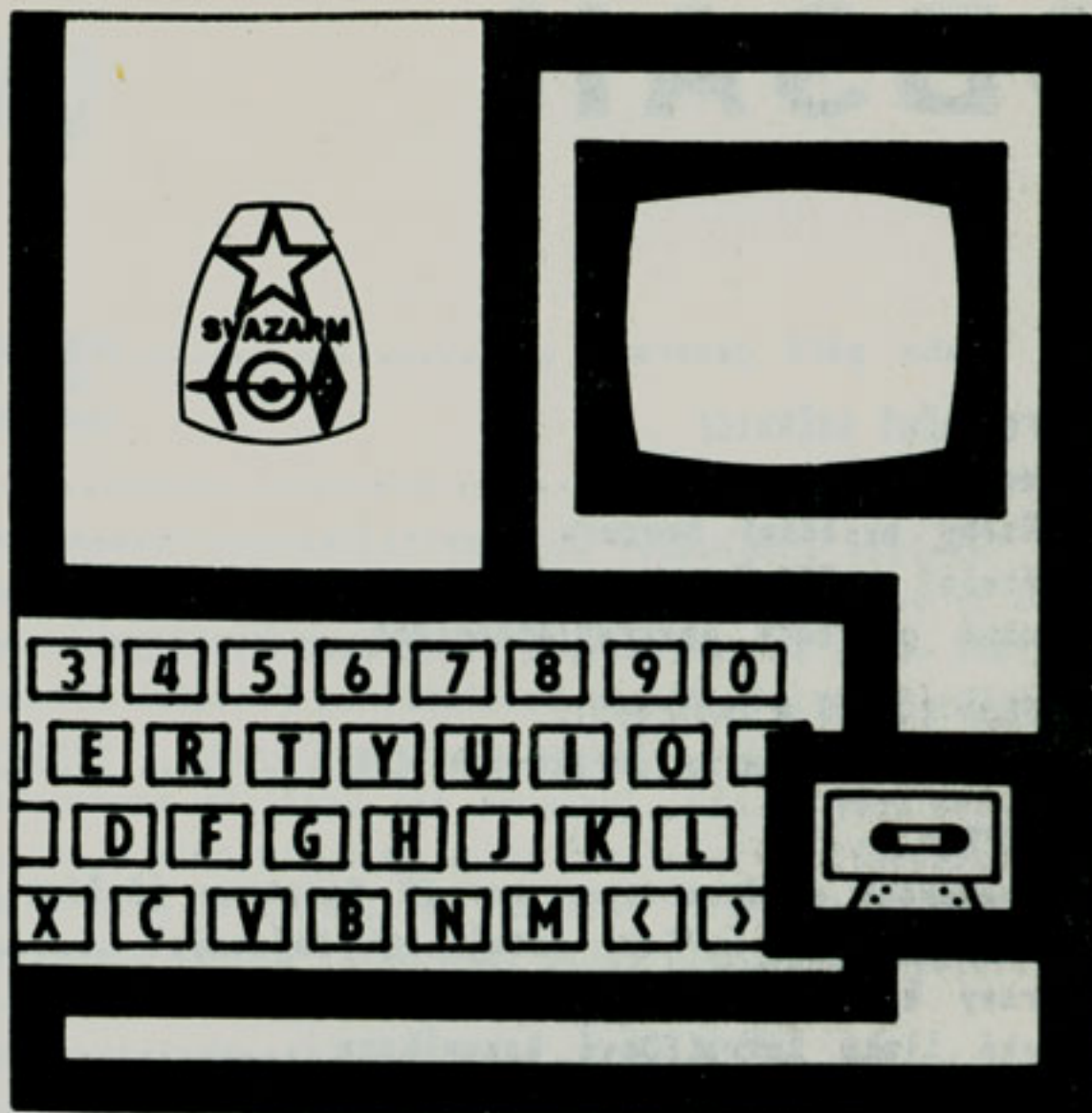


**Členský  
zpravodaj**

**3**



# MIKROBÁZE



**Společná služba  
Amatérského radia  
a 602. ZO Svazarmu  
pro uživatele mikropočítačů**

# OBSAH

Na prahu páté generace .....	3
● <b>INFORMAČNÍ SBĚRNICE</b>	
Spectrum 128 .....	5
Způsoby ovládnání kurzoru .....	9
Světelný počítač .....	12
Osobní počítače Amstrad/Schneider .....	14
● <b>HERBÁŘ NÁPADŮ A ZKUŠENOSTÍ</b>	
Kompresce a expanze obrazových dat .....	20
● <b>SPECTRUM</b>	
Integrovaný wordprocessor pro ZX Spectrum (3.část) .....	27
Kterak program opřísti zatajenými tajnými .....	34
Úpravy her pro ZX Spectrum .....	39
Horká linka infraLEDové komunikace .....	44
Mikroprocesor Z80 .....	47
● <b>PROGRAMOVÁ NABÍDKA MIKROBÁZE</b> .....	53
ZX Spectrum .....	56
Karel .....	57
Index / Nová služba členům Svazarmu .....	58
Pokyny k objednávání programů .....	60
Slovo k náhodným čtenářům .....	62



# NA PRAHU PÁTÉ GENERACE

Snad žádný o oboru lidské činnosti neprodělal takovou "populační explozi" jako elektronika v posledních desetiletích. Vynechme čísla statistik a vezměme si běžný příklad ze života. Zatímco generace, která dnes slaví padesátiny, ve svých školních letech (a to mám na mysli školství odborné) pokládala za vrchol miniaturizace "mejdlonetu" (pro ty později narozené - jednolampovku v krabičce od mýdla), generace jejich synů zná klasický elektronkový přijímač jen okrajově. Pro ni jsou běžné mikroprocesory, optocuplery, ví, co je strojový kód, volný čas tráví u klávesnic počítače a na superhit svého otce se dívá se shovívavým úsměvem - transistor, hit padesátých let, se dnes, v době vysokého stupně integrace, stává historickou záležitostí.

Nebývalý rozvoj elektroniky zcela zákonitě ovlivnil všechny obory lidské činnosti, řadu z nich velmi výrazně. Počítačové systémy ovládly ekonomiku a řízení, evidenci a dokumentaci, vývoj, konstrukci i komunikace, stále více se prosazují ve zdravotnictví a školství. Počítač se stal zaklínadlem, příslibem, řešením i nadějí.

V roce 1986 stojíme na prahu páté generace výpočetních systémů, které by měly rozhodujícím způsobem ovlivnit sociální a ekonomickou činnost společnosti. V roce 1983 uváděla statistika počet počítačů nasazených v čs. národním hospodářství číslem 4000 (šlo o velké a střední celky, nikoli mikropočítače). Vytvářela se koncepce řady JSEP (Jednotný systém elektronických počítačů) a MSEP (malých elektronických počítačů). Začínaly vznikat první počítačové sítě a informační systémy, které by z koncepčních hledisek měly nabýt své finální podoby v průběhu 7. pětiletky, rozvíjely se první kooperační kontakty v rámci spolupráce zemí RVHP. Úlohu informačních systémů zdůraznily i závěry 8. zasedání ÚV KSČ, věnovaného problematice urychlení uplatnění vědy a techniky v národním hospodářství. Dnes vystupuje do popředí problematika využití výpočetní techniky při soustředění, třídění, zpracování a zpřístupnění informací.

Počítač se stal zaklínadlem, příslibem, řešením i nadějí. Dnes již jsou za námi doby, kdy počítač zpracovával mzdovou agendu a nad jeho "využitím" vládla všeobecná spokojenost. Počítače začínají zaujímat své místo v řídicí a koordinační sféře, ve spojení s periferními zařízeními vystupuje stále naléhavěji otázka zavádění a efektivního využívání databank včetně spojovacích systémů. Tím se vyostřuje potřeba výchovy kvalifikovaných pracovníků schopných s výpočetní technikou tvůrčím způsobem pracovat. Je zřejmé, že jakékoli zařízení na bázi výpočetní techniky se stane řešením jen tehdy, bude-li s ním pracovat kádr techniků a inženýrů schopných jeho dispozice v plné míře využít a aplikovat.

Tento trend se zrcadlí i v našem vzdělávacím a výchovném systému. Vedle již zavedené vysokoškolské výuky programování přistupujeme letošním rokem k výuce informati-



ky na středních školách, nemalými náklady budujeme výuková terminálová pracoviště, pokračuje vývoj malých počítačů určených nejmladší generaci.

Připadá nám vlastně samozřejmé, že stranou tohoto výchovného procesu nezůstal ani Svazarm. Organizace, která letos slaví 35.výročí své úspěšné a nezastupitelné činnosti, se aktivně zapojuje i do řešení této společensky naléhavé problematiky. 602.ZO Svazarmu jako první u nás zachytila tento vývojový trend a stala se osvětovou základnou zájemců o výpočetní techniku, základnou, z níž mohou vyrůstat odborníci rozhodující o tom, zda počítač přestane být nadějí a stane se řešením.

Nosným prvkem projektu Mikrobáze, realizovaného společně s redakcí Amatérského radia, nebude jen tento Zpravodaj. Spolu s dalšími službami bude postupně přinášet programovou tvorbu pro vybrané typy mikropočítačů a současně napomáhat vzniku původních česky a slovensky komunikujících programů na profesionální úrovni, zprostředkovávat výměnu softwaru a zkušeností.

Tvrdím při nejrůznějších příležitostech, že "náš amatér dokáže zázraky". Proč o tom píší? Protože právě ten amatér zde sehraje velmi významnou roli. Těžko někoho při návštěvě specializované prodejny potěší nabídka, kterou v oblasti součástkové základny poskytuje. Neprozradím žádné tajemství, když řeknu, že jsme vývoj elektroniky poněkud zaspali. Náběh výroby našich mikropočítačů málokoho potěšil - jejich cena tím méně - a kdyby mi někdo řekl, že třeba v "Dlouhé" prodávají modemy, myslel bych si, že mluví ze spaní. Přesto věřím, že amatér tyto obtížné podmínky překoná a sehraje svou nezanedbatelnou úlohu v rozvoji naší výpočetní techniky i při jejím zavádění do praxe. A i když v té současnosti je hodně míst, "kde jsou lvi", určitě jich bude postupně ubývat. Svým způsobem tomu nepochybně napomůže i Mikrobáze spolu s řadou dalších agilních základních organizací Svazarmu.

kš





## SPECTRUM 128

### Základní údaje

Procesor	Z80A, 3,5 MHz.
Paměť	128 K RAM, 16 stran, 32 K ROM, z toho 16 K původní ROM Spectra pro mód 48 K, druhých 16 K pro práci v módu 128 K.
Video	totéž jako u Spectra 48 K.
Klávesnice	extended mode jako u Spectra 48 K, v módu 128 K, možnost editace i bez klíčových slov, oddělená "krabička" s číselnou klávesnicí a některými editačními funkcemi (128 K).
Zvuk	čip 8912, tříhlasý, čistý zvuk/bílý šum, ovládaný z Basicu příkazem PLAY; Spectrum nemá reproduktor, zvuk vychází z nf výstupu TV přijímače.
Vestavěný software	ZX BASIC, textový editor, kalkulátor, basicový obrazový editor.
ZX BASIC	v módu 48 K stejný jako u Spectra 48 K, v módu 128 K se všechny příkazy vypisují; nové funkce - PLAY (zvuk), LOAD!, SAVE!, CAT! (pro práci s RAM diskem).
I/O	porty RS232, MIDI, RGB, video, MIC, EAR, expanzní konektor stejný jako u Spectra 48 K.
Cena	50000 španělských peset (cca 230 Lstg.).



Informace o novém "Spectru" velmi těžko získávají i sami Angličané. Dále uvedené údaje pocházejí ze zjištění redaktora časopisu Your Spectrum, který musel osobně zajet do Španělska, aby se k nové verzi dostal; jinde se zatím nevyrábí. Španělé ji budou vyvážet zatím jen do Mexika. Protože se tato verze nijak nejmenuje, pojmenujme si ji S128, ať víme, co máme na mysli.

Redaktor podává v podstatě jen velmi stručnou zprávu, během níž někdy musí vyslovovat domněnky, protože producent S128 (španělská Investronica) nebyl příliš sdělný. Původní projekt verze vypracoval Sinclairův Metalab. Prý už je hodně počítačů S128 v anglických softwarových firmách, které "tajně" připravují nové a upravují staré programy pro tuto novinku. Ze všeho vyplývá, že Sinclair nevychází s S128 na trh proto, aby zatím doprodal obrovské zásoby Specter 48 K a QL, včetně některých jejich periférií. Aby byl doprodej úspěšný, padlo rozhodnutí o výrazném snížení cen. A tak koncem roku 85 klesla cena QL na pouhých 199 Lstg. Clive Sinclair se tak snaží vymotat z vysoké zadluženosti, do níž zabředl díky své, zvláště poslední dobou velmi špatně vedené a poněkud zaslepené obchodní politice.

Jistě nejeden z vás, kteří vlastníte Spectrum, se časem ocitnete na vážkách, zda si S128 opatřit či ne. Snad vám trochu pomohou dále uvedené postřehy.

Vzhled S128 je prakticky shodný se Spectrem plus. Po pravé straně je mohutný chladič, který silně hřeje - a to tak, že je bolestivý na dotyk. Součástí S128 je oddělená "kalkulačková" klávesnice, která je točenou šňůrou připojena k počítači konektorem zasunutým do počítače pod pravou rukou zepředu - pohled na obrázek naznačuje, že vám tak šňůra může velmi šikovně překážet při práci.

Konektory MIC a EAR jsou na levé boční straně. Vedle nich je kombinovaný konektor portů RS232 a MIDI. Na zadní straně je výstup RGB pro barevný monitor. Investronica dodává k S128 monitory BMC pro QL. Obraz je prý vynikající. Z téhož výstupu by měl vycházet i barevný video signál pro připojení na video vstup běžného televizoru.

S128 je, dle hodnocení autora článku, schizofrenní, tedy rozdvojené - po připojení na zdroj a inicializaci se ocitá v módu 128 K. Příkazem SPECTRUM jej přepnete do módu 48 K. Bohužel, všechny vymoženosti módu 128 K po přepnutí do módu 48 K mizí. Žádnou z nich nelze využít ani pomocí nějaké strojně-kódové finesy. Při přepnutí do 128 K se celý systém vždy inicializuje (reset).

Veškerý spektrovský software by měl být v módu 48 K plně kompatibilní. Autor podtrhuje slovo *m ě l*, protože na počítači, který testoval, *n e b y l* kompatibilní. Investronica to vysvětlovala tím, že v něm byla zabudována experimentální "epromka", která zápasila s časováním turbo loadu. Vše je již vyřešeno a kompatibilita běžně prodávaných přístrojů je prý plně garantována.

Konektor sběrnice je zapojen stejně jako u Spectra a v módu 48 K lze na něj připojit cokoli bez jakýchkoli změn. Vše by mohlo pravděpodobně fungovat i v módu 128 K. Problémy mohou nastat při připojení periférií, které svým softwarem obsazují část paměti RAM počítače. Ve spojení s S128 budou některé z nich zřejmě disfunkční.

Jednoduché basicové programy Spectra pracují v obou módech stejně dobře. Ale jakmile je v nich užito některých z mnoha jemných spektrovských fines, mód 128 K je "mimo mísu". Tento problém se týká 99 procent komerčně vyráběného softwaru. Příčina tkví především v tom,



že v módu 128 K pracuje nová ROM. A tak bude nutno kupovat software upravený.

## EDITACE

S128 používá full screen editor. To znamená, že při vytváření programu nepíšeme a needitujeme basicové řádky "dole" v editační zóně, ale program vytváříme kdekoli na obrazovce, prostě tam, kde máme umístěn kurzor - podobně jako při zápisu textu v Taswordu. Jemu velmi podobné jsou i některé další editační funkce. Tak můžeme mazat písmena i slova v obou směrech, i celé řádky, projíždět jimi přímo nahoru i dolů, rozdělovat je atd. Syntaxové chyby v editovaném programu vysvitnou ihned po stisknutí tlačítka ENTER, kterým se v editačním programu spouští debugger.

Obrazový editor můžeme užít i jako velmi jednoduchý textový editor. Je zařazena např. funkce word wrap, která dbá na to, aby slova na konci řádek nebyla dělena, dále můžete užít insert, overtype a nastavení levého okraje textu. Autor testu přemítá nad užitkem textového editoru S128 a dochází k závěru, že zařazen být nemusel - hlavně proto, že je příliš chudičký (do slovního procesoru má až příliš daleko). Snad najde využití při sestavování zdrojových textů jiných jazyků nebo datových souborů.

Určitou výhodou při editaci basicových programů je, že můžete využít vy-možeností editace v módu 128 K (a pokud si nebudete vymýšlet přílišné programové triky, bude vám program fungovat i při zpětném přepnutí na mód 48 K; to však už beze ztráty obsahu RAM nejde!).

Milé jazyky tvrdí, že S128 bude mít zařazenu funkci RENUM pro přečíslování řádek basicu...a to prý ještě není

vše...inu, snad už brzy uvidíme.

## ZVUK

Ke stereotypnímu bzučení Spectra má velmi daleko díky zapojení obvodu 8912. Ten umožňuje zvučet až třem tónům současně, tedy v akordu. Ovlivňovat můžeme tvorbu čistého zvuku, bílého šumu a obálky. To platí pro oba módy. Tak se S128 možnostmi tvorby zvuků dostává na úroveň mikropočítačů BBC nebo C64. Bohužel, basic S128 nedovoluje plné využití dispozic zvukového čipu. Basicový příkaz PLAY má za sebou jeden nebo více řetězců se symboly a čísla, které formují výsledný zvuk. PLAY může být užit i pro funkci MIDI interfejsu. Zde ovšem neplatí omezení na tři tóny, ale výstup je "paralelně osmihlasý". Tak můžeme ovládat jakýkoli syntezátor a jinou hudební elektroniku, která má vyvedeny náležité I/O porty pro spojení s MIDI. Teoreticky by měly na S128 fungovat všechny jednoduché programy psané pro čip 8912. Jenže ten můj S128 jakoukoli spolupráci v tomto směru odmítal. Ovšem na druhou stranu programy napsané přímo pro něj dávají efekt vynikající. A abych nezapomněl - nečekejte, že zvuk půjde zevnitř počítače. Jeho sílu a zabarvení ovládáte přímo na zvukových potenciometrech svého televizoru (resp. zesilovače).

## ROM/RAM

Celých 128 K je rozděleno na stránky po 16 K - jakékoli čtyři z nich se mohou objevit na jakémkoli místě z mikroprocesorem Z80A adresovatelných 64 K.

32 K ROM je rozdělena na dvě části po 16 K. První část je "stará" romka Spectra pro mód 48 K. Druhá slouží v módu 128 K. Investronica mi prozradila, že při přepnutí do módu 48 K je



ROM překopírována do jedné stránky paměti RAM, která je pak zajištěna proti přepsání a slouží jako pseudoROM. Lze si snadno domyslet, že by nebylo až takovým problémem se do této části paměti dostat a udělat si v ní libovolné softwarové změny, ba dokonce ji vůbec využít volnou! Ale (ale, ale!) - obrazová paměť zase sedí na adrese 16384!! A tak se můžeme s CP/M zase rozloučit...

Při módu 48 K jsou nad prvních "romkových" 16 K přiřazeny další 3 stránky po 16 K. Jejich přepínání s jinými je znemožněno, a zdá se, že nezvratně.

V módu 128 K se děje totéž, až na to, že jednotlivé stránky můžete vyměňovat (přepínat) dle libosti. Tak máte k dispozici 104 K pro basicový program, resp. asi 120 K pro program ve strojovém kódu (nezapomeňte, že systémové proměnné a obrazová paměť zabírají kolem 8 kB RAM). To ovšem platí tehdy, pokud si i v módu 128 K budeme moci "přivlastnit" spodních 16 K pseudoROM.

Z BASICu se manipulace s RAM diskem provádějí příkazy LOAD!, SAVE! a CAT!. Můžete ovšem užít i IN, OUT, PEEK a POKE, nebo vše řídit přímo strojovým kódem.

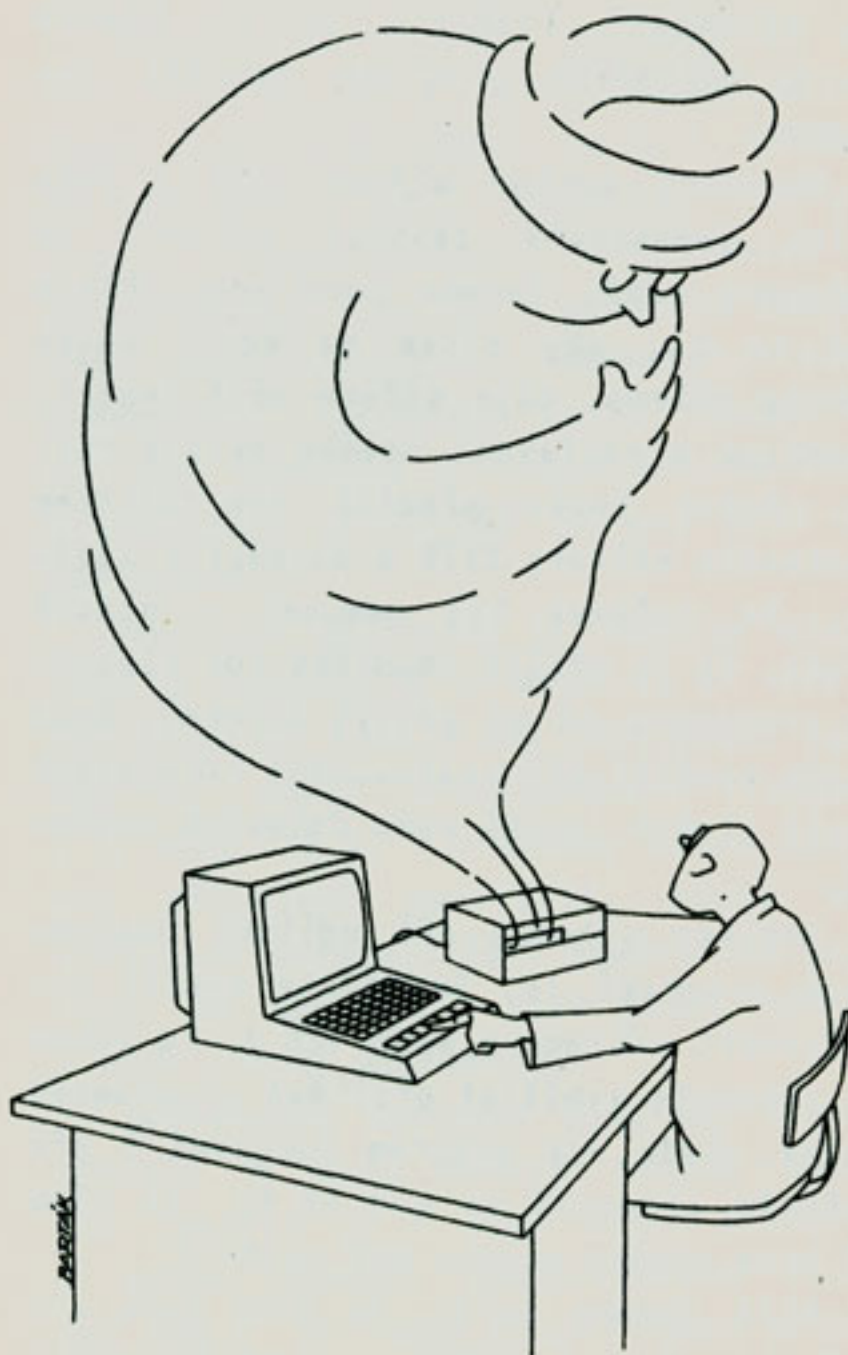
## Závěr

Poněkud nepraktické je to, že výrobce nezajistil, aby port RS232 byl tzv. full-duplex, tedy schopen vysílat i přijímat data v obou směrech současně. Tak je veškerá možnost připojení modemu zase ztracena - pro jeho připojení musíte mít další zařízení. Port RS232 u S128 je dobrý jen pro spojení s tiskárnou či do komunikační sítě s jiným počítačem. Otázkou zůstává, jak se bude S128 chovat při připojení Interfejsu 1, plnou odpověď dá až praktická zkouška.

Ve Španělsku se Spectrum 48 K prodává za 27000 peset (122 Lstg.), Spectrum plus za 36000 peset (164 Lstg.) a S128 za 50000 peset (230 Lstg.). Tato cena je pro britský trh nepřijatelná; rozhodně by neměla v Anglii přesáhnout 180 Lstg.

Jaké tedy je S128? Větší paměť a rozšířené zvukové možnosti vytvářejí předpoklady pro efektivnější programové aplikace. Bohužel však chybějí funkce Interfejsu 1. Staříčkový ZX BASIC zůstává beze změn; jeho minimální rozšíření nedává možnost adekvátnějšího využití nového hardwaru. Jaká bude verze S128 pro Británii, uvidíme. Ale určitě by měla být lepší.

Your Spectrum 21/85





# Způsoby ovládání kurzoru

Existuje něco takového jako perfektní zařízení pro řízení pohybu kurzoru? Na tuto otázku lze těžko odpovědět jednoznačně. Některé výzkumy ukazují na to, že volba způsobu ovládání kurzoru je více věcí osobního citu - někdo dává přednost tlačítkům před joystickem, jiný naopak ... Nutit někomu způsob, který mu nesedí, nikam nevede. Proto by mělo být ponecháno na osobní volbě každého, kdo počítač používá, aby kurzor mohl ovládat tak, jak je mu to nejpřirozenější.

Dále je uveden přehled všech dosud existujících i vyvíjených způsobů ovládání kurzoru, z nichž mnohé se uplatňují především v systémech CAD (computer aided design) pro tvorbu počítačové grafiky a její využití v řadě profesí.

## DIGITALIZOVANÁ TABULE (Digitizer Tablet)

Na základě kapacitní nebo elektromagnetické interakce mezi perem nebo "pukem", kterým se pohybuje po tabuli a tabulí samotnou se elektronicky zjišťuje jejich vzájemná poloha, jíž se určují souřadnice kurzoru. Jedná se o jeden z nejoblíbenějších způsobů převodu kresby z předlohy na obrazovku. Tabule může vedle toho plnit i některé jiné funkce, např. po doteku perem na určitém místě tabule se provede některá z možností volby funkcí menu grafického programu.

## "STOPUJÍCÍ" DIGITIZÉR (Tracking Digitizer)

Používá se takřka výhradně při práci s mapami. Je víceméně obdobou výše uvedené tabule. Ve variantě automatizované ("motorizované") citlivý senzor, nasazený na určitou čáru ("stopu"), je schopen ji sledovat a přenášet její tvar do paměti počítače. Indikace pozice senzoru může být odečítána např. vzájemným poměrem natočení dvou potenciometrů v rameni pantografu. Zařízení se vyrábí i v provedení s ručním snímáním čáry sledované posouváním lupy po předložce.

## MYŠ (Mouse)

Jedná se o elektromechanický (výjimečně elektrooptický) způsob přenosu pohybu koule v držáku. Koule musí být ve styku s jakoukoli pevnou podložkou. Tlakením shora



na držák při současném "koulení" po podložce se mění i pozice kurzoru na obrazovce ve směru, odpovídajícím směru pohybu myši. Mnozí lidé mají s kreslením myší značné problémy. Rádi ji však používají při "bodové" volbě funkcí menu, resp. při tzv. icon driven games, tedy hrách, které ovládáme umístěním kurzoru na některé z proměnlivě zobrazovaných symbolů (např. hra Shadow Fire). Oproti výše uvedeným dvěma variantám, které jsou z hlediska ovládní pozice kurzoru absolutní (určité místo v řídicím zařízení vždy odpovídá určité souřadnici v obrazové síti počítače), je myš pozičně relativní (nezáleží na tom, kde s ní začneme běhat po podložce - jedná se jen o přenos pohybu, nikoli umístění myši). Myši elektrooptické potřebují speciálně linkovanou podložku.

### SVĚTELNÉ PERO (Lightpen)

Je poměrně sporným, pro mnohé obtížným způsobem kresby na obrazovce. Jednou ze vžitých mylných představ je, že pero přímo kreslí. To ovšem není pravda - pero pouze čte světelný paprsek a na základě časovaných testovacích cyklů ve vztahu k časovému průběhu paprsku vytvářejícího televizní obraz, se jeho sejmutím v místě přiložení senzorem vybaveného pera přečte jeho pozice a převede na souřadnice grafické sítě obrazové paměti počítače. Trochu nepříjemnou součástí kresby světelným perem je, že nám neodpadá nutnost nějak ovládat kurzor. Počítače, které reagují na dialog mezi perem a obrazovkou trochu pomaleji, se snadno mohou stát mučícím nástrojem svého uživatele (těžknoucí ruka, svalové křeče, únava atd.).

### JOYSTICK

s určitou dávkou ironie překládaný coby veselá hůl. Používá se především pro hraní počítačových her. Pro přesnější ovládní kurzoru se však příliš nehodí. Jeho obdobou je joydisk (talířek volně uložený na kontaktech, pohyb se vyvolává tlakem), s nímž přišel Tektronix. Má však naprosto stejné nedostatky.

### STOPOVÁ KOULE (Trackball)

se velmi často používá v počítačem řízených vojenských zařízeních. Její komerční využití je však poměrně vzácné. Jedná se o kouli z vící velikosti koule kulečnickové, která je umístěna v držáku ("kapse"). Otáčením koule v dlani se mění pozice kurzoru na obrazovce.

### DOTEKOVÁ OBRAZOVKA (Touchscreen)

Tento způsob ovládní kurzoru i volby funkcí má několik variant. Žádná z nich se však nijak neprosadila. Důvodem je to, že lidský prst není patřičně zašpičatělý, aby mohl udat přesnou pozici kurzoru v jemné grafické síti. A pokusy nahradit jej "ukazovací" jehlou se rovněž neujaly pro nepříjemné pocity, jaké vyvolává styk kovu se sklem.



**DOTEKOVÁ PODLOŽKA  
(Touchpad)**

Oproti dotekové obrazovce je podložka ještě méně populární. Má obvykle rozměry cca 3 x 2 palce. Změny pozice kurzoru se dosahuje tlakem prstu na její povrch. Bohužel reakce kurzoru, co do jeho umístění, je poněkud nepředvídatelná a rovněž nelze opomenout ochabnutí svalstva prstu plynoucí z nutnosti neustálého vyvíjení tlaku na podložku.

**PALCOVÁ KOLEČKA  
(Thumbwheels)**

v počtu dvou částečně vyčnívají z nějaké krabičky (třeba klávesnice). Jejich otáčením palcem a ukazováčkem měníme horizontální a vertikální pozici kurzoru na obrazovce. Jsou využitelná spíše jen pro najetí kurzorem na nějaké místo obrazovky. Pro kreslení se moc nehodí, přestože v některých zařízeních se, i když jen velmi zřídka, objevují i dnes.

**KURZOROVÁ TLAČÍTKA  
(Arrow Keys)**

Velkou měrou pro ně platí totéž, co pro předchozí kolečka.

**HLAS  
(Voice)**

Ačkoli zařízení pro ovládání funkcí počítače hlasem již nejsou nijak nákladná, pro ovládání kurzoru se nehodí (nejen proto, že bychom na počítač museli neustále pokřikovat). Existují však i velmi promyšlené grafické systémy, kde lze širší paletou jednoduchých příkazů hlasem zadávat kresbu složitých tvarů (dokonce se jim říká speaker-independent, tedy nezávislé na veliteli) - jsou však extrémně drahé. Ty levné naopak mohou velmi dobře posloužit uživatelům CAD pro uvolnění rukou od četných tlačítky zadávaných funkcí programů.

**GRAFICKÉ ZOBRAZOVAČE  
(Plotters)**

Mnoho plotterů může být využito obdobně jako digitalizovaná tabule (resp. "stojící" digitizér). Pohyb hlavy zobrazovače se řídí např. joystickem apod., přičemž počítač převádí pozici hlavy (obvykle vybavené lupou) na pozici kurzoru na obrazovce. Tento způsob vyžaduje určitou praxi.

**PEDÁLOVÁ MYŠ  
(Foot-controlled-Mouse)**

Spíše by se měla nazývat pedálovým joydiskem. Je položena na podlaze a ovládána pohyby na ní položené nohy. Zkušenosti s ní jsou zatím příliš malé.



## SNÍMAČE POHYBU OČÍ (Gaze Input)

byly vyvinuty pro vojenské piloty. Zaměřují střelbu podle zaměření pilotova pohledu (skutečně "pohled, který zabíjí"). V komerčním využití výpočetní techniky se tento typ ovládání kurzoru zatím neobjevil a je zde uveden pro informaci o tom, co nás může ještě čekat.

## MYŠLENKOVÉ ŘÍZENÍ (Thought Control)

Ač se to zdá být neuvěřitelné, vývoj zaměřený tímto směrem přinesl své ovoce. Snímače některých slabých elektrických polí, vznikajících při myšlení v určité části lidského mozku, mohou ve výsledku řídit počítač podobně jako je tomu např. při jeho řízení hlasem. Princip věci tkví v tom, že počítač napřed musíte naučit rozeznávat vzorek vašeho elektrického pole (podobně jako u hlasu), jemuž přiřadíte určitý funkční význam (ovládání kurzoru nebo jiných programových funkcí). Tato zařízení jsou dokonce poměrně laciná. Je však zvláštní, že lidi s nimi neradi pracují (zřejmě z pocitu i obav ze zasahování techniky do své osobnosti).

Výběr typu ovládacího zařízení je však jen na vás. Výzkumy, které se snažily přijít na to, které z uvedených zařízení je nejefektivnější, došly k závěrům, že efektivita jejich využití neleží tolik v nich samých, jako v tom, kdo je ovládá - resp. že se můžete naučit efektivně ovládat takřka kterýkoli ze současných řídicích systémů.

Computer Graphics World 5/85

# Světelný počítač

Lidé mají tendenci považovat počítač spíše za elektronický než logický stroj. Stejně tak bychom neměli zapomínat, že logické procesy nemusí probíhat jen v zapojeních s polovodičovými prvky. Jakákoli hmota, která může rychle měnit svůj stav - ať už chemická, fyzikální, či dokonce biologická - může teoreticky plnit funkce počítačového hardwaru.

Dnes už lze s jistotou říci, co bude dalším článkem v řetězci technologického vývoje součástek pro výpočetní techniku. Bude to laserový transfázor, resp. optický tranzistor.

Současná technologie doslova vydupala z polovodičových čipů jejich praktické maximum. Rychlost překlápění spínacích tranzistorů, obsažených v integrovaném



obvodu, se dostala ke svému teoretickému limitu - času překlopení kolem jedné nanosekundy. Transfázory jsou tisíckrát rychlejší; přechod z jednoho stavu do jiného jim "zabere" kolem jedné pikosekundy! Jsou i mnohostrannější - např. základní logické operace AND, OR, NOT mohou být prováděny jediným transfázorem (místo propojení několika polovodičových prvků). Z toho vyplývá i vyšší kompaktnost "světelného" hardwaru. Velkou předností laserových počítačů bude možnost přímého přenosu jejich světelných dat bez nutnosti převodu na nějaký "komunikační" formát. Počítačové sítě, propojené světelným vláknem, tak budou mnohem jednodušší.

Experimentální zařízení používají transfázory přepínané malými změnami intenzity laserového paprsku. Optický tranzistor je v podstatě krystalem, který mění svůj index lomu v závislosti na intenzitě světla. Vhodnou volbou krystalického materiálu a vlnové délky laserového světla může být zkonstruováno zařízení, v němž malými změnami světelné intenzity dosáhneme velkých změn přenášeného světla.

Tento princip byl objeven už v roce 1896 a demonstrován pomocí Fabry-Perotova interferometru, který sestává ze dvou částečně průchodných zrcadel, oddělených mezerou. Při vyslání paprsku na čelní zrcadlo je intenzita druhým zrcadlem odraženého světla nižší. Zvažovalo se, jak by bylo možno řídit vztah mezi parametry světelného vstupu a výstupu. Bohužel, v této sestavě toho nelze dosáhnout jinak, než změnou charakteristik zrcadel a šířkou mezery.

Problém byl vyřešen použitím zdroje koherentního světla (laseru) a materiálu (např. antimonidu india), jehož index lomu (poměr rychlosti světla ve vakuu k jeho rychlosti v daném materiálu)

není pevný, ale mění se v závislosti na intenzitě a vlnové délce světla v "mezeře" krystalického materiálu. Výsledkem je prvek s optickou bistabilitou, v němž dochází k prudké změně při dosažení určité intenzity vysílaného koherentního světla.

Tento revoluční objev přinese do světa počítačů jeden ohromný zvrat. Leží před námi zcela nová organizace počítačové logiky - dnes vyvinuté transfázory totiž umějí přepínat nejen z jednoho stavu do druhého a zpět, ale mohou pracovat i s vyšším počtem logických úrovní než tradičními dvěma.

Kromě toho může v transfázoru probíhat několik operací současně, aniž by se navzájem rušily. Tak třeba pět (ale i mnohem víc) různých operací může být odvozeno z jednoho zdroje světla, rozděleného do pěti částí (svazku paprsků). Tento způsob práce s daty dostal název parallel processing.

Pokud ovládáte alespoň assembler nebo pracujete s hardwarem, jistě si dovedete domyslet, co to vše znamená - i když jakákoli dnešní představa bude zřejmě pokulhávat za tím, co nám invaze laserů do výpočetní techniky skutečně přinese.

Kdy budeme mít takový počítač doma, je stále ještě otázkou vyřešení některých podstatných technických problémů. Stále se hledají materiály vhodné pro výrobu integrovaných obvodů s optickými tranzistory - ty zatím objevené ostře mění svou charakteristiku v závislosti na okolní teplotě. Hledá se zdroj světla s malými rozměry a spotřebou při vysokém výstupním výkonu, aniž by "vytápěl". Podobně pokračují výzkumy v oblasti využití laseru pro záznam a snímání dat, a to jak na ploše, tak i ve vlastní trojrozměrné struktuře vhodných médií. Objevy nových typů laseru i nových re-



frakčních materiálů a jejich kombinací však na sebe určitě nenechají dlouho čekat.

-elzet-

# OSOBNÍ POČÍTAČE AMSTRAD/SCHNEIDER

Anglická firma Amstrad společně se západoněmeckou firmou Schneider vstoupily mezi výrobce osobních mikropočítačů poměrně nedávno - v létě roku 1984. Byl to však vstup pečlivě připravený a - jak je po téměř dvou letech vidět - velice impozantní. Vždyť které nové počítačové firmě se v dnešní těžké konkurenci podaří, aby se hned její první počítač stal ve své kategorii počítačem roku. A že se firma Amstrad snaží, aby její výše zmíněné úspěchy nezůstaly ojedinělé, o tom svědčí její dva poslední modely CPC 6128 a PCW 8254. Přitom jejich ceny nejsou nijak vysoké. Naopak (hlavně vedle fy Atari) patří Amstrad/Schneider mezi výrobce, kteří ceny na počítačových trzích srážejí prudce dolů.

Všechny počítače Amstrad/Schneider se dodávají v kompletní sestavě, tj. společně s barevným nebo zeleným monitorem a paměťovým médiem (kazetovým magnetofonem nebo třípalcovou disketovou jednotkou). Jejich uvedení do provozu je proto velmi snadné. Vážnějším zájemcům je dostupná i velmi rozsáhlá a podrobně zpracovaná dokumentace o operačním systému jednotlivých typů.

## Všeobecný popis

Osobní počítač Schneider sestává z vlastního počítače s profesionální klávesnicí QWERTY s oddělenou klávesnicí numerickou. Součástí počítače je kazetový magnetofon (u typu 464) nebo jednotka pružného disku 3" (typ 644 a 6128) a monochromatický nebo barevný monitor se zabudovaným napájecím zdrojem pro počítač. Počítač má i výstup RGB. Nabízený vf modulátor s vestavěným napájecím zdrojem umožňuje připojit i běžný TV přijímač.

Vestavěné stykové obvody dále umožňují připojit:

- a) dva ovládače (joysticky)
- b) tiskárnu (paralelní výstup Centronics)
- c) nf stereo zesilovač
- d) kazetový magnetofon (u typů 644 a 6128)
- e) pohonnou jednotku diskety 3"
- f) pohonnou jednotku disku Vortex 1,4 MB
- g) pevné disky Winchester 10 MB
- h) 256 paměťových modulů ROM
- i) paměť RAM až do 512 kB
- h) další interfejs (digitizér, svět. pero, modem apod.)



## Struktura počítače

Deska počítače obsahuje následující obvody:

Mikroprocesor - Z80A (4 MHz)  
 RAM - 64K (CPC 464, 644), 128 K (CPC 6128)  
 ROM - 32K(CPC 464), 48 K (CPC 644, 6128)  
 ULA - časování, DMA  
 9845 - řízení monitoru, grafika  
 AY-3-8912 - tři zvukové kanály, i šumový, 8 oktáv  
 PPI 8255 - ovládání tiskárny, magnetofonu

## Programovací jazyk

Implikovaný BASIC byl vytvořen firmou Locomotive Software. Je skvěle disponován pro programování, editace a ladění vlastních programů. Největší předností interpreteru je jeho rychlost. Standardní testovací programy jasně ukazují, že Amstrad patří mezi nejrchlejší osmibitové počítače. Ve srovnání s počítačem ZX Spectrum je Schneider čtyřikrát rychlejší a předčí dokonce i některé šestnáctibitové počítače (např. IBM PC).

## Seznam instrukcí BASICu

ABS, AFTER, ASC, ATN, AUTO, BIN↓, BORDER, CALL, CAT, CHAIN, CHR↓, CINT, CLEAR, CLG, CLOSEIN, CLOSEOUT, CLS, CONT, COS, CREAL, DATA, DEF, DEFINT, DEFREAL, DEFSTR, DEG, DELETE, DI, DIM, DRAW, DRAWR, EDIT, EI, ELSE, END, ENT, ENV, EOF, ERASE, ERL, ERR, ERROR, EVERY, EXP, FIX, FN, FOR, FRE, GOTO, GOSUB, HEX↓, MINEM, IF, INK, INKEY↓, INP, INPUT INSTR, INT, JOY, KEY, LEF↓, LEN, LET,

LINE, LIST, LOAD, LOCATE, LOG, LOG10, LOWER↓, MAX, MEMORY, MERGE, MID↓, MIN, MOD, MODE, MOVER, NEXT, NEW, NOT, ON, ON BREAK, ON ERROR GOTO, ON SO, OPENIN, OPENOUT, OR, ORIGIN, OUT, PAPER, PEEK, PEN, PI, PLOT, PLOTR, POKE, POS, PRINT, RAD, RANDOMIZE, READ, RELEASE, REM, REMAIN, RENUM, RESTORE, RESUME, RETURN, RIGHT↓, RND, ROUND, RUN, SAVE, SGN, SIN, SOUND, SPACE↓, SPC, SPEED, SQ, SQR, STEP, STOP, STR↓, STRING↓, SWAP, SYMBOL, TAB, TAG, TAGOFF, TEST, TESTR, THEN, TIME, TO, TRON, TROFF, UNT, UPPER↓ USING, VAL, VPOS, WAIT, WEND, WHILE, WIDTH, WINDOW, WRITE, XOR, XPOS, YPOS, ZONE.

CPC 644 a 6128 mají navíc tyto povely:

CLEAR INPUT, COPYCHR↓, CURSOR, DEC↓, DERR, FILL, FRAME, GRAPHICS PAPER, GRAPHICS PEN, ON BREAK CONT

## Grafické možnosti

Textový a grafický výstup na monitor nebo TV přijímač si může uživatel zvolit ze tří variant:

	text		grafika
	/sloupce/	řádky/	/body/
a/ MODE 0	20	25	160 x 200
b/ MODE 1	40	25	320 x 200
c/ MODE 2	80	25	640 x 200

Uživatel má k dispozici paletu 27 barev, z nichž může využít maximálně 16 najednou. Plochu obrazovky lze rozčlenit až na 8 okének (WINDOWS), které se mohou vzájemně prolínat a v nichž lze jednotlivě pracovat s textem nebo grafikou. Programově lze užít 4 vnitřní čítače, které mohou v definovatelném časovém intervalu spustit čtyři různé podprogramy. Pomocí příkazu KEY lze 32 tlačítkům přiřadit různé funkce.



Zabudovaný kazetový magnetofon je programově ovládán a pracuje se dvěma rychlostmi přenosu dat - 1000 a 2000 baudů.

### Popis jednotlivých typů CPC 464

Tento model byl prvním počítačem firmy, s nímž vstoupila na trh v květnu 1984. Je postaven na bázi mikroprocesoru Z80A s kmitočtem hodin 4 MHz. Obsahuje 64 kB RAM a 32 kB firmwaru v ROM. Napájecí proud je přiváděn jedním kabelem do monitoru a dva kabely spojují monitor s klávesnicí. Jeden přenáší video signál, druhý slouží napájení. Klávesnice obsahuje mikroprocesor, paměti a kazetový magnetofon. Monitor je buď zelený nebo barevný. V roce 1984 bylo prodáno asi 200 000 ks, o rok později asi 600-000 ks CPC 464.

Horních 16K RAM tvoří obrazová paměť. Asi 5K užívá operační systém a zbylých 43K je k dispozici uživatel programujícímu v BASICU. Aby Z80, který může adresovat max. 64K, obsáhl jak RAM, tak i 32K ROM, je část paměti "zdvojená" tak, že ROMka je rozdělena do dvou bloků po 16K, které jsou "pozadím" paměti RAM - jsou umístěny v tzv. dolní části (0000H-3FFFH) a v tzv. části horní, umístěné mezi adresami C000H a FFFFH. Operační systém je v části dolní, překladač BASICu v části horní - ta obsahuje i video RAM. Horní část je možno softwarově přepínat mezi 256 externími ROMkami, které mohou obsahovat další uživatelské programy.

Konstrukce (zvláště klávesnice) je velmi dobrá. Vedle klasické klávesnice QWERTY je dobře umístěna číselná klávesnice pro rychlé zadávání numerických dat. Tlačítka lze rovněž předefinovat pro zadávání různých funkcí (i jejich

souborů) a výrazů. Kurzorové klávesy jsou umístěny nad numerickou klávesnicí.

K ovládání her slouží joystick podobného typu, jaký se používá u počítačů Commodore a Atari. Připojuje se přímo k uživatelskému portu počítač (zezadu). Joystick je vybaven dalším portem pro připojení druhého joysticku.

Zobrazení na monitoru je řízeno obvodem HD6845SP. Lze jej softwarově programovat a dosáhnout tak zajímavých obrazových efektů. Obvod dále umožňuje snadnější připojení grafických periférií.

Barevný monitor obsahuje obyčejnou čtrnáctipalcovou TV obrazovku. Přesto je obraz mnohem klidnější a zřetelnější než na běžných televizorech. I zelený monitor má diagonálu 14". Čtení z ní je ještě příjemnější než u monitoru barevného. Abychom v případě potřeby transportu počítače nemuseli přenášet i monitor, lze si opatřit televizní adaptér obsahující napájecí jednotku počítače, s nímž dohromady váží méně než 4 kg.

Všechny tři zobrazovací módy umožňují současné užívání textu i grafiky s výběrem barev, jejichž počet je závislý na zvoleném módu - MODE 2/2 barvy/, MODE 1/4 barvy/, MODE 0/16 barev z možných 27/. Barvy mohou pulsovat ve všech třech módech, a to s volitelnou dobou jednotlivých pulsů. Ze základních 27 barev lze zvolit jednu pro pozadí.

Dobrý výběr grafických příkazů napomáhá využití operační rychlosti počítače při vytváření grafického výstupu. Příkazy DRAW, PLOT a MOVE jsou zastoupeny jak v relativní, tak absolutní formě. Všechny mají stejnou syntaxi, bez ohledu na zvolený grafický mód. Systém neumožňuje sprajty; rychlé animované efekty lze tvořit příkazy SYMBOL, LOCATE, INK. Příkaz SYMBOL umožňuje definovat vlastní



množinu znaků. LOCATE přemísťuje kurzor, INK nahrazuje zadanou barvu barvou jinou.

K dobrému využití obrazovky přispívá možnost tvorby oken (uživatelé definovaných částí obrazovky obdélníkového tvaru, fungujících jako "malé obrazovky", v nichž lze nezávisle volit barvu pozadí, nezávisle je mazat, přičemž vše, ležící mimo oblast okna, zůstává nezměněno, atd.). Celkem může být definováno 8 textových oken a i grafické. Okna se mohou libovolně překrývat. Volba okna, do něhož chceme na obrazovce psát, se provádí pomocí "streamu" (symbol#).

Jednotlivé "streamy" jsou indentifikovány čísly od 0 do 9. Slouží k označení vstupu a výstupu. #0-#7 indentifikují textová okna, #8 vstup klávesnice a výstup tiskárny, #9 magnetofon nebo diskovou jednotku. Tak lze jednoduše zadávat příkazy vstupu a výstupu. Např. PRINT #i, a vypisuje hodnotu proměnné a; pro i=0, ..., 7 do příslušných textových oken, pro i=8 na tiskárně a pro i=9 zaznamená hodnotu proměnné na záznamové médium.

V počítači je též zvukový generátor, jaký používají počítače MSX. Syntaxe příkazů pro jeho ovládání je však odlišná. Každý ze tří kanálů generátoru zahrnuje 8 oktáv. Výstup směřuje do vestavěného reproduktoru (ovladatelná hlasitost) a do stereo výstupu pro případné zesílení vnějším zesilovačem.

Jednoduchý zvukový příkaz má formu SOUND(kanál, tón, trvání). Složitější příkazy umožňují formovat tvar zvukové vlny specifikací tónové i amplitudové obálky (podobně jako u syntezátoru).

4 nezávislé interní časové čítače jsou přístupné z BASICu. Mohou být využity ke spuštění až tří nezávislých podprogramů. Tvar příkazu je EVERY(časová

delka, číslo čítače)GOSUB, nebo AFTER(časová delka, číslo čítače)GOSUB. Podle zvoleného příkazu je provádění programu buď pravidelně nebo jednorázově přerušeno a je spuštěn příslušný podprogram.

Přínosem jsou dobré editovací a odlaďovací dispozice počítače. Listování programem na monitoru lze jednoduše zastavit i spustit, celý program (nebo jeho část) může být přečíslován. Po příkazu AUTO počítač čísluje programové řádky automaticky. COPY-kurzorem lze přepisovat a kopírovat text, který je již někde na obrazovce zobrazen. Kromě toho lze používat i tradičnější řádkově orientovanou editaci.

Odlaďování programu napomáhají běžná chybová hlášení s uvedením místa a druhu chyby a příkaz ON ERROR GOTO, kterým můžeme program při výskytu chyby větvit a následně ji zpracovat. Trasování slouží příkaz TRON, který průběžně vypisuje číslo právě prováděné řádky programu.

Funkce magnetofonu jsou řízeny programově. Během čtení a zápisu jsou všechny ostatní funkce přerušeny, včetně časových čítačů. Záznam na pásku je uveden hlavičkou, která obsahuje informace o typu dat (basicový program, binární data, ASCII data, chráněná data, ev. jiný typ dat dle volby uživatele), délku dat, adresu, od níž mají být uložena a v případě programu i adresu, od níž má být program spuštěn. Záznamy jsou členěny do bloků po 2 kB, přičemž každý blok má svou vlastní, zkrácenou hlavičku. Zapisovat však lze i "v jednom kuse". Rychlost záznamu je nastavitelná. Při čtení se operační systém automaticky časuje podle informace, kterou odebírá ze zaváděcího signálu záznamu. Zařazena je i obdoba uživatelům Sinclairů známé funkce MERGE.

Výstup Centronics má bohužel nejvy-



šší bit vždy ve stavu log.0, takže je vlastně sedmibitový. Pro disketové jednotky firma nabízí svůj vlastní systém AMSDOS, který je jen malou modifikací systému ovládání vestavěného kazetového magnetofonu. Vážnější zájemci mohou využívat systém CP/M 2.2, který otevírá přístup k cca 8000 programů profesionálního softwaru. Je však nutno vzít na vědomí, že velký počet těchto programů vyžaduje, aby počítač měl rozsah RAM 128K, čehož lze dosáhnout připojením vnějších modulů (až do 512K).

Disketová jednotka s jednou snímací hlavou je standardem HITACHI 3". Jedna strana diskety může nést 100 kB záznamu formátovaných dat. Oproti podstatě rozšířenějšímu standardu SONY 3,5" je tato jednotka lacinější a zajišťuje snadnější převoditelnost softwaru z tradičních záznamových médií. K počítači lze připojit ještě jednu jednotku formátu 3" nebo 5,25" fy Vortex (1,4 MB).

### CPC 664

Tento model byl dán na trh v květnu 1985. Ale již v září byl nahrazen výkonnějším typem CPC 6128. Protože jej firma rychle stáhla z prodeje (především pro nepříliš vhodný operační systém), budeme se v našem průvodci zabývat rovnou

### CPC 6128

Jeho RAM obsahuje 128K. Pro basicové programátory je užitečných asi 41K. Vedlejších 64K slouží jako RAM disk - pro ukládání dat, programu ve strojovém kódu, nebo pro uložení až 4 obrazovek. Přístupu do této části paměti slouží dodávaný pomocný program Bankman. Lepšímu využití paměti napomáhá i operační systém CP/M Plus, který je kompatibilní

s CP/M 2.2 a zároveň umožňuje používat software vytvořený pro systém CP/M 80 bez jakýchkoli paměťových omezení. Kompatibilita s CP/M 2.2 umožňuje používat všechny software vytvořený pro CPC 464 bez jakýchkoli úprav.

Bankman je z Basicu přístupný příkazy RSX. Paměť RAM je členěna do bloků po 16K, které mohou být střídavě zapínány a vypínány podobně jako je tomu u paměti ROM v případě CPC 464. Paměťové bloky po svém zapnutí vždy zaberou oblast adres 4000H-7FFFH. Protože obrazovka zabírá 16K paměti, lze přidaných 64K využít pomocí příkazů SCREENCOPY nebo SCREENSWAP k uložení čtyř různých obrazovek. I když tyto příkazy nejsou prováděny tak rychle, aby je bylo možno využít pro animaci, je jejich rychlost dostatečná pro celou řadu dalších aplikací. Jednou z nich je např. možnost vytvoření MENU obrazovek s jejich začleněním do programu tak, aby jejich vyvoláním nebyl zničen obsah hlavní obrazovky. Tyto obrazovky bývají využívány pro zobrazení různých diagramů, grafů apod.

### PCW 8256 JOYCE

Tento, zatím poslední výrobek firmy, poněkud vybočuje z kontinuity dosavadní řady počítačů CPC. JOYCE má sloužit především jako levný, avšak výkonný textový editor, resp. slovní procesor. Tomu odpovídá i základní konfigurace, v níž je dodáván - 256K RAM/ z toho 116K pracuje jako RAM disk/, i disketová jednotka 3" s kapacitou 178K po každé straně diskety, zelený monitor, klávesnice a maticová tiskárna s módem NLQ. Lze přidat další floppy jednotku, podobně jako u CPC 464. Obě jsou umístěny kolmo vedle obrazovky monitoru. Mikroprocesorem je opět Z80. Přepínání



paměť je obdobné systému CPC 6128.

Monitor zobrazuje 32 řádek v 90 sloupcích. Přejít na tradičnějších 80 znaků na 25 řádkách umožňuje "screen driver" systému CP/M.

Tiskárna pracuje se všemi obvyklými typy papíru. Její vybavení je na úrovni tiskáren střední třídy. Přestože píše v obou směrech, je rychlost jejího tisku poměrně malá. Tisk je ovládán přímo z klávesnice počítače, proto tiskárna nemá žádné ovládací prvky. Pohodlí práce s tiskárnou zvyšuje šachta na archy papíru, z níž jsou listy odebírány a do tiskárny vkládány automaticky.

Určitou zajímavostí je, že JOYCE neobsahuje žádný interpretor ani kompilátor, tedy žádný jazyk. Obsahuje pouze jednoduchý systém. Vše potřebné je nutno do něj načíst z disket. Odečteme-li tak od 256K RAM prostor potřebný pro BDOS a BIOS, pro RAM disk a video RAM, zbývá ještě 61K, což je dostačující pro velkou část aplikací CP/M. V základním vybavení je k počítači dodán Dr Logo, Mallard BASIC od Locomotive Software, CP/M utility a samozřejmě LocoScript pro zpracování textu. Další softwarové firmy dodávají alternativní BASIC, Pascal (včetně verze Turbo), Cobol,

Fortran, Lisp, Prolog a C. JOYCE může pracovat s programy v jejich široké škále od dBaseII po WordStar. Přitom je cena softwaru adekvátní ceně počítače.

### Aplikace

Příslušenství osobních počítačů Amstrad/Schneider se rozšířilo nejen o paměťové moduly 64KB až 512K, ale i o digitizéry, světelná pera a myši. Takové příslušenství již umožňuje využívat prvky CAD v běžné praxi projektových, výrobních a dalších organizací bez vysokých pořizovacích nákladů. Počítače lze propojovat do lokálních sítí s centrální databankou. Přenos v takové síti lze snadno uskutečnit pomocí operačního systému CP/M. Firma Vortex připravuje IBM PC emulátor se 16tubitovým procesorem, který umožní pracovat s programy v systému PC DOS. Tak by byla zajištěna kompatibilita s počítači IBM PC.

Jaromír Šiška  
Ota Luňák





# Herbář nápadů a zkušeností

## Kompresse a expanze obrazových dat

Vyskytují se případy, kdy bychom s povděkem přijali možnost "stlačení" rozsahu 6912 bajtů obrazové paměti ZX Spectra. Kompresse dat šetří především plochu záznamového média. Používáme-li k záznamu kazetový magnetofon, pak kompresí můžeme výrazně zkrátit i nahrávací a snímací čas.

Níže uvedený program je obdobou těch několika podobných, které "běhají" mezi majiteli ZX Spectra. Základním principem komprese je sloučení řady stejných bajtů, které obsahují nulu nebo FFH. Tak např. obsahuje-li paměť na adresách 16384 až 16484 nulové bajty, bude těchto 101 bajtů zkomprimováno na 3 bajty - první obsahuje informaci o tom, jaké bajty byly komprimovány, tedy nulu. Další dva bajty obsahují počet zkomprimovaných bajtů, přičemž první z těchto dvou bajtů obsahuje nižší část, druhý vyšší část celkového počtu zpracovaných bajtů. V uvedeném případě to bude decimálně 0 101 0. Obdobně je tomu u řady bajtů FFH - v tom případě bude identifikační bajt pochopitelně FFH. Např.

při kompresi 513 ( $2 \cdot 256 + 1$ ) bajtů FFH bude její výsledek 255 1 2 decimálně, resp. FF 01 02 hexadecimálně. Všechny bajty, které neobsahují ani 0, ani FFH jsou uloženy do tabulky komprimovaných dat samostatně, bez jakékoli identifikace - tedy co jeden bajt obrazové paměti, to jeden bajt v tabulce komprimovaných dat. Nevýhoda tohoto postupu se objeví tehdy, když obrazová paměť obsahuje příliš velký počet samostatných bajtů 0 nebo FFH, resp. i jejich párů. Protože výsledkem komprese nul nebo FFH jsou vždy 3 bajty, v uvedených případech dosáhneme zcela opačného efektu komprese. Vzhledem k tomu, že tyto případy nejsou příliš časté, není nutno se jimi v amatérské praxi zabývat. V programu je zařazen test, který vás na takový případ upozorní - pak se záznam komprimovaných dat neprovede. Kdybychom chtěli takováto data komprimovat, museli bychom vytvořit program, který by detektivně prošetřil, zda a jaké sestavy bajtů se v obrazové paměti vyskytují nejčastěji, přidělit jim spe-



ciální kód a vytvořit tabulku pro jejich dekódování mezi ostatními nekomprimovanými bajty.

Uvedený postup - selsky řečeno - "vymlácení" bajtů 0 a FFH je užít pro kompresi kresby (bez atributů), tedy 6144 bajtů obrazové paměti v rozsahu adres 16384 až 22528 (57FFH) včetně. Na adrese 5800H je uložen první ze 768 bajtů atributů, poslední je na adrese 5AFFH. Od adresy 5B00H začíná tiskový buffer (viz manuál ZX Spectra).

Kompresi atributů nevěnuje zvláštní pozornost bajtům 0 a FFH, ale přistupuje ke všem stejně. Zjišťuje tedy, zda v řadě atributů existuje jakýkoli počet shodných bajtů, kterým pak přiřazuje identifikaci podle výše uvedeného algoritmu. Nevýhoda postupu je stejná jako u kresby. Narazíme-li na zvlášť "probarvenou obrazovku", může snadno dojít k tomu, že výsledek komprese atributů přesáhne číslo 768.

Tento nedostatek nebyl u žádného z komprimačních programů, které jsem zatím viděl, nijak řešen. Program s atributy pracuje tak, že nejdříve všechny převede do 768 adres, počínaje adresou 32000. Pak se provede komprese. Nastane-li nevídaný případ, jsou původní atributy zařazeny za poslední bajt tabulky komprimované kresby. Při expanzi jsou atributy přeneseny instrukcí LDIR na své místo. Díky jednoúčelovosti programu nikterak nevadí, že do paměti RAM přidáme "navrch" ještě oněch 678 bajtů od adresy 32000. Věc by šla řešit i dekompresí "přetečených" atributů. Zvolené řešení však považuji za jednodušší.

Oblasti nekomprimovaných atributů od adresy 32000 využívám pro možnost jejich samostatného zobrazení. Samostatně lze zobrazit i jen kresbu. Myslím, že v mnoha případech nebude nezajímavé se podívat na obě složky obrazu zvlášť.

Pomocné programy u BASICu provádějí úkony, které by bylo zbytečně pracné převádět do strojového kódu. U takto zřetelně jednoúčelového programu by to byl i zbytečný luxus. BASIC komunikuje s uživatelem programu a informuje o kompresi dosažených výsledcích. Tak můžete sami rozhodnout, zda jich z hlediska efektivity využijete.

Program BASICu pro kompresi a expanzi jedné obrazovky ukládá na patřičnou adresu (32772) informační kód - je-li 0, na pásek se nahraje komprimovaná tabulka dat kresby bez atributů; je-li 1, komprimovaná tabulka dat kresby včetně atributů; je-li FFH, bude ke komprimované kresbě přiřazeno 768 atributů v jejich původním provedení. Tento kód je velmi důležitý při expanzi dat.

Další program v BASICu, který by měl sloužit spíše jen jako vodítko, je rozšířen o možnost komprimace většího počtu obrazovek - komprimovaná data se ukládají postupně do volné paměti. Takto lze do jednoho bloku dat zkomprimovat (i s informačními kódy) třeba až 9 obrázků. Celý tento blok můžete zaznamenat na pásek, kdykoli jej načíst a jednotlivé obrazovky postupně expandovat (i tisknout). Volbě slouží včleněné menu. Příkazy pro záznam a čtení dat mají syntaxi pro Wafadrive. Příkazy si pochopitelně upravte podle toho, jak to vyžaduje vaše vlastní záznamové zařízení. Program je vybaven již i voláním neuvedené screen-dumpové rutiny pr1 a pr2 (tovněž upravte dle svých potřeb).

Uvedeným programem tisknu serii předem komprimovaných 21 obrázků na tři "zátahy" (řádky 555 až 580). Obrázky (bez atributů; a=0) se postupně automaticky expandují a tisknou. Na list A4 se vejdou čtyři - proto pauza pro vložení nového listu (řádka 700). Při tisku



na perforovaný papír by nebyla potřebná, načítání i tisk by proběhly od začátku do konce samy. Tuto část programu si opět upravte podle sebe. Někdo možná nebude znát POKE 23658,8 - jím vyvoláte mód psaní velkých písmen (pro usnadnění testů vkládaných písmen u příkazů INPUT) Vše ostatní je zřejmé. V obou pomocných programech v BASICu vložená řádka 9998 slouží jen tisku výpisu programu přes paralelní port Mafadrivu.

Řešení komprese a expanze rutinami strojového kódu má řadu variant. Zvolil jsem řešení, které v případě komprese zabere o nějaký ten bajt navíc, je ale rychlejší. Komprese kresby během odběru řad nul a FFH probíhá bez testu adresy obrazové paměti. Proto je zpočátku na první adresu atributů uložen bajt 01 - při odběru všech bajtů odlišných od nuly nebo FFH test adresy probíhá. Po odběru posledního bajtu kresby je změněnému bajtu atributů vrácena jeho původní hodnota.

Expanze je jednodušší, ale trvá o nějakou tu mikrosekundu déle - cyklicky se provádějí "zbytečné" testy, je vynechána možnost přeskočení adres s obsahem 0 (např. instrukcí ADD HL,BC), a tak se i nuly do obrazové paměti ukládají instrukcí LDIR.

Rutina komprese obsahuje 114 bajtů, expanze 58 bajtů. Obě rutiny se spouštějí od své první adresy.

Jistě vás bude zajímat, jaká je úspora rozsahu obrazových dat. U běžného nákresu, v němž je řada vodorovných linek na jednobarevném podkladu, z cel-

kového počtu 6912 bajtů ušetříte kolem 3000 (ale třeba i 5000) bajtů. To rozhodně už stojí za to. Komprese "prázdné obrazovky" dává výsledek pouhých 6 bajtů (3 pro kresbu, 3 pro atributy). Je však nutno mít na zřeteli, že k nim přibude ještě 5 informačních bajtů z adres 8000H - 8004H.

Do programu bylo možno zařadit i dělení komprese na jednotlivé třetiny obrazové paměti. Pro praxi se však jedná o určitý luxus, který má využití dost zřídka. Účelná by taková komprese byla pouze tehdy, kdy by některá ze třetin obrazu dávala výrazně negativní výsledek (postup by pak byl obdobný výše uvedené "negativní" kompresi atributů). Zkušenější z vás si program pro tento účel i případné jiné aplikace snadno upraví.

Pro ty z vás, kteří se prokousávají základy programování mikroprocesoru Z80, jsem pro zajímavost přidal něco ze stylu zvaného self-modification. Je to metoda programování, která způsobuje modifikaci programových instrukcí v průběhu programu. Ve výpisu rutin jsou modifikace značeny " ". Týkají se adres MDF1 až MDF4. Chod rutin je zřejmý z připojeného komentáře. Mohu jen dodat, že je používám především pro tisk vlastních obrázků na tiskárně - s velkou úsporou času i plochy archivačního záznamového materiálu. Komprimované obrázky si rovněž můžete (ale nemusíte) zaznamenat i s rutinou expanze jejich dat, a pak je expandovat voláním adresy 65143.





## Výpis rutin komprese a expanze obrazových dat

```

ORG 65000
MDF1:EQU FE22H
MDF2 EQU FE34H
MDF3:EQU FEBCH
MDF4:EQU FEC1H
    
```

### Rutina KOMPRESSE

```

=====
FDEB 010003   START:LD BC,300H   ;Incializace čítače počtu bajtů
      21005B   LD HL,5800H   ;atributů
      7E      LD A,/HL/     ;Uschování 1.bajtu atributů
      F5      PUSH AF
      E5      PUSH HL
      11007D   LD DE,32000  ;1.adresa určení přenosu 768
      EDB0    LDIR         ;bajtů atributů
      E1      POP HL
      3E01    LD A,01      ;Uložení bajtu 01 na 1.adresu
      77      LD /HL/,A    ;atributů
      110580   LD DE,8005H  ;1.adresa tabulky komprimov.dat
      210040   LD HL,4000H  ;1.adresa obrazové paměti

FE00 7E      DOKOLA:LD A,/HL/ ;Přenos bajtu z obr.paměti do A
      A7      AND A        ;Je roven nule?
      2004    JR NZ,JEFF?  ;Když NE, skok na JEFF?
FE04 CD18FE   HOPO:CALL KOMPR ;Volání subrut.KOMPR
      7E      LD A,/HL/   ;Další přenos bajtu obr.paměti
FE08 FEFF    JEFF?:CP FFH   ;Je roven FFH?
      2007    JR NZ,FFANIO ;Když NE, není to ani 0, ani FFH
      CD18FE  CALL KOMPR  ;Volání subrut.KOMPR
      7E      LD A,/HL/   ;Další přenos bajtu obr.paměti
      A7      AND A        ;Je roven nule?
      2BF1    JR Z,HOPO    ;Když AND, skok na HOPO
FE13 CD2DFE   FFANIO:CALL ATR? ;Celá bodová síť zkomprimována?
      18EB    JR DOKOLA   ;Když ne,pokračovat v komprimaci

FE18 32FE18   KOMPR:LD /MDF1/,A ;Inicial.čítače bajtů 00 /*FFH/
      010000   LD BC,0000  ;Další adr.obraz.paměti
FE1B 23      ZNOVA:INC HL    ;Zvýšení obsahu čítače o 1
      03      INC BC      ;Další bajt obr.paměti pro test
      7E      LD A,/HL/   ;Je roven nule /*FFH/?
      FE00 /*FEFF/ CP 00    ;Když AND, pro další bajt
      2BF9    JR Z,ZNOVA

FE25 2B      TAB:DEC HL     ;Nalezení hodnoty komprimovaných
      7E      LD A,/HL/   ;bajtů, uložení 00 /*FFH/ do
FE27 12      TABATR:LD /DE/,A ;tabulky a na
      13      INC DE     ;dvě další adresy tabulky
      79      LD A,C      ;uložení počtu těchto bajtů jeho
      12      LD /DE/,A   ;přenosem z párového registru BC
      13      INC DE
      78      LD A,B
FE2D 12      ATR?:LD /DE/,A ;Zvýšení DE pro příští uložení
      13      INC DE     ;Příprava další adr.obraz.paměti
      23      INC HL
      3E57    LD A,57H    ;Je obsah HL vyšší než 57FFH?
      94      SUB H      ;Když AND, na ZMENA /*ATRIBUT/
      3801 /*380F/ JR C,ZMENA ;Když NE, návrat
      C9      RET
    
```



```

FE36 3E0F      ZMENA:LD A,0FH      ;Modifikace instrukce JR 01 na
3234FE      LD /MDF2/,A ;JR 0FH pro komprimaci atributů
1B          DEC DE
ED530080    LD /8000H/,DE;Uložení počtu dat bodové sítě
13          INC DE
F1          POP AF      ;Odhoz nepotřebné adresy návratu
F1          POP AF      ;Vyvolání a uložení původního
77          LD /HL/,A   ;bajtu na 1.adr.atributů

FE44 010000   ATRBUT:LD BC,0000 ;Inicial.čítače shodných bajtů
D5          PUSH DE
FE48 3E5A     OPET:LD A,5AH   ;Je obsah HL větší než 5AFFH?
BC          CP H
380B       JR C,HOTOVO ;Když AND, skok na adr.HOTOVO
7E         LD A,/HL/     ;Když NE, pokračování v testu
23         INC HL
5E         LD E,/HL/    ;Je další bajt shodný s obsahem
BB         CP E         ;reg.A?
03         INC BC       ;Zvýšení čítače o 1
28F4       JR Z,OPET    ;Když AND, otestovat další bajt
2B         DEC HL       ;Když NE, snížení HL o 1 a
D1         POP DE
18CF       JR TABATR   ;uložení parametrů do tab.dat

FE58 D1        HOTOVO:POP DE
1B         DEC DE
ED530280   LD /8002H/,DE;Uložení počtu všech bajtů tab.
3E01       LD A,01      ;Zpětná modifikace instr.JR 0FH
3234FE     LD /MDF2/,A ;na JR 01
C9         RET         ;Konečný návrat do Basicu

FE64 ED5B0080PRENDS:LD DE,/8000H/;Přenos atributů na adresy
13         INC DE      ;32000-32767
010003     KUK:LD BC,300H
21007D     LD HL,32000
EDB0       LDIR
C9         RET

FE72 11005B   KUKATR:LD DE,5800H ;Subrut.volaná z Basicu pro
18F2       JR KUK     ;zobrazení samostatných atributů

```

```

; =====
; Rutina EXPANZE
;

```

```

FE77 110480   EXPANS:LD DE,8004H ;Adresa informace o tom, zda
1A         LD A,/DE/   ;byla komprimována
47         LD B,A      ;všechna obrazová
05         DEC B       ;data nebo jen bodová síť nebo
3E5A       LD A,5AH    ;bude následovat přenos atributů
32BCFE     LD /MDF3/,A ;v jejich původní podobě
280C       JR Z,POCET  ;Skok na POCET
3E57       LD A,57H    ;Byla komprimována jen kresba
32BCFE     LD /MDF3/,A ;Modifikace obsahu adr.MDF3
04         INC B       ;Budou přenášena původní data
2804       JR Z,POCET  ;atributů? Když NE, na POCET
AF         XOR A
32C1FE     LD /MDF4/,A ;Modifikace obsahu adr.MDF4
FE90 13       POCET:INC DE ;1.adresa tab.komprimovaných dat
210040     LD HL,4000H ;1.adresa obrazové paměti

FE94 1A       ROZVES:LD A,/DE/ ;Test bajtu
A7         AND A       ;Je nulový?
2807       JR Z,TABULE ;Když AND, skok na subr.TABULE

```



FEFF 2803 77 1815		CP FFH JR Z, TABULE LD /HL/, A JR TEST	; Je to FFH? ; Když AND, skok na subr. TABULE ; Bajt není ani 0, ani FFH, proto ; je uložen sám; Skok na subr. TEST
FE9F 77 13 1A 4F 13 1A 47 0B 00 00 78 B1 2807 D5 E5 D1 13 EDB0 D1	TABULE:	LD /HL/, A INC DE LD A, /DE/ LD C, A INC DE LD A, /DE/ LD B, A DEC BC NOP NOP LD A, B OR C JR Z, TEST PUSH DE PUSH HL POP DE INC DE LDIR POP DE	; Přenos bajtu z adr. /DE/ do /HL/ ; Zjištění počtu bajtů pro jejich ; expanzi - převod z /DE/ a ; /DE+1/ do reg. BC  ; Sníž. BC o 1, aby při exek. LDIR ; nedošlo k přenosu o 1 adr. dál ; "Pozůstalé nopy" z experimentů ; Test, zda je BC=0  ; Je-li, jde jen o 1 bajt /atr./ ; Příprava pro expanzi bajtů FFH ; nebo 0 /pro kresbu/ a všech ; /kromě samotných/ bajtů ; atributů pomocí instrukce LDIR
FEB4 13 23 3E57 BC 30D9 3E5A /*3E57/ BC 1A D8 18DD /*1800/ 010003 EB EDB0 3EDD 32C1FE C9	TEST:	INC DE INC HL LD A, 57H CP H JR NC, ROZVES LD A, 5AH CP H LD A, /DE/ RET C JR TABULE LD BC, 300H EX DE, HL LDIR LD A, DDH LD /MDF4/, A RET	; Byly zpracovány všechny bajty ; obrazové sítě? ; Když NE, skok na ROZVES ; Modifikovaný bajt - když je 57H ; pak návrat, jinak provést ; expanzi atributů ; Podmíněný návrat do Basicu ; Skok na adr. tabule /*pokrač./ ; Přenos 768 atributů v případě ; jejich "negativní" komprese  ; Zpětná modifikace obsahu ; adresy MDF4 ; Návrat do Basicu



```

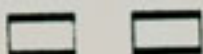
1 REM Vypis pomocneho Basicu pro komprimaci a expanzi jedne obrazovky
2 REM
15 CLS *: LET a=32772: POKE 23658,8: INPUT "WILL YOU LOAD "; INVERSE 1;"SCREEN*"; INVERSE 0;"
OR COMPRESSED "; INVERSE 1;"DATA?"; INVERSE 0;" (S/D) "; LINE n$: IF n$="D" OR n$="d" THEN LO
AD "CODE": GO TO 200
16 IF n$("<>S") THEN GO TO 15
17 CLS *: LOAD "SCREEN*
20 RANDOMIZE USR 65000
25 LET sit=PEEK (a-4)+256*PEEK (a-3)-a: LET vse=PEEK (a-2)+256*PEEK (a-1)-a: LET atr=vse-sit
30 CLS : PRINT AT 1,4; BRIGHT 1;" RESULTS OF COMPRESSION:"; LET s$="shorter": LET l$="longer "
: LET b$=" bytes "; LET d$=" Don't use it!"
35 PRINT AT 4,6; BRIGHT 1;" TOTALLY "; INVERSE 1;vse; INVERSE 0;b$
40 LET kosi=6144-sit: LET koat=768-atr: IF kosi<0 THEN PRINT BRIGHT 1"....." DOTS are ";ABS ko
si;b$:l$; FLASH 1;d$: GO TO 250
50 PRINT BRIGHT 1"....." DOTS are "; INVERSE 1;sit; INVERSE 0;b$"" (";kosi;b$;s$;)
60 IF koat<0 THEN RANDOMIZE USR 65124: LET ldir=3: PRINT BRIGHT 1"....." ATTR are ";ABS koat;b$
;l$; FLASH 1;"I've got 768 of them back again!"; PAUSE 100: GO TO 25
70 PRINT BRIGHT 1"....." ATTR are "; INVERSE 1;atr; INVERSE 0;b$"" (";koat;b$;s$;)" : LE
T ldir=0
80 INPUT "WILL YOU SAVE? (Y/N) "; LINE a$
90 IF a$="Y" THEN GO TO 100
95 IF a$("<>N") THEN GO TO 80
97 GO TO 150
    
```



```

100 INPUT "SAVE WITH ATTRIBUTES? (Y/N) "; LINE a$: IF a$="Y" THEN LET col=1: GO TO 120
110 IF a$="N" OR a$="n" THEN LET col=0: GO TO 120
115 GO TO 100
120 IF col AND ldir THEN LET col=255
125 POKE a,col: INPUT "NAME OF COMPRESSED FILE? "; LINE n$: IF col THEN SAVE n$CODE a-4,vse+5
130 IF NOT col THEN SAVE n$CODE a-4,sit+5
140 PRINT #1;"PRESS A KEY FOR VERIFICATION": PAUSE 0: INPUT "": VERIFY n$CODE : GO TO 250
200 LET b=65143: GO SUB 400
250 INPUT "PRINT/LOAD/QUIT/SCREEN/INFO?" (P/L/Q/S/I)", LINE n$: IF CODE n$=0 THEN GO TO 250
260 IF n$="L" THEN GO TO 15
270 IF n$="S" THEN GO TO 310
280 IF n$="P" THEN GO TO 500
290 IF n$="I" THEN CLS *: GO TO 25
300 STOP
310 IF NOT col THEN GO TO 200
320 INPUT "DOTS/ATTR/TOGETHER? (D/A/T)"; LINE n$: IF n$="T" THEN GO TO 200
330 IF n$="D" THEN LET z=PEEK a: POKE a,0: GO SUB 400: POKE a,z: GO TO 250
340 LET b=65138: LET g=1: GO SUB 400: LET b=65143: GO TO 250
400 CLS *
405 RANDOMIZE USR b: FOR f=1 TO 50: NEXT f: IF INKEY$<>" THEN GO TO 420
407 IF g THEN LET g$="00000000000000000000000000000000": POKE 23659,0: FOR f=1 TO 24: PRINT g$
: NEXT f: POKE 23659,2: LET g=0
410 PRINT AT 20,10; BRIGHT 1;"PRESS A KEY": FOR f=1 TO 50: NEXT f
413 IF n$="A" THEN PRINT AT 20,10; BRIGHT 0;"0000000000": IF INKEY$="" THEN GO TO 405
415 PRINT AT 20,10; BRIGHT 0;"": IF INKEY$="" THEN GO TO 405
420 FOR f=1 TO 22: PRINT #1;"": NEXT f: CLS *: RETURN
500 REM priradit a odtud volattiskovou rutinu (screen dump), resp.zaradit prikaz COPY
600 LET sit=PEEK (a-4)+256*PEEK (a-3)-a: LET vse=PEEK (a-2)+256*PEEK (a-1)-a: LET atr=vse-sit:
RETURN
9000 LOAD "b:bytescomp": RUN
9998 OPEN #4,"c": PRINT #4;CHR$ 27;CHR$ 51;CHR$ 27: PRINT #4;CHR$ 27;CHR$ 77;: PRINT #4;CHR$ 27
;CHR$ 71;: LIST #4: CLOSE #4: STOP
9999 SAVE #4:"b:bako" LINE 9000: VERIFY #4:"b:bako": SAVE #4:"b:bytescomp",65000,230: VERIFY #4:"b:bytes
comp"

```



```

1 REM Vypis pomocneho Basicu pro komprimaci a expanzi souboru obrazovek
2 REM
10 LET n=1: POKE 23658,8
15 CLS *: INPUT "LOAD SCREEN$ OR DATA FILE? (S/D)""', LINE a$: IF a$="D" THEN GO TO 500
17 LET n$=STR$ n: LET suma=0: GO SUB 1099: INPUT "NUMBER OF FILES?";num
20 LET suma=suma+1: LOAD ""SCREEN$ : RANDOMIZE USR 65000: GO SUB 599
25 POKE a,0: REM GO SUB 400
30 PRINT AT 1,4; BRIGHT 1;" RESULT OF COMPRESSION:"; LET b$=" bytes "
40 LET kosi=6144-sit
50 PRINT BRIGHT 1'"" DOTS are "; INVERSE 1;sit; INVERSE 0;b$' " (";kosi;b$;"shorter)
"
70 LET a=a+sit+5: LET v=64700-a: PRINT AT 15,8;"FREE MEMORY=";v;AT 18,5;"NUMBER OF DATA FILES=
";suma
71 IF num=suma THEN GO TO 100
73 REM PRINT #1;"ENTER-NEXT LOAD OTHER-SAVE IT": PAUSE 0: IF CODE INKEY$<>13 THEN GO TO 1
00
75 IF v<4000 THEN INPUT FLASH 1; BRIGHT 1;"WARNING! CONTINUE TO LOAD? (Y/N)"; FLASH 0; BRIGH
T 0', LINE i$: IF i$="N" THEN GO TO 100
80 GO SUB 1000: GO TO 20
100 SAVE #4:"Z80pictrs"+n$,32768,a+6-32768
140 PRINT #1;"PRESS A KEY FOR VERIFICATION": PAUSE 0: INPUT "": VERIFY #4:"Z80PICTRS"+n$: IF n=3
THEN STOP
150 LET n=n+1: GO TO 17
400 CLS *
405 RANDOMIZE USR b: FOR f=1 TO 50: NEXT f: IF INKEY$<>" THEN GO TO 420
410 PRINT AT 20,10; BRIGHT 1;"PRESS A KEY": FOR f=1 TO 50: NEXT f
415 PRINT AT 20,10; BRIGHT 0;"": IF INKEY$="" THEN GO TO 405
420 CLS *: RETURN
499 REM load and print
500 GO SUB 1099: LOAD #4:"Z80pictrs1"
555 CLS *: RANDOMIZE USR b: RANDOMIZE USR pr1: GO SUB 599: LET a=a+sit+5: GO SUB 700
560 FOR n=1 TO 2: FOR f=1 TO 4: GO SUB 600: NEXT f: GO SUB 700: NEXT n: GO SUB 1099: LOAD #4:"Z80
pictrs2"
570 FOR n=1 TO 2: FOR f=1 TO 4: GO SUB 600: NEXT f: GO SUB 700: NEXT n: GO SUB 1099: LOAD #4:"Z80
pictrs3"
580 FOR f=1 TO 4: GO SUB 600: NEXT f: STOP
599 LET sit=(PEEK (a-4)+256*PEEK (a-3))-a: RETURN
600 CLS *: GO SUB 599: LET x=b+1: LET y=a: GO SUB 1010: RANDOMIZE USR b: RANDOMIZE USR pr2: LET
a=a+sit+5: RETURN
700 CLS *: PRINT #1;"PUT IN NEW PAPER & PRESS A KEY!": BEEP 1,.05: PAUSE 0: RETURN
1000 LET x=adrkomp: LET y=a+1: GO SUB 1010: LET x=len: LET y=a-4: GO SUB 1010: LET x=b+1: LET y=
a
1010 LET hi=INT (Y/256): LET lo=Y-256*hi: POKE x,lo: POKE x+1,hi: RETURN
1099 POKE 23658,8: LET a=32772: LET w=a: LET ADRKOMP=65019: LET LEN=65086: LET b=65143: LET pr1=
64840: LET pr2=pr1+3
1100 POKE ADRKOMP,5: POKE ADRKOMP+1,128: POKE b+1,4: POKE b+2,128: POKE len,0: POKE len+1,128: R
ETURN
8888 OPEN #4,"c": PRINT #4;CHR$ 27;CHR$ 51;CHR$ 27;: PRINT #4;CHR$ 27;CHR$ 77;: PRINT #4;CHR$ 27
;CHR$ 71;: LIST #4: CLOSE #4: STOP
9000 LOAD #4:"byteskomp": RUN
9999 SAVE #4:"Z80komp" LINE 9000: VERIFY #4:"Z80KOMP": SAVE #4:"byteskomp",64796,435: VERIFY #4:"bytes
komp"

```



# SPECTRUM

## Integrovaný wordprocessor pro ZX Spectrum

3.část

Wordprocessor sestává ze 4 obvodů. IO4 je analogově-digitální převodník (ADC) na bázi CMOS. Jeho spotřeba je pouhých 10 mW. IO1 může být jeden z novějších obvodů řady 74HC CMOS s vysokou rychlostí přepínání - 74HCT138 - nebo starší 74LS138. IO2 - paměť EPROM - stráví většinu času "mimo provoz". Tak nejvíce energie spotřebuje IO3 - Z80APIO. Přestože je možné napájet celé zapojení prostým připojením k napájení Spectra, rozhodně doporučuji zvláštní napájecí linku 5 V. Vyhnete se tak možným nepříjemnostem, včetně kolapsů programu, které občas vzniknou, je-li na stejnou napájecí linku současně připojeno i analogové zařízení (v našem případě ADC).

### Logika

V uvedeném zapojení není nic převratného. IO1 dekóduje vrchní 3 adresové linky a rozděljuje paměťovou mapu mikro-

procesoru Z80 na 8 dílů po 8 K. Q1 je druhý výstup dekodéru - je ve stavu log. 0, kdykoli jde adresa do rozsahu 2000H-3FFFH. Logická nula aktivuje stínovou EPROMku a je tranzistorem T1 invertována na log.1, která na výrobu ROM-CS (ROM chip select) odpojí ROMku Spectra. IO1 nereaguje, dokud na jeho vývodu 6 nebude log.1. Normálně je na něm log.0 (vlivem R7). Pokud IO1 řídíte obvodem Z80APIO (jak je tomu i v uvedeném zapojení), můžete R7 klidně vynechat - linka PA2 je na úrovni log.0 vždy po zapnutí přívodu proudu a zůstává na ní, dokud nedostane pokyn ke změně od softwaru. Linky PA2, PA1 a všechny PB jsou výstupní. PA0 a PA3 až PA7 jsou vstupní. PIO je aktivován log.0 na A5. Jeho čtyři vnitřní registry jsou určeny hodnotami na A8 a A9 - tak vytvářejí tyto adresové kombinace:





9 8 7 6 5 4 3 2 1

---

0 0 1 1 0 1 1 1 1 FCDF 65735 PADATA  
0 1 1 1 0 1 1 1 1 FDDF 64991 PACONTROL  
1 0 1 1 0 1 1 1 1 FEDF 65247 PBDATA  
1 1 1 1 0 1 1 1 1 FFDF 65503 PBCONTROL

---

I04 je aktivován logickým součinem signálů na A6 a IORQ v konfiguraci diod D2 a D3. Log.0 na A6 dává adresu 01111111 tedy FFBF či 65471. V naší aplikaci jsou Vref a AGND (analogová zem) I01 jednoduše připojeny přímo na zdroj. Můžete si však referenční napětí zvolit a upravit, jak budete potřebovat. Signál "konverze hotova" -INSTR - jde na PA7 PIO, odkud jej můžete programově odebrat, budete-li mít pro to nějaké další využití.

### Konstrukce

Jak vyšlo z dalších pokračování najevo, časopis Electronics and Computing připravil pro zájemce o stavbu zařízení desky s plošnými spoji i hotové EPROMky s programovým vybavením. Obojí tedy v našem překladu bude z větší části chybět. Proto náš seriál pojímejte jako nástin možností podobné vlastní konstrukce.

### Software

Rozdělení paměti ukazuje (zhruba) následující tabulka:

---

Adresa:	Obsah:
2000-2BFF	základní program - začátek EPROM
2000-2FFF	převodní tabulka DEFB
3000-33FF	rutiny prohledávání
3400-3BFF	obsluha tiskárny
3000-3FFF	rozšířený soubor znaků - konec EPROM
8000-EFFF	26 K textu (cca 4000 slov)
(EA00-EFFF)	možný přídatný soubor znaků pro tiskárnu
F000-FFFF	pracovní prostor

---



V paměti RAM je pouze krátký program v BASICu pro ovládání záznamového zařízení. Do této paměti si samozřejmě můžete libovolně doplňovat jakýkoli software pro vaše osobní potřeby i experimenty.

Výpis celého programu není pro svou délku myslitelný, stejně jako komentář ke všemu, co se v něm děje. Probereme si jen některé jeho hlavní části. Jejich vzájemnou kombinací a některými doplňky byste mohli dosáhnout vytvoření softwaru pod 1 K. Samozřejmě, že na takovém slovním procesoru by se nedaly hned psát romány. Ale ke vstupu do této oblasti aplikace počítače to bude dostatečně ilustrativní.

3 K základního programu obsahují rychlé vertikální scrollování, insert, stanovení délky řádky, zarovnávání řádky (justification), přenos volitelného počtu řádek (block copy), mazání znaků i řádek, hledání a výměna slov, tisk celého nebo části textu. Pokud vám ve výčtu něco chybí, musíte si uvědomit, že nejde o wordprocessor typu WordStar. Jeho síla je v některých jeho specialitách - ovládání kursoru pomocí ADC interfacu, konverze hexadecimální na decimální, prohledávání podle voleného pole, manipulace s grafikou tiskárny Epson RX80, zvětšování textu na jakékoli maticové tiskárně apod.

Struktura programového vybavení je na obr. 2. Do programu se vstupuje jedním ze tří vstupů: Machine-cold-start, User-cold-start a Warm-start. Po vstupu do Basicu se můžete vrátit zpět do vstupu Warm-start příkazem RUN. Tento vstup vás přivede vždy na pozici kurzoru v textu. User-cold-start vymazává celý dosavadní text, ale také vám dovolí úpravy některých bajtů (např. pro řízení tisku). Machine-cold-start je prostou inicializací celého programu tak, jak byl původně sestaven (případný text

se vymaže rovněž).

Povelová rutina zjišťuje, jaké tlačítko je stisknuto. Podle toho buď vypíše znak na obrazovce nebo vstoupí do jedné ze 26 řídicích rutin. Pokud vám tento počet nebude stačit, můžete přidat další (umístit je však musíte do RAM). Všechny rutiny ROMky jsou vektorově adresovány z RAMky. Změnou adresy vektoru můžete převést řízení na vaše vlastní rutiny v RAM. Tak lze upravit třeba počet znaků na řádku, který je normálně 32.

Přestože obrazovka Spectra je jednou z nejlepších u tohoto druhu malých mikropočítačů, je tu problém s časem, po který trvá zpracování každé nové informace směřované do obrazové paměti. Spectrum pracuje i se znaky v grafickém módu - oproti BBC, Acornu, Memotechu a dalším, které jsou přepínatelné z módu znakového na grafický a naopak - ty pracují se znaky jako s bloky (většinou jich je 24 x 40) a nepropočítávají stále pozici každého bodu zvlášť. Spectrum při každé změně musí provést refresh 4 K.

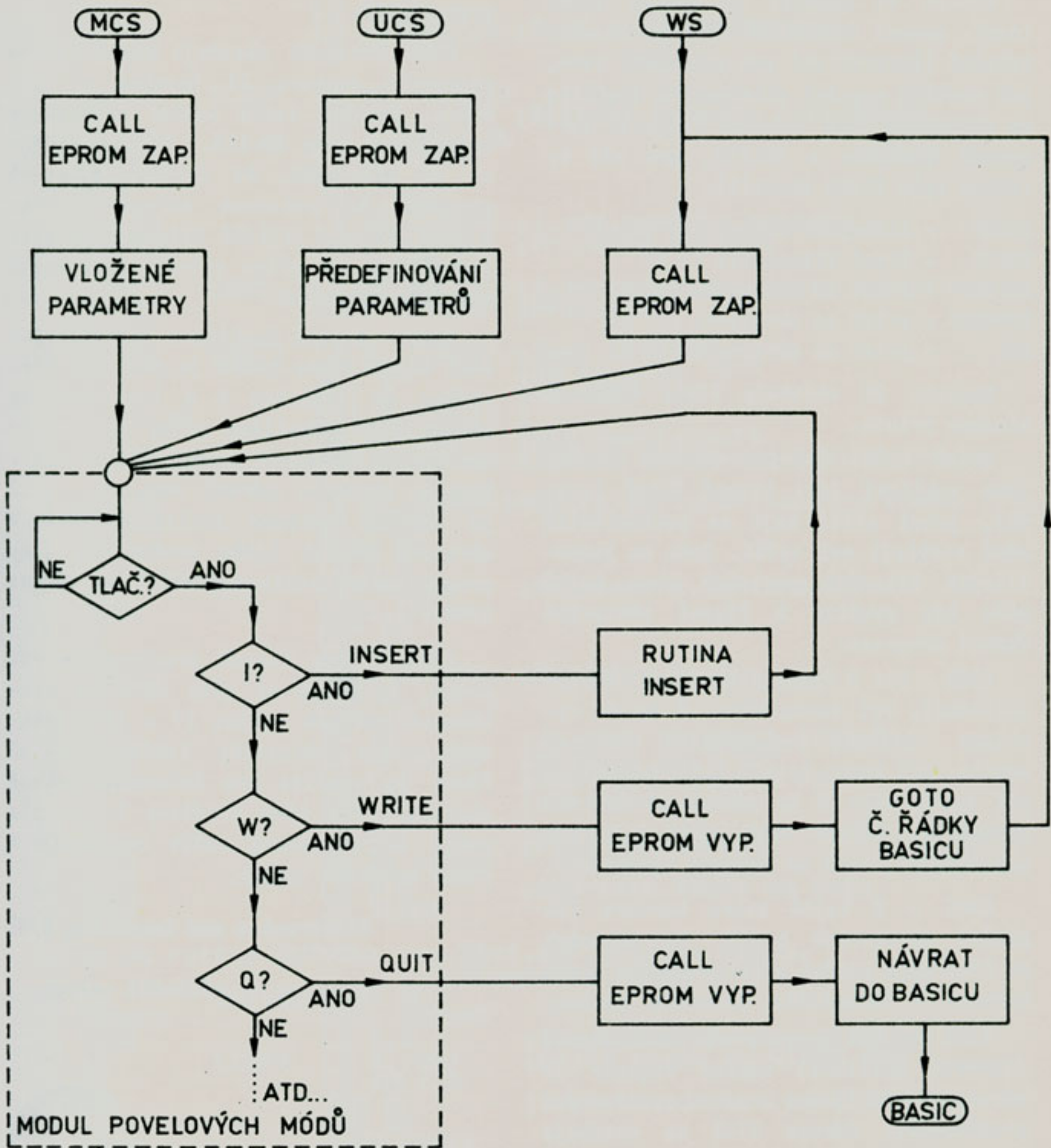
Je lépe se vyhnout i printovým rutinám Spectra (start od adresy 10H), které kontrolují spoustu funkcí, které nám nejsou k ničemu. Náš program ovládá obrazovou paměť přímo. Příklad toho, jak to dělá, je uveden ve Výpisu 1.

Na rozdíl od tisku na obrazovku, je text klávesnice velmi rychlý - je proto využit i v našem programu. Test probíhá každou 1/50 vteřiny. Je řízen módem přerušení IM 1, který převede program do rutiny testu klávesnice od adresy 02BF ROM Spectra. Posledně stisknuté tlačítko změní obsah několika systémových proměnných.











## Výpis 1

Rutina se spouští od adresy 50000 (např. pomocí RANDOMIZE 50000). Je dostatečně ilustrativní pro pochopení toho, jak probíhá vypsání stisknutého tlačítka na obrazovce (zde nebudou fungovat kurzorová tlačítka a DELETE, pro něž by rutinu bylo třeba doplnit).

TLOOP - blikající kurzor (28 bajtů)  
CONT2 - čtení klávesnice (3 B)  
OUTVDU - print na obrazovku (14 B)  
CONVERT - print na obrazovku (17 B)  
CHL - print na obrazovku (13 B)

Registr BC obsahuje číslo řádky a sloupce printu.

Obsah COUNT (adresy C3E1-2) slouží jako cyklický 16ti bitový čítač (reg. HL), jehož nižší bajt se pochopitelně mění rychleji než vyšší. Změn na jednotlivých bitových pozicích obou bajtů může být využito k řízení některých funkcí programu:

C362 - bit 7 "zapíná a vypíná" kurzor (blikání). Při testu bitu 6 by se rychlost blikání pochopitelně zdvojnásobila, při testu bitu 5 zečtyřnásobila, při testu bitu 8 by se naopak dvakrát zpomalila. Můžete však volit i kurzor neblikající, čímž se program zjednoduší.

C3D1 - 8 bajtů tvaru kurzoru (64 plných obrazových bodů). Na adresách od C3CD máte uveden příklad jiného tvaru kurzoru - úhlopříčka printové pozice.

C37A - ROM Spectra po inicializaci (zapnutí počítače) naplní registr IY adresou 23610 (jedna z adres systémových proměnných). Adresa IY=1 se vztahuje k proměnné FLAGS, jejíž bit 5 signalizuje, zda bylo stisknuto nějaké tlačítko - v tom případě má hodnotu log.1 (a naopak).

C380 - po stisku tlačítka musíme bit 5 proměnné FLAGS překlopit zpět

do log.0, jinak by pokračoval tisk stále téhož písmene na obrazovce ("ilegální opakování").

C384 - reg.HL obsahuje první adresu znakového souboru ZX Spectra v jeho ROMce. Bohužel tento soubor neobsahuje znaky pro kódy 00-1FH. Stínová ROMka už je mít bude. Na dalších adresách programu je nalezena adresa prvního z osmi bajtů, jejichž bitová kombinace jedniček a nul přímo vytváří tvar znaku pro jeho zobrazení.

C3AC - bitová kombinace jednotlivých osmi bajtů tvaru znaku je postupně převáděna do registru A a instrukcí na adrese C3AD odesílána přímo na patřičnou pozici obrazové paměti (VIDEO RAM), kterou obsahuje párový reg. HL, jenž je zároveň adresou pozice kurzoru v obrazové paměti (jeho souřadnice jsou propočítávány rutinou CONVERT na adrese C3B8).

C3B2 - tato časová prodleva je zde proto, že během ní může nerušeně proběhnout refresh cyklus 17 x 32 printových pozic obrazovky (zbývajících 7 řádek je využito pro informace určené obsluze). Bez této prodlevy by se nám mohly na obrazovce začít objevovat i různé nevídané znaky.

Elektronics and Computing 6/85

přeložil -elzet-





ORG 50000

```

C350 3E01          LD A,1
C352 320A5C       LD /23562/,A ; Syst.prom.REPPER
C355 010000       LD BC,0

C358 2AE1C3       TLOOP LD HL,/COUNT/
C35B 23           INC HL ;Blikání kurzoru /viz text/
C35C 22E1C3       LD /COUNT/,HL
C35F 21E1C3       LD HL,COUNT
C362 CB7E         BIT 7,/HL/
C364 2808         JR Z,CONT
C366 11D1C3       LD DE,CURSHP ;1.adr.grafiky kurzoru
C369 CDA6C3       CALL OUTVDU ;Projekce kurzoru
C36C 1806         JR CONT2
C36E 11D9C3       CONT LD DE,NOCUR ;1.adr.grafiky výmazu kurzoru
C371 CDA6C3       CALL OUTVDU ;Projekce

C374 3A085C       CONT2 LD A,/23560/ ; Test tlačítka
C377 FE30         CP "0" ; Je to znak "0"?
C379 C8           RET Z ; Návrat
C37A FDCB016E     BIT 5,/IY+1/ ;
C37E 28DB         JR Z,TLOOP ; Nic nestisknuto
C380 FDCB01AE     RES 5,/IY+1/ ; Zpětné nastavení bitu 5 /=0/

C384 21003C       LD HL,3D00H-100H ;1.adr.graf.souboru znaků
C387 110800       LD DE,8
C38A C5           PUSH BC
C38B 47           LD B,A
C38C 19           CHL ADD HL,DE ; Nalezení 1.adr.znaku
C38D 10FD         DJNZ,CHL
C38F C1           POP BC
C390 EB           EX DE,HL
C391 CDA6C3       CALL OUTVDU ; Projekce znaku
C394 0C           INC C
C395 79           LD A,C
C396 FE20         CP 32 ; Je znak na konci řádky?
C398 20BE         JR NZ,TLOOP ; NE-skok na TLOOP
C39A 0E00         LD C,0 ; AND-inic.sloupce na 0
C39C 04           INC B ; a zvýšení řádky o 1
C39D 78           LD A,B
C39E FE18         CP 24 ; Plná obrazovka?
C3A0 20B6         JR NZ,TLOOP
C3A2 0600         LD B,0 ; AND-inic.řádky na nulu
C3A4 18B2         JR TLOOP

; Vstup do rutiny OUTVDU: reg.DE=1.adr.grafiky znaku,
; BC=pozice kurzoru
C3A6 C5           OUTVDU PUSH BC
C3A7 CDBBC3       CALL CONVRT
C3AA 0608         LD B,8 ; 8 cyklů uložení grafických
C3AC 1A           LOOP4 LD A,/DE/ ; bajtů znaku do VIDEO RAM
C3AD 77           LD /HL/,A
C3AE 24           INC H ; Zvýšení adr.obr.paměti o 256
C3AF 13           INC DE
C3B0 10FA         DJNZ LOOP4
C3B2 0600         LD B,0
C3B4 10FE         D1 DJNZ D1 ; Zařazení prodlevy
C3B6 C1           POP BC
C3B7 C9           RET

C3B8 C5           CONVRT PUSH BC ; Výpočet pozice kurzoru na
C3B9 78           LD A,B ; obrazovce
C3BA E618         AND 18H

```



```

C3BC C640          ADD A,40H
C3BE 67           LD H,A      ;H=vyšší část adr.obr.paměti
C3BF 78           LD A,B
C3C0 E607        AND 7
C3C2 0F          RRCA
C3C3 0F          RRCA
C3C4 0F          RRCA
C3C5 B1          ADD A,C
C3C6 6F          LD L,A      ;L=nižší část adr.obr.paměti
C3C7 C1          POP BC
C3C8 C9          RET

C3C9 0102040B
C3CD 102040B0    UDB      DB 1,2,4,8,16,32,64,128

C3D1 FFFFFFFF
C3D5 FFFFFFFF    CURSHP  DB 255,255,255,255,255,255,255,255

C3D9 00000000
C3DD 00000000    NOCUR   DB 0,0,0,0,0,0,0,0

C3E1 00          COUNT  DB 0
C3E2 00          CHIGH  DB 0

```

;Rutina se spouští od adresy 50000.

# Kterak program opřísti zatajenými tajemnými

I když název článku jakoby vypadl z černokněžníkovy orákula, bude se zabývat velmi střízlivou programovou logikou, kterou se mnohé softwarové firmy i autoři programů sami snaží skrýt obsah svých programů před očima nepovolanými. Ještě donedávna si vymýšleli různé finesy, které měly zabránit kopírování programů. Dnes však existuje už tolik spolehlivě pracujících kopírovacích metod, že nakonec přistoupili k vymyšlení a prosazení velmi nepopulárního, tvrdého

zákona na jejich ochranu.

V následujícím si povíme o několika zatajovacích finesách, které, stejně jako to čeká i ty budoucí, byly už dávno odhaleny. Vztahují se k ZX Spectru, ale jejich princip lze převést prakticky na jakýkoli počítač. Jedná se spíš o zajímavosti pro začínající programátory; ti vyspělejší většinou nakonec dospějí k názoru, že utajování není k ničemu.



## Barva inku a papíru

Jedna z nejstarších záhad - dáte-li příkaz k projekci inku i papíru ve stejných barvách, nepřečte si na obrazovce nikdo nic. Poněkud fikanější je využití tohoto typu utajení pro "zmizení" třeba jen jednoho, velmi důležitého příkazu na konci Basicové řádky. Provádí se tak, že řádku vyvoláme do editační zóny, kurzorem najedeme před příkaz, určený ke zmizení, přejdeme do extended módu (CAPS a SYMBOL SHIFT), podržíme tlačítko CAPS SHIFT a stiskneme 7 (sedmičku) -

musíme však mít ink černý a papír bílý. Příkaz z obrazovky (nikoli z programu) zmizí. Co se stalo? Vložili jsme do řádky jen řídicí kód s parametrem pro print inku v barvě bílé. Po "odpálení" řádky zpět na její místo v programu však uhlídáme, že nám zmizelo i vše za ní. Tomu zamezíme tak, že za příkaz, který má zůstat neviděn, stejným způsobem vložíme ink 0, tedy černý. Abychom se ve zmizelých částech řádek v editační zóně vyznali, změňme barvu borderu - pak se řádka vybarví vždy v kontrastní barvě, včetně ukrytého příkazu.

Podobně můžeme měnit i barvu papíru Basicových řádek. Pro ni stačí přejít jen do extended módu a stisknout pětice číslo barvy. Budete-li chtít přebarvení odstranit, nemusíte je měnit způsobem, jakým jste je vyvolali. Stačí jen najet kurzorem do začátku přebarvení a mazat a mazat (DELETE), až přemažete všechny objevující se otazníky a barvy se dostanou zase do svých původních mezí. Jistě sami přijdete na to, jak docílit aktivity FLASH a BRIGHT. Takováto výtvarná činnost na řádkách programu však nemusí být zaměřena jen na ututlání čehokoli, ale má svůj značný efekt pro zvýraznění důležitých částí programu, což vede k podstatnému zlepšení orientace v jeho struktuře.

Složitěji můžete postoupit tak, že např. na Basicové řádce 10 NEXT a vynecháte mezi jejím číslem a příkazem 4 printová místa: 10 NEXT a

Do těchto mezer pak vložíte řídicí kódy pro INK a PAPER spolu s jejich parametry. Je-li to řádka první, budou adresy mít hodnotu 23759 až 23762 (pokud nemáte připojen nějaký drive, který posunuje začátek Basicového programu). Zavelíme:

```
POKE 23759,16:POKE 23770,1:POKE 23771,17:POKE 23772,1
```

Celá obrazovka se naplní barvou modrou, výpis programu nikde. Vysvětlení - 16 je řídicí kód pro INK, 17 pro PAPER, 1 je kód modré barvy. Míst můžete vynechat víc a vložit kódy a parametry i pro FLASH, BRIGHT, OVER a INVERSE.

Pro programové odborníky není problémem vymyslet žertíky, kdy pomocí strojového kódu bude v módu přerušování IM 2 měnit barvu INK, PAPER a třeba i BORDER současně 50krát za vteřinu. Bolavé oči odradí každého od jakéhokoli pokusu o odtajení programu.

## Stringy, neboli řetězce a jiné proměnné

Všichni, kdož jste už měli něco společného s Basicem, dobře víte, že zadávat příkazy počítači můžete dvěma způsoby - buď prostřednictvím programu (příkaz je na programové řádce), nebo přímo (z editační zóny). V tom druhém případě - po stisku ENTERu příkaz navždy zmizí, nikoli však parametr příkazu. Jde-li o zadání hodnoty proměnné, např. LET W=111, číslo 111 i s názvem proměnné W se usadí v oblasti Basicových proměnných. Takto můžeme skrytě zadat celou řadu proměnných, u nichž se v průběhu



programu buď hodnota nemění, nebo slouží jen jako výchozí pro spuštění programu. Musíme se ovšem vyhnout použití příkazu CLEAR, který vymazává celou oblast Basicových proměnných. Odtajit takový program lze pomocným programem pro výpis hodnot všech proměnných před spuštěním vlastního programu.

### Neviditelná řádka

Na (většinou první) řádku lze za příkaz REM uložit program ve strojovém kódu. Pokud takovou řádku napíšeme nejprve samotnou a pak zavelíme POKE 23755, 255, řádka zcela zmizí. Přidělili jsme jí totiž číslo 255\*256, které je vyšší než max. přípustné 9999. Rutina v ROMce takovou řádku pak nevypíše, avšak strojový kód zůstane na svém místě (od adresy 23778). Pokud by na takové řádce byl Basic, nemohli byste jej spustit. Po této operaci můžeme pokračovat v normálním zápisu Basicového programu, který bude "spustitelný". Odhalit tuto fintu lze "poukováním" na l.adresu Basicu (změna čísla l.řádky). S číslováním řádek je možno si všelijak hrát - a některé efekty jsou velmi zajímavé. Zkuste sami.

### Zablokování spodních řádek monitoru

Systémová proměnná DFSZ na adrese 23659 má inicializovanou hodnotu 2. Je to počet řádek pro jejich výpis v editační zóně. Pokud obsah DFSZ změním na 0, budou obě spodní řádky pro editaci zablokovány a Basicový program bude pracovat na celé ploše 24 řádek! Zároveň ovšem toto zablokování povede k tomu, že kdykoli bude mít nastat moment výpisu

nějakého hlášení na spodní řádce obrazovky, počítač se zachová tak, že začerní celou obrazovku a dál se s ním na ničem nedomluvíte. Pak už lze jen vypnout přívod proudu. Zkuste si tento krátký program:

```
10 POKE 23659,0
20 FOR f=1 TO 24:PRINT f:NEXT f
30 PAUSE 0
```

Stisknete-li tlačítko, nastane uvedený kolaps, protože hlášení (v tomto případě OK) se nemá kam vypsat. Ven z takového programu se dostane jedině tehdy, pokud do něj někde vložíte POKE 23659,2. Odtahujeme-li takový (i jakýkoli jiný) program, vždy místo příkazu LOAD použijeme MERGE, abychom se vyhnuli automatickému spuštění programu. Někdy dokonce nezbude, než si půjčit hlavičku od jiného programu, resp. vytvořit vlastní.

### ERR SP

Tato systémová proměnná na adresách 23613 a 23614 obsahuje vyšší a nižší bajt vektorové adresy v zásobníku (machine stack) pro skok do rutiny chybových hlášení (viz článek ON ERROR GO TO). Pokud ji vynulujeme, program skočí na adresu 0 (to je totéž, co RANDOMIZE USR 0) a obsah paměti navždy zmizí. Protože adresa ERR SP je stále kontrolována, musíme její obsah často měnit - po všech příkazech GO TO, GO SUB, v každé smyčce a ve všech dlouhých PRINTech. Protože je to přece jen moc práce, je tato varianta uvedena spíš pro její zajímavost. Ovšem změnou tohoto vektoru můžete převést řízení programu i někam jinam (např. v ROMce) a dosáhnout nečekaných obrátů, které nemusí nutně vést ke kolapsu programu.



## Basic v převlečku za "stroják"

Jedním z velmi často užívaných programátorských "chvatů", kterým se nejen utají program, ale jenž má i svůj praktický smysl, je záznam programu v BASICu (většinou spolu s obsahem printer bufferu, obrazovky, systémových proměnných i programem ve strojovém kódu umístěným hned za Basicem) - příkazem:

```
SAVE " " CODE, adresa, délka
```

Specialitou v tomto případě je, že adresa má hodnotu mezi 1. bajtem printer bufferu a posledním bajtem systémových proměnných. Jeho délka může zahrnout i poslední bajt případně přidaného strojového kódu. Striktně řečeno - můžeme tak nahrát celou paměť počítače od adresy 0 do 65535 bez ohledu na to, co v ní je zakódováno. Načtení takového programu do paměti je nutno provést ekvivalentně: LOAD ""CODE. Jde o věc trochu komplikovanější, kterou se pokusím vysvětlit pomocí praktické ukázky.

Za prvé - noticka směrem k novým adeptům computerových múz - paměť počítače je adresována postupně od 0 až do 65535 (je-li využito celého rozsahu adresování osmibitového mikroprocesoru - tedy máme-li v počítači paměťové obvody s celkovou kapacitou 64 K). Ať už jde o cokoli (obrazovku nebo Basic atd.), na jedné adrese je vždy jeden bajt. Záleží jen na tom, jak k nim programově přistupujeme (zda je interpretujeme jako Basic, nebo jako instrukce strojového kódu, či je bereme jen jako parametry, data, tedy definované bajty). Prostě - co jedna adresa, to "nějaký" jeden bajt.

Z toho - za druhé - vyplývá, že je vlastně jedno, jak bajty nahrajeme na pásek nebo disk apod. Jde však o to, abychom po jejich načtení sdělili počítači, jak které z nich interpretovat

(dekódovat, zpracovávat). Z toho vychází i následující postup, který ukazuje, jak v takovém bloku bajtů objevit skrytý Basic.

Vložme do počítače program uvedený v části pojednávající o zablokování spodních obrazových řádek a zapišme jej na pásek příkazem SAVE "blok" CODE 23500 500 (délka je zvolena od oka tak, aby se bajty našeho programu v BASICu, rozprostírajícího se od adresy 23755, určitě na pásek dostaly; 1. adresa zápisu může být kdekoli pod adresou 23755 - nemusí to být zrovna 23500).

Pokud to, co máte na pásku, načtete pomocí LOAD "" CODE, program se automaticky spustí od 1. řádky Basicu (a samozřejmě se při pokusu do něj vstoupit zablokuje). Profesionální programy využívají tohoto automatického spuštění Basicové části k následnému okamžitému spuštění části programu ve strojovém kódu - většinou tak, že je v Basicu příkaz RANDOMIZE USR xxxx, kde xxxx je startovní adresa části programu ve strojovém kódu.

Pro odhalení zašifrovaného Basicu musíme znát adresu, od níž se celý blok bajtů načítá (tu vám vyjeví jakýkoli hlavičkář, někdy je to však trochu detektivka). Vraťme se k našemu programu. Víme, že začíná od adresy 23500. Teď ořízneme bajty adres 23500-23754, abychom "vylohuovali" Basic, začínající na adrese 23755 (opakuj, že některá přídatná zařízení mohou tuto adresu posunout někam výš - pak je třeba zjistit, kde oblast Basicu začíná; jinak je postup shodný).

Z pásku načteme bajty programu od jiné adresy - třeba 3000 - повеlem:

```
LOAD "" CODE 3000
```

Tak se nám Basic nemůže spustit. Teď odečteme 23755-23500 a dostaneme počet "zbytečných" spodních bajtů -



je jich 255. Ty musíme dále vynechat. Proto nahrajeme na pásek jen bajty od adresy 30255 povelom:

```
SAVE"blok A"CODE 30255,246
```

Před načtením bloku A do paměti musíme vykonat některé přípravné práce. Spočívají v tom, že vložíme do počítače řádku 1 a na ni za příkaz REM vložíme 246 (nebo více) libovolných "nesmyslných" znaků (jakákoli písmena nebo číslice):

```
1 REM 111ERTyuiH674MMM (...atd.,  
dokud jich nebude aspoň 246)
```

Protože je to práce značně zdlouhavá, byly vytvořeny pomocné programy, které ji zjednoduší. Připravíme si magnetofon a zadáme povel:

```
LOAD "" CODE 23755
```

...a Basic se objeví v celé své kráse, aniž by se spustil (uvedený postup je ekvivalentem Basicového povelu MERGE). Někdy stačí takovýto typ bloku během načítání do paměti prostě "brejknout" a Basic se objeví taky.

## ZÁVĚR

Uvedené příkazy, kterými ututláváme obsah programu, jsou všechny vkládány z Basicu do Basicu. A jak známo, s ním se nedá udělat zdaleka tolik, co se

strojovým kódem. Profesionální firmy pracují na utajení a nekopírovatelnosti svých programů různým způsobem. Jednou z hezkých fint je např. změna průběhu zaváděcího (leader) signálu pro části programu před jejich LOADem. Podobně lze prudce změnit rychlost přenosu pár bajtů během načítání z pásky, nebo vložit mezi bajty programu pár "nesmyslných" pulsů apod. Algoritmů pro záznam a čtení bajtů lze vytvořit nepřeberné množství. Běžné kopírovací programy si pak s nimi vůbec neporadí. Jednou z dalších fint je přepisování čtecí rutiny vlastním programem při jeho LOADu. Mást "protivníka" lze také vytvářením speciálních hlaviček. Bezhlavičky by vydaly na jednu zvláštní kapitolu.

I pro Basic existují další způsoby zatajování. Jedna hezká "schovávačka" při výpisu programu ukáže, že všechny proměnné se rovnají nule (přestože to samozřejmě není pravda).

Hrátek tohoto druhu je dlouhá řada. Základem však zůstává, že neexistuje program, který by nešel odtajit. Každý totiž musí nějak začít normálním ukládáním bajtů do paměti. Jde jen o to je odchytit a dál už jen sledovat stopu. Leckdy je to velmi náročné a vlastně jde o to zamyslet se nad tím, zda taková činnost má vůbec nějaký větší smysl. I když, jak se říká, proti gustu...

-elzet-



# Úpravy her pro ZX Spectrum

Tento příspěvek je malou exkurzí do světa mikropočítačových her. Měl by umožnit všem, kteří ať už nemohou nebo nechtějí věnovat celý svůj volný čas zábavě, okusit slastný pocit vítězství ve hrách, v nichž jsou panáčci absolutně nesmrtelní ("neničí se životy"), resp. jsou upraveny tak, aby ulehčily těžký život počítačového dobrodruha. Najdete v něm několik rad, jak upravit některé oblíbené, ale příliš obtížné hry. Někde je připojeno i krátké vysvětlení smyslu programu.

Většina zjednodušení je založena na odstranění nepříjemného a znervózňujícího ubývání životů či energie po každé chybě. V několika případech jsou známé úpravy odstraňující ztrátovou indikaci takových chyb, i řešení nejradikálnější - nesmrtelnost hrdiny. V některých zahraničních časopisech zaměřených na mikropočítače bývá tomuto tématu věnována rozsáhlá plocha - mapy složitých bludišť, podrobné popisy účelu a cíle nových her, hodnocení nebo bodování všech novinek, uvádění fint a rad od "fanatických" hráčů včetně nesčetných pouků nebo i krátkých rutin pro úpravy originálních programů... Dá se říci, že nejméně o stovce her bylo napsáno tolik, že by každé z nich mohlo být věnováno celé číslo útlejšího časopisu.

Nejjednodušší a nejčastěji používanou úpravou je použití příkazu POKE n,x který se provede po nahrání celé hry. Změní se jí obsah jedné nebo několika buněk paměti. Zasvěcenější by ještě mohli požadovat vysvětlení, proč se mění právě daný bajt a proč zrovna tak a ne jinak, ale obecné vysvětlení neexistuje. Vždy je potřeba alespoň zběžná, někdy i hlubší analýza alespoň části strojového kódu programu - a pro obyčejné hráče nemá znalost smyslu úpravy stejně žádný význam.

Novější programy kladou jakémukoli zásahu takřka nepřekonatelný odpor. Vzrostla složitost ochrany a utajení, kterými se softwarové firmy brání neautorskému kopírování. Když se někomu podaří se do programu "nabourat", vyrobí kopírovatelnou verzi, která však laickým uživatelům většinou zůstává naprosto stejně nepřístupná. U špičkových her se málokdy stane, že by se během dalšího nahrávání nepřepsal úvodní basicový program (loader), do kterého se některé níže uvedené příkazy POKE ukládají. Také zastavení programu po jeho nahrání a provedení změn přímo z klávesnice není skoro nikdy možné. Proto jsou stále častější případy, kdy jsou známy příslušné adresy i čísla, která je na ně třeba uložit, ale jen několik nejzasvěcenějších je schop-



no tyto změny vůbec provést. Ostatním pak nezbývá, než čekat, až se objeví nová verze hry již obsahující potřebnou změnu. Jako ideální se jeví ta řešení, která uživatelé sama nabídnou, zda si přeje hrát původní verzi nebo některou z nových modifikací.

Nejjednodušší postup, jak upravit většinu programů staršího data, je následující. Nahrajeme basicový loader pomocí příkazu MERGE"". Tímto způsobem zajistíme, že nedojde ke spuštění basicu a k případnému zásahu do systémových proměnných, který by po stisknutí BREAKu vedl ke zhroucení systému. Pokud nelze basicový program nahrát ani takto, zbývají dvě možnosti. Buď změním v hlavičce bajt označující automatický start programu (např. pomocí HEADEREDITu) nebo lze zkusit jeden z programů zastavujících běh programu ve strojovém kódu - DESIF a SUPERSTOP.

Další postup je pak už pro všechny případy shodný. Nalezneme příkaz spuštění strojového kódu (většinou PRINT USR n, RANDOMIZE USR n apod.) a před něj umístíme všechny pouky. Je-li použit řádek 0, změním jej na jiný pomocí POKE 23756,x, kde x je číslo, na které chceme nulu změnit (při připojené periférii však může být adresa začátku programu jiná). Pak lze i takto chráněný program editovat.

Někdy je v programu umístěn příkaz CLEAR n, který snižuje RANTOP (n je jeho adresa). Je-li toto snížení velké, může se stát, že pro vložení dalších příkazů již nezbude volné místo a počítač napíše chybové hlášení (nebo začne bzuchet). Potom je třeba zkrátit loader tak, že místo čísla 0 se použije výraz NOT PI, místo čísla 1 např. PI/PI, místo čísla 3 se použije INT PI atd. Větší hodnoty, např. 25000 lze zkrátit pomocí funkce VAL na tvar VAL "25000". Důvodem, proč jde o zkrácení, i když zápis je delší, je neekonomický způsob ukládání konstant BASICu Spectra. Nejlehčím zkrácením je vypuštění nedůležitých příkazů jako je nastavení barev, výpis titulků apod.

Tento návod neuvádí všechny druhy utajení "loaderů". Někdy jsou řádky přebarveny, uvnitř bývají změněny hodnoty konstant oproti těm, které se vypíší na obrazovku, do programu jsou vkládány řídicí znaky návratu cursoru, přechodu na nový řádek a další, což úplně znemožní zjistit pravou podobu programu bez přímého výpisu obsahu paměti. Záleží na každém uživateli, jestli na tyto, často triviální, ale o to tajemnější finty přijde a dokáže je odstranit. Když se podaří všemi úskalími s menšími či většími problémy proplout, spustíme loader, kterým nahrajeme zbývající části hry... a je-li vše v pořádku, můžeme se do její upravené verze pustit.

Po nezbytném úvodu, který ti zkušenější četli s úsměvem, budou následovat příklady konkrétních úprav některých her. Zaměřím se nejdříve na ty, které lze získat prostřednictvím Mikrobáze a které představují určitý standard. Jejich další výhodou je, že ve většině případů nepatří k těm nejnovějším a tedy i nejzatajenějším.

## JETPAC

POKE 25020,0 - po provedení tohoto příkazu přestanou ubývat životy. Vložení příkazu do loaderu je bez problémů.

POKE 26075,0 - umožní start rakety nebo raketoplánu pouze s jedním palivem.

POKE 25373,x - úprava pro ty, kterým se předchozí dvě zdají být "unfair". Za x si dosadíte počet životů, s nímž do každé nové hry vstoupíte.



## LUNAR JETMAN

1. první část nahrajte pomocí MERGE ""
  2. proveďte POKE 23756,1
  3. před příkaz PRINT USR 23424 vložte požadovaný POKE
  4. program spusťte a nahrajte ostatní části.
- POKE 36965,3: POKE 36964,224 zajistí, aby neubývaly životy.

## MOON ALERT

POKE 39754,0 - neubývají životy.

POKE 42404,x - hodnota x může být v rozsahu 0-14 a udává počet životů.

POKE 42654,195 - zajistí nesmrtelnost, což znamená, že lunární vozítko je odolné proti všem nárazům, raketám i ostatním nebezpečím. Asi jen s touto úpravou lze hru dokončit.

POKE 42249,24 - zastavení času.

POKE 42585,2: POKE 52596,2 - urychlí scroll obrazovky. Tato úprava je sama o sobě samoučelná, hra je i tak rychlá až dost.

## MANIC MINER

O této hře bylo již napsáno velmi mnoho. Následující dvě úpravy proto představují jen malý výběr ze všech známých změn.

Do jakékoli místnosti se dostanete pomocí tohoto postupu:

Po vstupu do první místnosti stiskněte postupně tlačítka TYPEWRITER, nebo 6 0 3 1 7 6 9. Vyzkoušejte obě možnosti, záleží na verzi hry, kterou zrovna máte. Posloupností číslic dosáhneme úspěchu u starších verzí (údajně se jedná o staré telefonní číslo autora). Pokud jste postupovali správně, dole na obrazovce se objeví bota na znamení toho, že si nyní pomocí kombinace tlačítek 1 - 6 můžete vybrat místnost a současným stisknutím devítky se do tohoto místa přenést.

POKE 35136,0 - nekonečný počet životů.

## JET SET WILLY

O této hře lze říci totéž, co o předchozí. Autorem obou je (v době jejich vzniku čtrnáctiletý) Mathew Smith. A i když dnes by asi ani jedna z nich nepatřila k nejlepším, ve své době bezkonkurenčně vedly žebříčky všech časopisů. Opět tedy jen malý výběr možností úprav:

POKE 34493,195 - tento pouk bude asi ve vaší verzi již "natvrdo", neboť umožňuje spuštění programu bez nutnosti znát "startovací" barevný kód.

POKE 35599,0 - nekonečný počet životů.

Posloupnost písmen W R I T E T Y P E R stisknutá při zobrazení místnosti FIRST LANDING spolu s kombinací číslic 1 - 6 vybavenou tlačítkem 9 vás přenesou do jedné



ze šedesáti místností. K oběma hrám si můžete sestavit tabulku místností a jim odpovídajících kódů. Jediné nebezpečí spočívá v tom, že některými kombinacemi můžete způsobit zhroucení systému. Zjišťovat všechny tyto kombinace je časově příliš náročné.

POKÉ 35123,0 - vymaže všechny pohybující se předměty.

POKE 41983,256-x - za x si dosadíte počet objektů, které musí Willy sebrat, aby ho jeho žena Maria pustila do postele. Původní hodnota x je 83 a získáte ji za 79 blikajících předmětů, z nichž dva se počítají za 2 body (neblikají, protože jsou neviditelné). Jejich umístění vám neprozradím - ale lehce je poznáte podle toho, že vám na počítadle přibude bod.

POKE 38240,0 - komu není Maria sympatická, může se jí touto úpravou elegantně zbavit.

### ATIC ATAC

POKE 36519,0 - nekonečný počet životů.

Při vkládání se neobjeví žádný nový problém, o všech byla řeč již u předchozích her. 3 kousky velkého zlatého klíče je třeba sestavit tak, aby z nich vznikl obrázek celistvého klíče na listině v pravé části obrazovky. Pak lze odejít hlavní branou, u které hra začíná. Předměty se sbírají i odkládají pomocí SYMBOL SHIFTu. Protože je nemůžete mít u sebe všechny najednou, je vhodné zvolit si jeden nebo více pokojů jako "skladiště" všeho, co by se mohlo někdy hodit.

### SABRE WULF

POKE 43575,244 - nekonečný počet životů. Při vkládání opět žádné nové problémy. Cílem hry je nalézt čtyři části medailónu znázorňujícího hlavu vlka. Potom vás příšerka (hlídač) pustí do jeskyně uprostřed džungle, kde je cíl. Cesta do této jeskyně je složitá, i když je hned naproti východu ze startovní mýtiny.

### UNDERWURLDE

Jste uprostřed zámku s velmi rozsáhlým podzemím. Přestože má tři východy, není vůbec jednoduché dostat se ve zdraví ven. Pro vaši představu - hrací plocha má 50 pater. Na každého z démonů hlídajících pod velkým krápníkem platí jiná zbraň. Na obyčejnou "havěť" platí cokoliv. Přes démona se dá dostat i bez jeho odstranění. Budete-li se mu neustále vnucovat, "propasíruje" vás nakonec některá z příšerek na druhou stranu. Chce to jen trpělivost a trochu štěstí.

POKE 59376,0 - nekonečný počet životů.

POKE 38043,0 - nesmrtelnost po nalezení diamantu (gemu) se z několika sekund prodlouží navždy.

POKE 45019,201 - zmizí všechny dotěrné příšerky.



POKE 59591,0 - opět pouk pro sportovnějšší typy. Hra se nezmění, pouze zbraně naleznete vždy na stejném místě.

**KNIGHT LORE**

POKE 53567,0 - nekonečný počet životů.

POKE 50084,201 - odstraní v některých okamžicích krajně nevhodný přerod z rytíře na vlka a naopak.

POKE 49759,x - za x dosaďte počet věcí, které je třeba hodit do kotle, abyste hru dokončili. Do kotle musíte vhodit vždy ten předmět, jehož tvar se nad ním objevuje. Pokud vám vadí, že na překonání vyšších překážek vždy nějaký předmět spotřebujete, zkuste vždy těsně po odrazu stisknout "brací" tlačítko.

POKE 50205,0: POKE 50206: POKE 50207 - zastavení času.

**PSSST**

POKE 24984,0 - nekonečně životů.

**COOKIE**

POKE 26197,0 - z popelnic přestanou vyletovat odpadky, které kazí vámi vařenou dobrotu.

POKE 28698,0 - nekonečný počet životů.

**HORACE GOES SKIING**

POKE 29009,0 - odstraní ze silnice většinu automobilů.

**TORNADO LOW LEVEL**

Tento program patří k těm zatajenějším, jeho úprava je těžší.

Pomocí MERGE"" nahrajte úvodní BASIC a změňte řádky 20, 50 a 3000 takto:

```
20 DATA 55,62,255,221,33,0,64,17,156,191,205,86,5,62,0
```

```
50,190,136,50,15,132,201
```

```
50 FOR n 65423 TO 65444
```

```
.
```

```
.
```

```
.
```

```
3000 RANDOMIZE USR 65423
```

Tak vám nebudou ubývat letadla a zastaví se čas.

Další úpravy se týkají her, které Mikrobáze zatím ve své nabídce nemá, ale lze předpokládat jejich rozšíření mezi majiteli ZX Spectra, protože jsou již staršího data.

ZZOOM - POKE 24743,0 - nekonečný počet životů.



TRANSAM - POKE 25446,0 - nekonečný počet životů.

SCUBA DIVE - POKE 55711,x - x je počet životů.

ARCADIA - POKE 25776,0 - nekonečný počet životů.

THE PIRAMID - POKE 44685,0 - neubývá energie.

BOOTY - POKE 58294,0 - nekonečný počet životů.

HUNCHBACK - POKE 26888,0 - nekonečný počet životů.

- POKE 26903,x: POKE 24760,x - za x dosaďte počet životů v intervalu  
0 - 18.

TERROR DAKTYL - POKE 37629,0 - nekonečný počet životů.

KOKOTONI WILF - POKE 43742,0 - nekonečný počet životů.

Pokud vám tento příspěvek zkrátil bezesné noci a navrátil vás zpět rodině nebo známým, jeho účelu bylo dosaženo. Zkušenost totiž říká, že k jednou - byť "nepoctivě" - dohrané hře se již žádný i sebevětší fanatik nevrací nijak zvlášť rád. A ještě jedna prosba nakonec. Nevyužívejte poznatků z tohoto článku k výrobě nových pozměněných verzí her. Je smutné, kolik programů bylo neodborným zásahem poškozeno nebo zničeno a kolik se jich přesto dostalo mezi roztrpčené uživatele. Ve světě mikropočítačů i tak vládne zmatek, který není třeba dál zvětšovat.

MB

# Horká linka infraLEDové komunikace

Amatérská činnost se neobejde bez experimentů, které v určitém počtu případů nefungují hned tak, jak si autor zapojení původně přál. Z tohoto počtu případů má určité procento následky katastrofální - přestane fungovat všechno. Pak nezbývá než zařízení - v lepším případě jen jeho neživotnou část

- svěřit popelnicím, připravit si procítěnou řeč k rodinnému kruhu o nezbytnosti potřeby zvýšení finančního rozpočtu na vědecký výzkum a se schválenou částkou v ruce se od příštího dne věnovat shánění chybějící součástkové základny.



Pokud se v našem zapojení zakouří jen z odporu či tranzistoru, újma je poměrně zanedbatelná. Horší věci mohou nastat třeba při experimentech s ovládním nějaké periférie výstupem z počítače. U vědomí toho, že nad dýmajícím computerem by nám sebeprocítěnější řeč nebyla nic platná, je lépe k pokusům toho druhu přistupovat se vši obezřetností. Následující zkrácený překlad z měsíčníku Computing Age 12/85 nabízí řešení problému popisem obvodů bezdrátového spojení počítače s perifériemi pomocí infračervených diod.

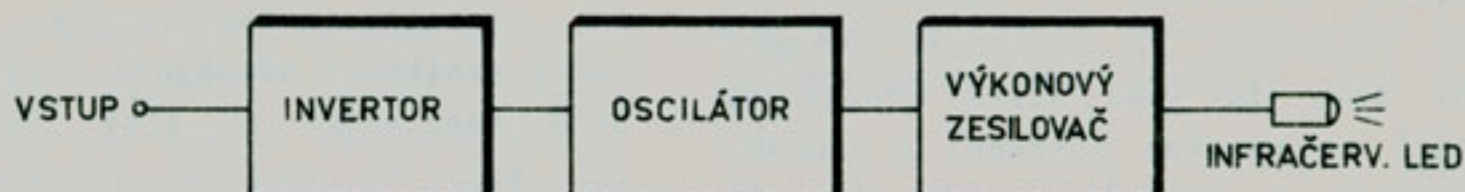
Zapojení sestává ze dvou částí - vysílače a přijímače digitálního signálu. Použitý princip je nejjednodušší možný - svítící dioda znamená log.1, zhasnutá log.0. Základem přijímače je zpětnovazební fázový závěs reprezentova-

ný obvodem NE567 ve funkci tónového dekodéru. Vysílač se opírá o zapojení s obvodem 555. Blokové schéma zapojení je na obr. 1.

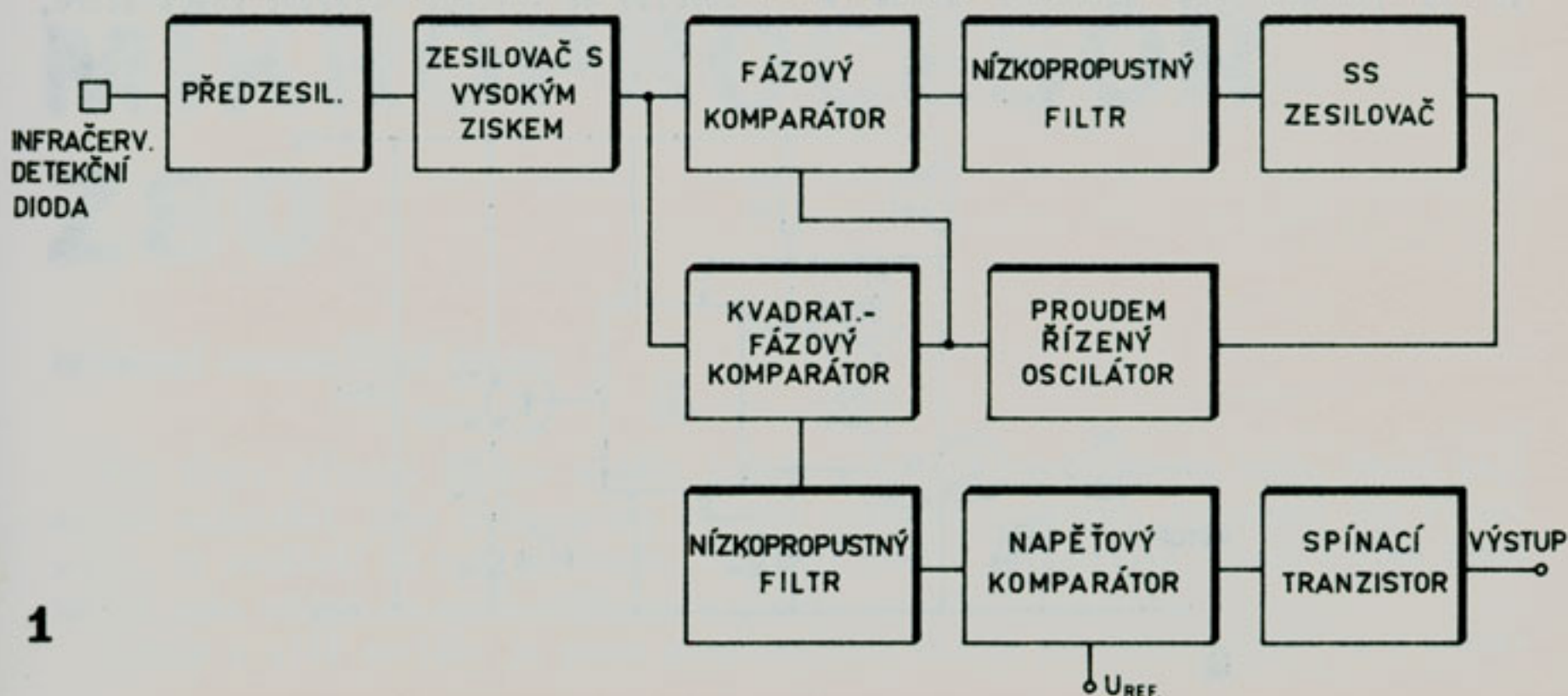
Vysílač je velmi jednoduchým oscilátorem, pracujícím na kmitočtu několika kHz. Oscilátor je klíčován invertorem, který překlápí signál přicházející z výstupu počítače. Za oscilátorem je zesilovač zajišťující optimální proudové vybuzení výstupní diody.

Přijímač zesiluje signál z diody ve třech stupních (předzesilovač a dvoustupňový zesilovač s velkým ziskem). Funkce obvodu NE567 je zřejmá z blokového schématu. V daném zapojení změna kmitočtu vstupního signálu mění napětí na výstupu (zde jsou jen dva vstupní stavy, které způsobují trvalé vypnutí nebo zapnutí výstupního tranzistoru

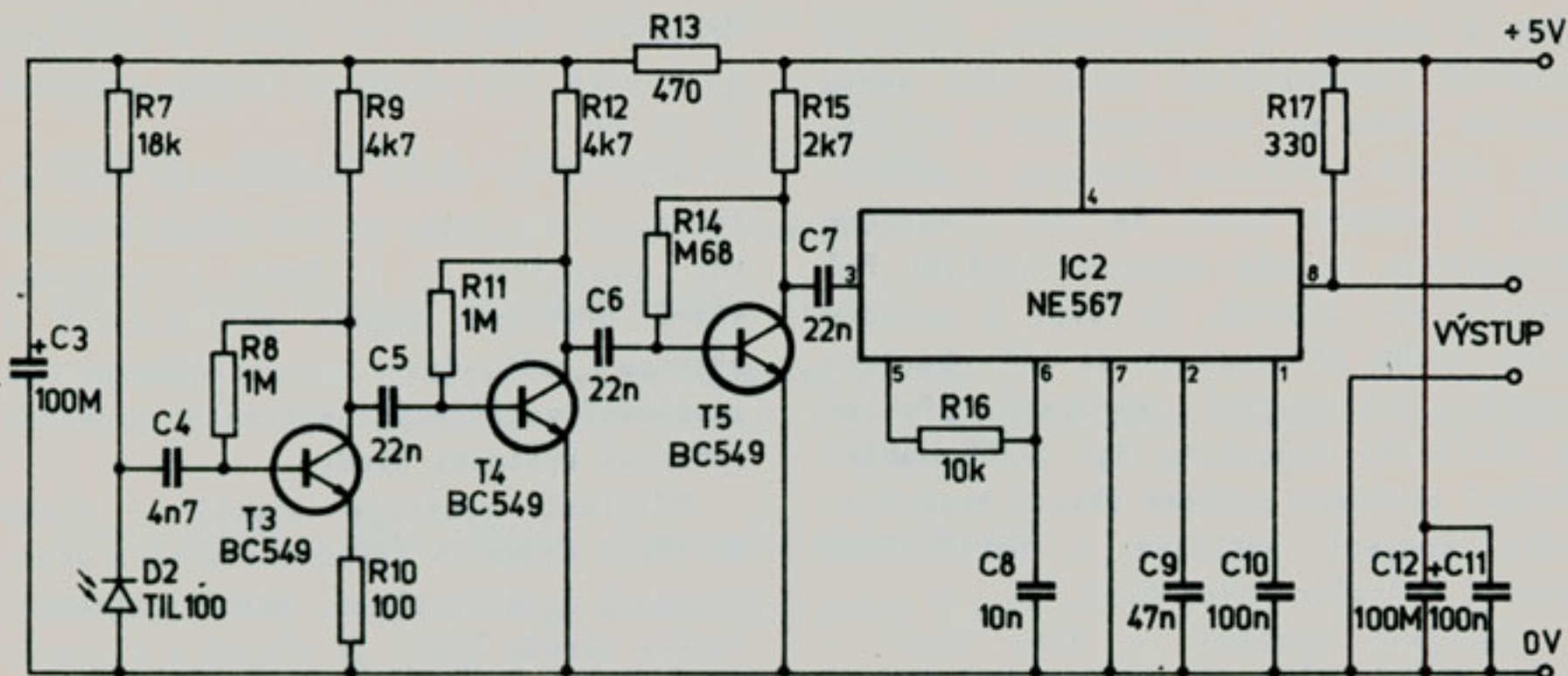
## VYSÍLAČ



## PŘIJÍMAČ







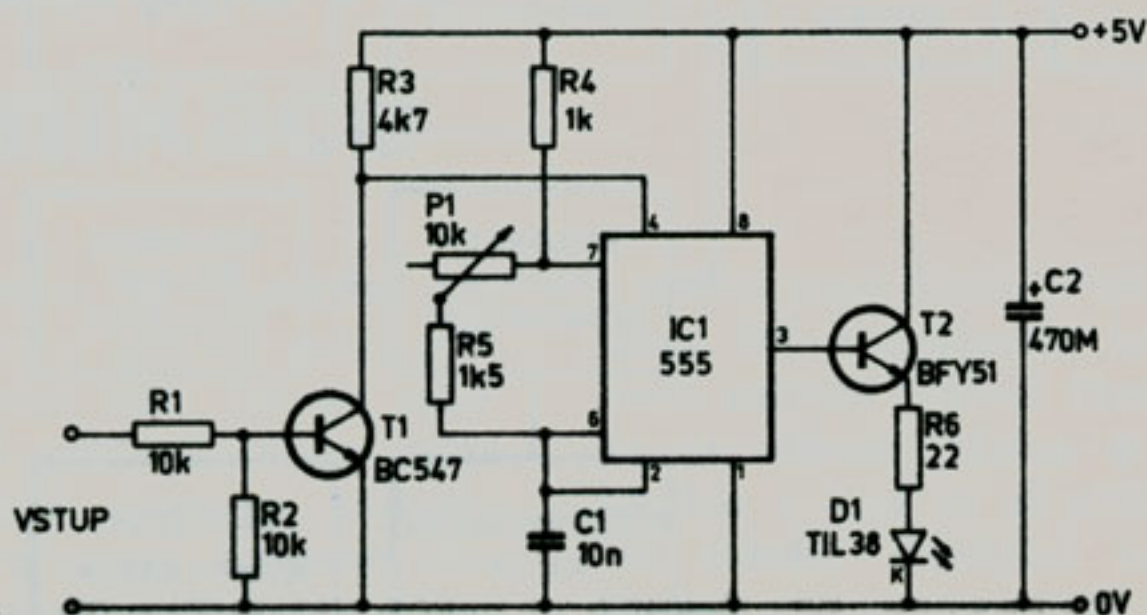
2

s otevřeným kolektorem obvodu NE567).

Zapojení vysílače je na obr.2. Obvod 555 je astabilní oscilátor s říditelným výstupním kmitočtem. P1 nastavuje výstupní kmitočet. T1 invertuje vstupní signál tak, že jeho nízká úroveň se převrátí na vysokou výstupní a naopak. T2 pracuje v kmitočtu nastaveném P1 a při otevření vyvolá emisi diody průtokem proudu asi 100 mA.

Obr.3 ukazuje zapojení přijímače. D2 je infračervená detekční dioda s

filtrem, propouštějícím jen infra paprsky. Impulzy z vysílací diody způsobí vznik malých záporných impulsů na D2 a R7. Hodnoty C5 a C6 byly zvoleny tak, aby zamezily interferencím, které by mohly vzniknout působením jiných spotřebičů napájených ze sítě ("šum" kolem 100 Hz). Po zesílení tranzistoru T3, T4 a T5 je signál dále přiveden na obvod NE567. R16 a C8 časují proudově řízený oscilátor obvodu. C9 (v součinnosti s vnitřním rezistorem obvodu) filtruje impulsy na výstupu fázového komparátoru,



3



aby výstup mohl být (v případě log.1 na výstupu počítače) kontinuálně kladný. C10 vyhlazuje impulsy na výstupu kvadratického fázového komparátoru.

Vysílač připojíme na datovou sběrnici počítače (D1-D8 u osmibitových). Samozřejmě, že na každou z osmi (programově ovládaných) linek můžeme připojit po jednom vysílači. Za úvahu stojí i možnost úpravy zapojení tak, že by na základě jednoho z několika možných kmitočtů vysílaného signálu byla momentálně řízena vždy jen jedna z několika periférií (nebo jejich sestava), z nichž každá by byla "naladěna" na svůj kmitočet.

Na závěr snad ani není třeba dodávat, že obě infra části musí být chráněny před působením světla zvenčí.

Toto jednoduché zapojení může posloužit v řadě aplikací počítačem řízených periférií. Je přímo předurčeno pro řízení domácích spotřebičů (včetně možnosti programového časování jejich chodu). Pochopitelně pak musíme výstupní signál připojit ke vhodnému obvodu (tyristor,

triak, relé), který bude ovládané zařízení ovlivňovat. Zapojením můžeme nejen vysílat řídicí pulsy z počítače, ale "obrátime-li" je, i přijímat informace z periférií a ty programově zpracovávat (např. nechat počítač automaticky rozhodovat, kdy a za jakých podmínek vypnout nebo zapnout topení, ventilaci, žárovky, magnetofon atd.). K řadě takovýchto rozhodovacích procesů je třeba indikace stavu řízených spotřebičů. To vyžaduje volbu optimálních indikátorů i jejich zapojení do obvodů, které budou na výstupu "hovořit jasnou řečí".

Abychom pronikli hlouběji do oblasti počítačem řízených periférií, anglicky zvané "interfejsing" (a pomohli tak těm z vás, kterým četba tohoto článku zatím nic neříká), v příštích Zpravodajích si povíme o zapojení a funkci jednoduchých převodníků A/D, D/A i s praktickými soft/hardwareovými ukázkami určenými uživatelům ZX Spectra (avšak lehce obměnitelnými pro jakýkoli jiný počítač).

-elzet-

# MIKROPROCESOR Z80

## AUTONOMNÍ ASSEMBLEROVÉ RUTINY

Pod tímto titulem budeme uvádět kratší programy ve strojovém kódu, které mají obecné uplatnění jako podprogramy velkého množství programů. Autonomní jim říkáme proto, že nezáleží na typu počítače, do jehož paměti je umístíme - není v nich žádné volání nebo odskok do rutin ROMky nebo jiného operačního systému.

Jedním z cílů seriálu je pomoc programujícím členům Mikrobáze, kteří se s assem-



blerem a strojovým kódem teprve začínají seznamovat; své v něm najdou určitě i ti pokročilejší. Jako další konkrétní čin v rámci této pomoci ve Zpravodaji najdete volný list se všemi assemblerovými instrukcemi Z80. Měl by se stát základní orientační pomůckou, kterou můžete mít při sestavování programu vždy pohotově po ruce. Zároveň tak vyhovujeme mnohým z vás, kteří po takovéto programové pomůcce volají.

Rutiny budeme zatím přebírat převážně z knihy *Assembler Routines for the Z80* z anglické edice *Personal Computer World*. Jejím autorem je David Barrow. Byli bychom rádi, kdybyste i vy přispěli do tohoto seriálu svými vlastními autonomními rutinami, které velmi rádi otiskneme.

Za zmínku stojí forma, v jaké jsou výpisy rutin uvedeny. Máme za to, že by se mohla stát standardem tiskového podání programů ve strojovém kódu (assembleru). K orientaci v užité průvodní dokumentaci (charakterizaci) programů je nutno podat vysvětlení.

Po názvu rutiny (předznamenáném symbolem \*) následuje její stručná charakteristika, která je dále rozvedena pod následujícími hesly:

#### Obecná charakteristika:

Činnost - k čemu (jakému účelu) rutina slouží.

Akce - metoda, jakou je činnosti dosaženo.

#### Systémová implementace:

CPU - typ procesoru, pro nějž je rutina sestrojena a některé další ev. údaje (např. kmitočet taktovacích impulsů v MHz apod.).

Hardware - v případě, že rutina spolupracuje s nějakou specifickou hardwarovou částí počítače nebo ovlivňuje (řídí) či potřebuje ke své činnosti určitý hardware, uvést jeho specifikaci.

Software - podobně jako u hardwaru.

#### Operační detaily:

Vstup - uvedení případně požadovaného stavu indikátorů (flags), registrů, obsahu paměti, zásobníku (stack), pokud musí být před vstupem do rutiny předem inicializovány pro její správnou činnost.

Výstup - uvedení všech významných změn, které po skončení (odskoku z) rutiny její činnost po sobě zanechá.

Chyby - nejobtížnější část hodnocení funkce rutiny - mají v ní být uvedeny všechny možné chyby (disfunkce), které mohou při její činnosti nastat - např. při vložení chybných dat, při přerušení běhu, nebo z jiných příčin.

Registry - uvedení všech registrů, jejichž obsah je činností rutiny ovlivňován (mohou nabýt jiné hodnoty, než jakou měly při svém vstupu do rutiny).

Zásobník - maximální počet bajtů, které jsou během chodu rutiny uloženy do stacku v jednom momentu (popř. max. rozsah paměti, kterou zásobník v činnosti rutiny zabírá); neuvádějí se bajty, kterými celou rutinu voláme, stejně jako bajty adresy návratu po skončení její činnosti.

RAM - počet bajtů, užitých pro výměnu (ukládání) informací (dat) mezi rutinou a volajícím programem (např. DEFB, workspace apod.); nepočítají se bajty



zásobníku; při užití procesoru 6502 nutno uvést jeho pseudo-registry MO-MF.

- Délka - počet bajtů, které rutina obsazuje.  
Cykly - průměrný čas (v hodinových cyklech), potřebný k vykonání celé činnosti rutiny.

### Klasifikace rutiny:

Třída - šest hodnotících kritérií; při splnění daného kritéria je toto označeno \*, při jeho nesplnění -. Třída 1 je jen taková, která splňuje všechna tato kritéria:

- diskrétní - nemění žádné proměnné kromě dat, pro jejichž zpracování je konstruována;  
přerušitelná - lze ji přerušit bez jakýchkoli negativních následků pro její další bezchybnou činnost i běh celého programu, do nějž je zařazena;  
promovatelná - je možno ji umístit do paměti PROM, EPROM atd., tedy musí splňovat požadavek neměnnosti svých bajtů během své činnosti;  
opakovatelná - do rutiny je možno znovu vstoupit po přerušení jejího běhu bez jakéhokoli negativního vlivu na její běh i chod celého programu;  
relokovatelná - bez jakýchkoli úprav ji lze umístit kamkoli do volné paměti, aniž pozbuje své funkčnosti (nevyskytují se v ní přímá volání jejích vlastních adres);  
robustní - jinými slovy "nezkolabovatelná".

### Výpis rutiny:

obsahuje zleva doprava:

návěští (label), assemblerovou mnemoniku, operand, komentář oddělený středníkem a strojový kód v hexadecimálním tvaru.

Dnes uvádíme rutiny pro časové prodlevy (delays), které mají při programování široké užití při zpomalování chodu, měření času, přerušení atd.

-----  
\* DLiS                    Prodleva na dobu 1 sec. (Delay)                    \*

-----  
:Činnost                časovaná rutina, obsahující cyklus, který spolu s instrukcí volání této rutiny zajišťuje prodlevu 1 sekundy.  
:Akce                    nastavení čítače cyklu INT/Clock Hz/n/,  
                          n je počet cyklů,  
                          jemné doladění.

-----  
:CPU                    Z80 s hodinovým kmitočtem 2 MHz  
:Hardware              -  
:Software              -  
-----



```

:Vstup      -
:Výstup     -
:Chyby      nepřesnost při přerušení
:Registry   -
:Zásobník   6
:RAM        -
:Délka      19
:Cykly      1999983 (42551*47+86)

```

---

```

:Třída 2    *diskrétní      -přerušitelná      *promovatelná
            -opakovatelná  *relokovatelná    -robustní

```

---

```

:
LPCNT      EQU      42251      ;hodnota čítače cyklu
:
:::Uložení a inicializace, 32 cyklů.
:
DLiS       PUSH AF           ;(11)   Uložení registrů a      F5
           PUSH BC           ;(11)   flagů užitých v DLiS      C5
           LD  BC,LPCNT      ;(10)   Nastavení čítače        01 37 A6
:
:::Hlavní cyklus; 47*42551-5 cyklů.
:
DLOOP      PUSH HL           ;(11)   21 cyklů                E5
           POP  HL           ;(10)                E1
           DEC  BC           ;(6)   Dekrementace čítače      0B
           LD  A,C           ;(4)   a test na nulu           79
           OR  B             ;(4)                80
           JR  NZ,DLOOP      ;(12/7) Opakování cyklu          20 F9
:
:::Jemné doladění, obnovení registrů a návrat; 59 cyklů.
:
           PUSH HL           ;(11)   21 cyklů                E5
           POP  HL           ;(10)                E1
           NOP                ;(4)   8 cyklů                00
           NOP                ;(4)                00
           POP  BC           ;(10)   Obnovení registrů        C1
           POP  AF           ;(10)   a flagů                  F1
           RET                ;(10)   Návrat po 1 sec.         C9

```

---

### Úprava DLiS na jiné kmitočty taktu procesoru

První zet-osmdesátky pracovaly s kmitočtem 2 MHz. Z80A pracuje na 4 MHz. ZX Spec-



trum používá kmitočet 3,5 MHz. Je nutno si dále uvědomit, že váš počítač bude pravděpodobně od jmenovitého kmitočtu trochu odchýlen. Proto je nutné rutinu dodatečně doladit.

32 cyklů pro vstupní uložení registrů a inicializaci rutiny, stejně jako 30 cyklů pro výstupní obnovení původních obsahů registrů a návrat, je neměnných. Podobně je tomu u testu dekrementovaných reg.BC (instrukce JR NZ,xx trvá vždy 26 hodinových cyklů, pokud zero flag nedosáhne hodnoty nula - pak tato instrukce trvá jen 21 hod. cyklů). Ostatní cykly lze měnit - prvotní hodnotou čítače v reg. BC a vkládáním "nesmyslných" instrukcí, jejichž smysl je v tom, že počítači zabírají určitý počet hod. cyklů pro jejich vykonání. Tak můžeme definovat časovou formulku rutiny DLiS:

$$(\text{hodinový kmitočet v Hz} - 17) = 62 + (\text{LPCNT} * (\text{"main"} + 26)) - 5 + \text{"fine"}$$

kde:  $0 < \text{LPCNT} < = 65536$ ; "main" =  $> 0$ ; "fine" =  $> 0$

a kde "nesmyslné" instrukce korigují dobu trvání cyklů "main" a "fine".

Pokud v počítači máte Z80A s kmitočtem 4 MHz, prodlevy 1 s dosáhnete jednoduše tak, že rutinu DLiS zavoláte dvakrát po sobě. Jinou alternativou je změna hodnoty čítače na 58822 (0E5Ch) při užití těchto "nesmyslných" instrukcí (pro "main" a "fine"):

```

PUSH HL          ;(11)          E5
POP  HL          ;(10)          E1
PUSH HL          ;(11)          E5
POP  HL          ;(10)          E1
a
PUSH HL          ;(11)          E5
POP  HL          ;(10)          E1
LD   A,R         ;(9)           ED 5F

```

Úpravu pro ZX Spectrum (3,5 MHz) provedeme takto:

- LPCNT nastavíme na hodnotu 51469 (0C90h),
- použijte (jen v "main") stejné "nesmyslnosti" jako u 4 MHz,
- do "fine" pro 4 MHz přidejte jeden NOP (nárůst na 34 cyklů).

V každém případě nezapomeňme rutinu jemně doladit.

-----  
\* UDELAY            Variabilní řízená prodleva (Universal Delay)            \*

-----  
:Činnost            časově řízená rutina pro precizní nastavení požadované  
                     prodlevy  
:Akce                vstupními daty se řídí délka výsledné prodlevy  
-----

```

:CPU                Z80
:Hardware           -
:Software           -
-----
```



```

:Vstup      registr A obsahuje požadovaný počet cyklů (0 až 255)
:Výstup     registr A je vynulován
:Chyby      nepřesnost v případě přerušení nebo když FRAC je assemblován jako
             hodnota nižší než 1223
:Registry   AF
:Zásobník   4
:RAM        -
:Délka      30
:Cykly      FRAC*A (včetně 17 pro volání rutiny)

```

---

```

Třída 2      -diskrétní          -přerušitelná      *promovatelná
             -opakovatelná      *relokovatelná    -robustní

```

---

```

:
:::Assemblování časové jednotky FRAC jako počet hodinových cyklů
:::v požadovaném zlomku vteřiny.

```

```

FRAC      EQU      nnnn          ;1223 < = nnnn < 65536
:
UDTOH     EQU      -1223        ;UDELAY časové přetečení, odečítané od
                                ;FRAC při 1. vstupu do NUMLP
NLTOH     EQU      -1149        ;NUMLP časové přetečení, odečítá se od
                                ;FRAC při každém průchodu NUMLP
DLTIME    EQU      -33          ;DIVLP operační čas, odečítaný od FRAC
                                ;opakovaně až do návratu z rutiny

```

```

:
:::Přetečení (vč. CALL): 17+32+1149-5+30=1223

```

```

UDELAY    PUSH BC              ;(11)  Uložení registrů a          C5
           PUSH HL              ;(11)  nastavení BC na          E5
           LD BC,UDTOH          ;(10)  UDELAY přetečení          01 39 FB

```

```

:
:::Opakování A krát; přetečení 10+61+1052+26=1149

```

```

NUMLP     LD HL,FRAC           ;(10)  Cykly čas. jednotky          21 nn nn

```

```

:
:::Počet cyklů 61+33*INT((FRAC-přetečení):33)

```

```

DIVLPL    ADD HL,BC            ;(11)  Odečet přetečení a          09
           LD BC,DLTIME        ;(10)  provedení cyklu          01 DF FF
           JR C,DIVLPL         ;(12/7) DIVLPL                38 FA

```

```

:
:::Počet cyklů 1052+ZBYTEK((FRAC-přetečení):33)

```



ZBYTLP	INC L	;(4)	12 cyklů, když není	2C
	JR NZ,ZBYTCT	;(12/7)	zbytek, jinak	20 01
	DEC HL	;(6)	13 a HL zpět na	2B
ZBYTCT	INC C	;(4)	správný počet	0C
	JR NZ,TBYTLP	;(12/7)	uskutečněných cyklů	20 F9
:				
	LD BC,NLTOH	;(10)	Inic.BC na NUMLP	01 83 FB
	DEC A	;(4)	časové přetečení	3D
	JR NZ,NUMLP	;(12/7)	Jedn. prodleva*A	20 EA
:				
	POP HL	;(10)	Obnovení registrů	E1
	POP BC	;(10)	a návrat po	C1
	RET	;(10)	A*nejkratší prodleva	C9

Uvedená rutina (při hodinovém kmitočtu 2 MHz) umožňuje řídit prodlevu v rozmezí 1/1600 až 8,4 vteřiny (pro nejdelší prodlevu je FRAC = 65535 a vstupní hodnota reg. A = 0). Použitý výraz přetečení je použit pro dobu trvání jednotlivých instrukcí, která se postupně odečítá (záporné hodnoty v EQU), aby k výpočtu zůstal jen produko- vaný "čistý" čas, který pro dobu trvání prodlevy zadáváme. Základní princip tvorby prodlevy však zůstává týž, jako v předchozím případě (cyklování). Využití rutiny je v programech, které vyžadují dynamiku procesů, tedy proměnné časové řízení.

V příštím Zpravodaji bude m.j. uvedena velmi užitečná rutina pro přemístění "nepřemístitelných" programů ve strojovém kódu.

## Programová nabídka MIKROBÁZE

**Vážení členové mikrobáze,**

v programové nabídce Mikrobáze dochází k zásadnímu obratu, který je přirozeným vyústěním celosvětového vývoje v oblasti autorskoprávní ochrany programů i sílící potřeby tvorby původního tuzemského softwaru. I když řada otázek spojená s ochranou programů u nás ještě zdaleka není vyřešena, z hlediska perspektivního se Mikrobáze rozhodla přistoupit k respektování autorských práv zahraničních programů.

Koncem roku minulého a zvláště začátkem tohoto přistoupily legislativní orgány řady zemí k řešení uvedené problematiky, na něž navazovalo místy až velmi tvrdé prosazování nových předpisů do praxe. Postupem času se "lokální" ochrana rozšiřuje



i na díla zahraniční na základě vícestranných smluv. Tak lze předpokládat, že není daleko moment vzniku mezinárodní konvence pro autorskopravní ochranu softwaru, jehož signatářem jako je tomu u konvencí ochrany jiných autorských děl, se stane i ČSSR.

Bohužel je nutno konstatovat, že současný stav věci u nás je v "bodu nula", který rozvoji výpočetní techniky nijak neprospívá. Konkrétně - autora programu nelze odměnit jinak, než mzdou, což je v rozporu s charakterem jím vytvářeného díla. Záležitost dále komplikuje skutečnost, že dílo samo potom nebude požívat právní ochrany. Nelze se proto divit, že se u nás ještě nerozběhla produkce původního domácího softwaru, který by byl běžně dostupný.

Přes uvedené administrativní vakuum se Mikrobáze rozhodla nečekat a přikročit k položení základů pro tvorbu původního tuzemského, kvalitního softwaru.

Nově vzniklá situace si žádá nový přístup k práci. Výbor 602. ZO Svazarmu proto rozhodl:

1. Vyloučit šíření zahraničních programů, pokud k jejich šíření nebude možno získat patřičné oprávnění.

2. Protože v současné době podstatnou část členů Mikrobáze tvoří majitelé mikropočítače ZX Spectrum, byly uzavřeny smlouvy na vytvoření zásadních programů pro tento počítač s přihlédnutím k tomu, aby programy byly bez problémů upravitelné i pro jiné mikropočítače. Jde o tyto programy:

a) **Český a slovenský slovní procesor** (analogie programů Tasword nebo Spectral Writer) s možností snadné úpravy pro jiné mikropočítače (minimální využití rutin paměti ROM), s měnitelnými softwarovými bloky pro různé typy tiskáren a interfejsů a s vytvořením standardu pro psaní českých a slovenských písmen na zahraničních klávesnicích typu QWERTY bez omezení původního rozsahu znakového ASCII kódu.

b) **Databázový program**, který bude spojovat rychlost programu VU-FILE s variabilitou databáze Master File. Nový program bude samozřejmě pracovat s diakritickými znaménky pro český i slovenský jazyk v plném rozsahu.

c) **Generátory a monitory strojového kódu**, vycházející z programů GENS a MONS. Jednotlivé verze budou rozšířeny o četné užitečné příkazy a funkce, bude ulehčena spolupráce generátoru s monitorem při tvorbě vlastního zdrojového kódu. Nově přepracovaný program bude snadno upravitelný pro všechny druhy mikropočítačů s potřebným rozsahem paměti.

Programy pod písmeny a) a b) budou zcela původní, u písmene c) se jedná o částečně kompilační využití partií jiných programů s cílem zvýšení jejich výsledné užitkové a edukační hodnoty. Verze všech těchto programů pro další mikropočítače bude Mikrobáze nabízet postupně tak, jak je jejich autoři pro ně upraví.



**NEPŘEHLÉDNĚTE!!!**



Vzhledem k nové situaci musím s omluvou vyzvat všechny zájemce o nové programy, aby si je obvyklým způsobem objednali znovu. Nestačí už zaslané objednávky na podobné zahraniční programy. Neobjednávejte programy, které nenabízíme.



Pokud jde o celkové provedení nových programů, budou na profesionální úrovni (samozřejmě ve strojovém kódu), přičemž smluvní povinností jejich tvůrců je nedopustit se některých funkčních nedostatků, kterými trpí uvedené programy původní. Zárukou kvality nových programů je i nově ustavená odborná rada Mikrobáze, která bude programy posuzovat z hlediska odborného i uživatelského formou oponentury.

Nesporná výhoda, kterou všem uživatelům přinese nastoupený směr, bude spočívat v tom, že programové manuály budou obsahovat řadu důležitých informací o stavbě nových programů, takže si je v mnohém budete moci sami upravit podle potřeb svých specifických aplikací. Nedostačující informace v zahraničních manuálech považujeme za přemrštěnou komerční ochranu, která uživateli brání v jednoduchých úpravách. O tom, že chceme při práci s počítačem použít svůj rodný jazyk, určitě není třeba diskutovat. Nové programy to v plné míře umožní.

Velmi podstatným aspektem nového přístupu je diference ve vztahu producent-uživatel, jaký praktikují komerční firmy a jaký bude uplatňován v Mikrobázi ve smyslu organizace-její člen. Všechny původní užitkové a systémové programy Mikrobáze budou diskutovány na stránkách tohoto zpravodaje. Opodstatněné praktické požadavky na změny nebo doplnění funkcí stěžejních programů budou realizovány v dalších verzích, eventuálně budou uvedeny ve zpravodaji. Díky tomu se tyto programy budou stále cizelovat a transponovat pro široké užití i na dalších typech mikropočítačů ve spojení s různými druhy externích zařízení (tiskárny, záznamové jednotky apod.).

Spektrum užití slovního procesoru v praxi je velmi široké. Díky tomu, že většina nových (a stále levnějších) maticových tiskáren je vybavována volnou pamětí pro definování vlastního znakového souboru, přestává být problémem tisk jakékoli abecedy kteréhokoli jazyka - obecně jakýchkoli znaků. Tiskem češtiny a slovenštiny v různém provedení podle typu tiskárny se v souvislosti s novým slovním procesorem budeme návazně zabývat. Jedná se jak o tisk diakritických znamének v grafickém módu, tak o perspektivní vytváření datových souborů různých typů písma pro volné paměti moderních maticových tiskáren.

Další informace se týká předpokládaných termínů:

- **DrMG** - tento upravený generátor a monitor strojového kódu je programově hotov od září. Dodání autorského textu manuálu je smluvně zajištěno na polovinu října. Poté proběhne redakce textu a jeho příprava do tisku.
- **DIAPEN** - vytvoření slovního procesoru pro editaci češtiny a slovenštiny (včetně manuálu) je smluvně zajištěno na konec t.r. Oponentura rady Mikrobáze (autorská obhajoba programu) proběhne v lednu 1987, během nějž bude na základě ev. výhrad rady program upraven. Souběžně proběhne redakce manuálu.
- **DATALOG** - na tento databázový program se vztahují stejné termíny jako na DIAPEN.

Přehrávací zařízení pro pořizování kopií programů je po všech stránkách připraveno. V polovině září nám tiskárna dodala první pěkné, barevné obaly na manuály. Dodávka obalů na kazety z téže tiskárny je za dveřmi. Pro majitele počítačů SORD M5 tato zpráva znamená, že jejich objednávky budou vyřizovány již od listopadu t.r. Velmi nás těší, že "sordisté" si převážně vybrali právě z bohaté nabídky původních programů. Malá odezva na nabídku zahraničních programů pro ZX Spectrum Zpravodaje č.2 mj. prokazuje, že výměnné sítě, jimiž se tyto programy rozšiřují, si v tomto ohledu vystačí samy, a že pro Mikrobázi tudy cesta opravdu nevede.



S obratem v činnosti Mikrobáze souvisí i změna v kalkulaci členské úhrady za původní programy. Pochopitelně, že čím více objednávek na programy obdržíme, tím nižší výchozí kalkulace bude. Dosáhne-li např. součet objednávek na 1 původní uživatelský nebo systémový program (provázený bohatým manuálem) jen 200 kusů, nepřesáhne výchozí kalkulace členské úhrady částku 200,- Kčs. Podstatná část kalkulace připadá na samotnou pořizovací cenu kazety. Značnou položku tvoří tisk malého počtu manuálů. Nejedná se o žádnou velkosériovou výrobu - ale odněkud se začít musí.

Je tomu více než rok, kdy nově založená Mikrobáze aspirovala na "statut" pouhé distribuční jednotky převážně zahraničních programů. Bohužel byly oslyšeny hlasy, které varovaly před touto cestou a naznačovaly, že problematika věci je velmi široká, že je třeba jít jinou, komplexní cestou za zvýšením intelektu a původnosti naší výpočetní techniky. Vývoj dal těmto hlasům zapravdu. Během prázdnin proběhla v Mikrobázi i kolem ní řada ostrých, ale velmi konstruktivních diskusí a organizačních opatření. Mikrobáze ve své činnosti vstupuje na novou, kvalitativně odlišnou platformu, na níž nachází pravý smysl své další práce a svého příspěvku všestrannému rozvoji uplatnění výpočetní techniky v životě naší společnosti.

# ZX Spectrum

## DrMG

Na bázi známé kombinace programů GENS3 a MONS3 postavená úprava, která umožňuje např. jednodušší spolupráci mezi oběma částmi programu, odpadá starost se studenými a teplými starty, lze měnit začátek pracovní oblasti, při disassemblování se monitor neptá na adresu, kam překlad uložit, ale sám si vyhledá konec zdrojového textu generátoru, uloží překlad za něj a upraví příslušné parametry generátoru, dále je přidáno tolik potřebné "pípání" tlačítek, průvodní texty jsou slovenské, přidáný modul provádí přepočty mezi různými číselnými soustavami atd. Původní funkce obou základních programů zůstávají zachovány. Doplňeno dvěma svazky bohatého manuálu.

## DIAPEN

Slovní procesor pro editaci textu v českém a slovenském jazyce. Název je složen ze dvou slov - PEN je dnes již mezinárodním označením mnoha druhů písátek, DIA je zkratkou slova diakritický (rozlišovací), které ve spojení se znaménky označuje čárky, háčky, tečky, kroužky apod. u písmen mnoha abeced různých jazyků. DIAPEN počítá i s velmi jednoduchou úpravou pro editaci jakékoli abecedy mnoha dalších jazyků (od azbuky po norštinu). Na rozdíl od všelijakých úprav anglických editorů pro editaci diakritických znamének dosahuje DIAPEN zvláštní úpravou toho, že všechny původní znaky ASCII kódu zůstávají zachovány a vůbec se nemění jejich pozice na klávesnici. Výhoda tohoto řešení je mj. v tom, že na DIAPENU napsaná např. česká slova se na jiném editoru promítnou nikoli jako změť nesrozumitelných znaků, ale



budou u nich chybět jen dia-znaménka. Rovněž pro tisk českého textu z jiného editoru nebude třeba provádět žádné úpravy - text se výtiskne jen bez dia-znamének. Manuál DIAPENU bude obsahovat instruktáž provedení tisku písmen s těmito znaménky na tiskárnách, které mají buď grafický mód, nebo tzv. down-load do volné paměti tiskárny. Přímou na kazetě budou softwarové bloky pro práci s některými typy tiskáren a interfaců. DIAPEN bude obsahovat všechny funkce pro práci s textem, jak je tomu např. u Taswordu nebo Spectral Writeru a některé funkce nové; ovládání tiskárny je rozšířeno. Obecně lze říci, že DIAPEN vyplňuje výraznou mezeru, která je v oblasti editace češtiny a slovenštiny z klávesnic zahraničních mikropočítačů. Doplněno bohatým manuálem.

### DATALOG

Databanka, spojující rychlost programů ve strojovém kódu s variabilitou známého programu Master File. Tvůrci programu jsou vedeni sbahou o co nejjednodušší řešení ovládání databanky (zvláště při tvorbě formy a formátu výpisů záznamů) při zachování vysoké variability struktury DATALOGU. Prohledávání v DATALOGU tedy nebude shodné s běžným hledáním v knižním katalogu, ale položka u jakéhokoli hesla se kdykoli bude moci stát heslem pro hledání dat dle daného klíče. Jednou z prakticky velmi cenných funkcí DATALOGU bude možnost provádění úprav a oprav jeho záznamů bez složitěho "kurzorování", šiftování, či přepisování opravovaných záznamů. Program umožní i snadný přepis dat na tiskárny všech typů, což mnoho ze známých databázových programů neumožňuje. DATALOG najde uplatnění při sestavování jakékoli informační databanky (od katalogu známek po seznamy členů organizací) s vysokou operativností při vyhledávání a aktualizaci dat. Zápis záznamů bude pochopitelně možný v češtině nebo slovenštině, rovněž abecední řazení proběhne podle dané abecedy. Doplněno opět bohatým manuálem.

- elzet -

307

# KAREL

Tento populární programovací jazyk vám Mikrobáze nabízí hned ve třech provedeních pro tři různé mikropočítače. Z toho verze pro ZX Spectrum je zcela novou modifikací programu. Programový manuál se zabývá nejen programovacími povely, jeho rozsah je významně rozšířen o celou metodiku programování s tímto typem jazyka. Ve své objednávce nezapomeňte uvést, pro jaký typ počítače program objednáváte. Dále si krátce připomeneme základní charakteristiku KARLA.

KAREL je mikropočítačový program, který je současně zábavnou hrou i seriózní učební pomůckou. Pochází ze Stanfordské univerzity, kde byl původně koncipován jako předstupeň výuky programovacího jazyka Pascal. V naší implementaci je do jisté míry setřena jednostranná orientace na Pascal a přidání některých nových prvků činí z



tohoto programu univerzální prostředek pro počáteční fáze výuky moderního programování.

Niklaus Wirth, autor programovacího jazyka Pascal, uvádí, že program - algoritmy + datové struktury. V systému KAREL jsou datové struktury potlačeny, což umožňuje snadnější chápání základních pojmů a postupů používaných při sestavování algoritmických struktur programů. Získané poznatky a dovednosti jsou pak využitelné ve většině ostatních programovacích jazyků.

KAREL má své opodstatnění i při přípravě k programování v jazyce Basic, který má nejširší uplatnění v oblasti mikropočítačů. Je holou skutečností, že Basic nemá dostatek prvků podporujících tvorbu strukturovaných a modulárních programů. Kdo však zvládnul KARLA, má i v jazyce Basic předpoklady k vytváření účinných, přehledných a dobře modifikovatelných programů. Mikropočítače a roboty sice směřují k takové dokonalosti, že je nebude třeba programovat speciálními programovacími jazyky, ale algoritmizace je dovednost využitelná obecně, nejen při programování počítačů.

V našem programu je Karel jméno robota, který je nakreslen na obrazovce mikropočítače. Karel má na obrazovce svoje město ohraničené zdí a v tomto městě vykonává funkci dopravní služby. Může se přesouvat z křižovatky na křižovatku a ukládat i sbírat na křižovatkách dopravní kužely, značky.

Program KAREL pro mikropočítač je dělán tak, aby sám dával návod k další činnosti obsluhy; lze s ním pracovat, aniž by uživatel potřeboval jakoukoliv příručku. V případech, kdy pro stručnost není jeho pokyn jednoznačný, lze správný postup nalézt metodou pokusů a omylů. Přesto je vhodné, či spíše nutné, doplnit práci s programováním Karla také vysvětlením základních pojmů strukturovaného a modulárního programování. K tomu slouží tištěný instrukční materiál.

Karlovi se dávají povely běžnými českými slovy. Pochopení nových poznatků proto není ztěžováno současným učením anglických slov. Karel je také úslužný robot, všechno, co si může domyslet, udělá nebo napíše sám. S formální stránkou svého učení v nejvyšší míře sám napomáhá. A tak se stává z trpělivého žáka ještě trpělivějším učitelem. (A pro zkušené programátory se z učitele stává zábavný společník!)

# INDEX



V nejbližších týdnech bude Mikrobáze vybavena technickými prostředky pro zpracování nabídkových a poptávkových korespondenčních lístků, na nichž nám buď nabízejte, nebo žádáte specifické programy (netýká se objednávek kazet s manuály z programové nabídky Mikrobáze). Výbor 602. ZO se rozhodl řešit danou situaci založením nové služby členům Mikrobáze, nazvané INDEX.

Všechny programy, které jste nám dosud nabídli (a dále budete nabízet), budou



zařazeny do databázového systému na větším počítači. Každému bude přiřazen index, kterým bude vyjádřena charakteristika programu z hlediska jeho zaměření, resp. využití v určité aplikační oblasti.

Každý požadavek na určitý typ programu, vyjádřený indexem, bude zpracován tímto způsobem: Do počítače bude vložen index, na jehož základě proběhne prohledání záznamů všech nabídnutých programů, zapsaných do databanky. Při nalezení shodnosti indexů u několika záznamů se na tiskárnu vypíší adresy členů Mikrobáze, kteří nám žádaný typ programu nabídlí. Tento výpis bude členovi Mikrobáze odeslán, aby se mohl kontaktovat se členy, kteří mu v jeho softwarovém hledání budou moci s velkou mírou pravděpodobnosti být nápomocni.

Cílem této služby je umožnit kontakt mezi členy s určitými specifickými zájmy, které se pravděpodobně budou prolínat s jejich profesí. Tak určitě existuje řada užitečných programů pro oblasti medicíny, stavebnictví, strojírenství, elektrotechniky, fyziky, chemie, astronomie, sportu, branných činností atd. Mikrobáze všechny tyto oblasti pochopitelně ve své programové nabídce postihnout nemůže. Takový úkol by ostatně byl nad síly jakékoli jedné jediné organizace. Nabízí se však možnost částečného zmapování stavu na těchto specifických softwarových tocích, po nichž díky INDEXu budete moci plout spolu s námi. INDEX bude však funkční jen tehdy, bude-li nabídka programů silná a stále doplňovaná; jinými slovy, ze samotných požadavků bychom databanku nenaplnili. V případě potřeby bude z databanky čerpat i Mikrobáze pro svou programovou nabídku.

INDEX je jen prostředníkem styku mezi členy; vlastní styk bude probíhat zcela mimo jakékoli případné nároky vůči Mikrobázi, tedy takříkajíc "na vlastní nebezpečí". Proto Mikrobáze nemůže jakkoli ručit za někým někomu nevrácenou kazetu či výpis programu apod. Na tomto místě se obracíme na ty z vás, kteří jste nám zaslali programové nabídky na korespondenčních lístcích v minulých měsících, abyste nám do 14 dnů po obdržení tohoto zpravodaje zaslali zamítavou zprávu (pouze) v případě, že si nepřejete, abyste do INDEXu byli zařazeni. Pokud si (nyní již všichni) nebudete přát uvedení své adresy na výpisech, které dostanou jiní členové, uveďte na novou řádku číslo 16 korespondenčního lístku s nabídkou -JEN TELEFON! (a dále město a číslo telefonu). Pokud chcete, aby na výpisech byla vaše adresa i telefon, vepište PLUS TELEFON (a město s číslem telefonu). Jména ani adresy žadatelů uvádět nebudeme. Nakonec ještě upozorňujeme na to, že pokud by chtěl někdo "využít" služeb INDEXu pro nedovolený prodej výsledků cizí práce, obdrží speciální index, který mu navždy zablokuje účast na jakýchkoli službách Mikrobáze a postupně i řady dalších ZO Svazarmu.



A kdy se INDEX rozběhne? Poté, co bude upravena struktura databáze pro dané funkce a do ní vloženy všechny záznamy z vašich nabídkových lístků. Pro tuto službu je v současné době připravován soubor indexů, který otiskneme v příštím zpravodaji. Podle něho se budete při využívání služeb INDEXu řídit. Do té doby můžete své nabídky posílat i nadále (typový index k nim zatím přiřadíme sami). Se zasíláním požadavků sečkejte ještě do vydání příštího zpravodaje. Ze součtu všech prací, které je pro hladký průběh INDEXu ještě nutno vykonat, vyplývá, že jeho ostrý start lze očekávat v lednu 87. Věříme, že INDEX řadě z vás přinese značný užitek nejen v zájmové, ale i profesní činnosti.



# Pokyny k objednávání programů

Samozřejmě, v platnosti zůstávají "pravidla hry" zveřejněná jak v Amatérském radiu (naposled v č. 5. ročníku 1985), tak v tiskovině s organizačními pokyny, kterou dostal každý, kdo projevil korespondenčním lístkem zájem o členství v Mikrobázi. Pro jistotu otiskujeme vzory vyplnění líce i rubu korespondenčního lístku k objednání programu z naší nabídky znovu.



Odesílatel:	
Ing. Jan Novák	
Jablonecká 56	
Liberec	
4 6 0 0 1	
82727	
MIKROBÁZE 250705/788	602. ZO Svazarmu
520214/0134 (rodné číslo)	Wintrova 8
	Praha 6
Vyhrazeno pro služební nálepky a údaje pošty	1 6 0 4 1 
50 h	



Rub lístku budete vyplňovat v řádcích 1, 5 a 15 (viz vzor A). Do řádku 1 napíšete OBJEDNÁVKA PROGRAMU, do řádku 5 označení programu (bloku programů) podle naší nabídky.

**POZOR!** Programy ještě nemají přesná katalogová označení, v nichž budou v budoucnu zakódovány typy počítačů. Proto zatím programy popisujte v této formě:

(řádek č. 5) 5. typ počítače                      číslo a název programu (bloku) dle nabídky

Rub korespondenčního lístku pro objednávku programu:



1.	OBJEDNÁVKA PROGRAMU	
2.		
3.		
4.		
5.	Sord M5	7. Dissassembler
6.		
7.		
8.		
9.		
10.		
11.		
12.		
13.		
14.		
15.	<i>Handwritten signature</i>	<i>8/12/1981</i>

Za typ počítače budete proto podle svých potřeb vypisovat Spectrum, IQ 151, SORD M5, PMD 85 nebo ZX 81, číslo a název programu opište z nabídky přesně. Je pochopitelné, že programy jsou mezi různými typy počítačů nepřenositelné. Nemůžete proto například požadovat program 2. PASCAL pro počítač IQ 151, ale jen pro Spectrum, tak jak je nabízen, jakkoli byste ho pro IQ 151 potřebovali a chtěli. (Na to jsou jiná "pravidla hry" - viz organizační pokyny expedované dříve.) Nu, a nezapomeňte v řádku 15 na podpis a datum. Jinak bude objednávka neplatná.





# Slovo k náhodným čtenářům

Dostal se vám tento zpravodaj Mikrobáze do rukou a zaujala vás aktivita rozvíjená v programových a technických službách pro uživatele osobních mikropočítačů? Redakce časopisu Amatérské radio jako iniciátor Mikrobáze a 602. ZO Svazarmu v Praze 6 jako realizátor vás zvou k členství v několikatisícovém kolektivu zájemců o výpočetní techniku a její aplikace. Jako člen Mikrobáze Svazarmu budete dostávat tyto zpravodaje (od roku 1986 čtyřikrát ročně), budete moci využívat programových nabídek a dalších plánovaných služeb.

Mikrobáze je službou pro mikropočítačovou techniku a ve své organizaci tuto techniku každodenně účelně a efektivně využívá. Proto je třeba i v případě, kdy teprve projevujete zájem o členství, dodržet určitou administrativní konvenci. O bližší informace o celém komplexu Mikrobáze a přihlašovací materiály je třeba požádat výhradně korespondenčním lístkem. Jeho líc vyplňte (zásadně strojem) podle vzoru č. 1. na straně 80. Rub korespondenčního lístku musí obsahovat v horní části čtyři číslované řádky s obsahem podle příkladu ve vzoru č. 2 (1. PŘIHLÁŠKA UŽIVATELE, 2. Jméno a příjmení, 3. Ulice, číslo, obec, poštovní směrovací číslo, okres, 4. Povolání/podnik, popřípadě škola). V dolní části lístku se podepište a uveďte datum vyplnění. Pak už stačí jen vhodit lístek do poštovní schránky a čekat na podrobné informační a přihlašovací materiály Mikrobáze. Dostanete je obratem.





1. PŘIHLÁŠKA UŽIVATELE
2. Ing. Jan Novák
3. Jablonecká 56. Liberec, 460 01, Liberec
4. Programátor analytik/Textilana

31. 12. 1985

*Novák*



## OPRAVY TISKOVÝCH CHYB VE ZPRAVODAJI MIKROBÁZE č. 2



Prosíme vás, abyste si v minulém čísle Zpravodaje opravili tyto chyby:

Ve výpisu rutin paměti ROM ZX Spectra na str.32 chybí 2 instrukce, které zařaďte za CALL 28B2, LOOK-VARS (4.řádka shora):

SET 7,C                      Bit 7 jména řetězce nastav na log. 1  
JR NC,0672,SA-V-OLD      Skok, když řetězec existuje

V článku Screen dump pro ZX Spectrum si na str. 46 v popisu bitové mapy znaku škrtněte všechny sedmičky (je jich celkem 7). V tomtéž článku si ve výpisu programu laskavě doplňte tyto 2 instrukce:

EESE 0620      LD B, 32              :32 sloupců print. řádky  
EE63 0E08      LD C, 08              :čítač osmi obrazových bitů

Na adrese EE84 má být hexadec. kód instr. 18CA (nikoli 18CE). Chyby vznikly při přepisu programu do slovního procesoru. Autor se všem čtenářům velice omlouvá.



ZO Svazarmu při SEŠ v Kroměříži nabízí ostatním ZO program využitelný při závodech DZBZ a SZBZ pro počítač PMD 85.

Využití programu: Pro místní a okresní klasifikační závody. Maximální počet závodníků 100. Lze rozdělit do všech kategorií. Program umožňuje i spolupráci s tiskárnou. Možno vytisknout startovní listinu (start. číslo, jméno, instituci) i listinu výsledkovou (pořadí, start. číslo, čas startu, čas cíle, ztrátové minuty, všechny položky střelby, zdržný čas, výsledný čas, sekundy za vítězem pro přidělení bodů podle tabulek DZBZ a SZBZ) časy startu a cíle je možné zadávat v čase reálném nebo nulovém. Při zadání času cíle se závodník zahrnuje do pořadí. Během závodu lze zjistit údaje o libovolném závodníkovi (časy, ztráty). Všechny údaje lze zadávat uspořádaně nebo neuspořádaně, ev. po blocích (časy startu, cíle, střelbu). Je možná i libovolná oprava jakýchkoli údajů kdykoli.

Veškeré informace o programu získáte na adrese:

Miroslav Pilát, předseda ZO Svazarmu při SEŠ Kroměříž, Pilařova 3/2, 767 11 Kroměříž.

Tato ZO vám rovněž může na základě objednávky zaslat na dobírku (50,- Kčs plus poštovné) kazetu s programem a jeho výpis.

---

Vydala 602. ZO Svazarmu pro potřeby vlastního aktivu. Zodpovědný redaktor ing. Jan Klabal, sestavení rukopisu Ladislav Zajíček. Adresa redakce: 602. ZO Svazarmu, Wintrova 8, 160 41 Praha 6, telefon 32 85 63. Neprodejně. Povoleno ONV Praha 6. Náklad 4000 výtisků. Praha, říjen 1986