

KANÁLY A PROUDY =====

Kanály proudy mohou být používány jak BASICem, tak programy ve strojovém kódu. V BASICových programech nelze ovšem využít všech možností, které mohou kanály uživateli poskytnout, i když i v BASICu je možné docílit některých zajímavých výsledků.

Myšlenka na níž je postavena existence kanálů je velmi jednoduchá: maximálně unifikovat způsob předávání informací takovým perifériím jako je tiskárna, mikrodrajv, disk, sp. - všechny tyto informace lze vyslat pomocí jediného příkazu a tím je PRINT.

Kanál vlastně představuje přenosovou linku spojující počítač s některým těchto zařízení. Chceme-li například předat informaci tiskárně, pak je nutné použít PRINT, ale použít k tomu odpovídající "linku" - kanál sloužící ke komunikaci s tiskárnou.

Normálně ve Spectru existují čtyři kanály:

- Kanál "P" PRINTER - komunikace s tiskárnou
- Kanál "S" SCREEN - komunikace s horní částí obrazovky (uživatelské okno)
- Kanál "K" KEYBOARD - slouží k vyslání nebo příjmu informace (např. INPUT, v době vkládání příkazů přímo z klávesnice atd.), využívá se dolní část obrazovky (systémové okno)
- Kanál "R" WORKSPACE - není přístupný z BASICu, slouží ke skladování informací, které pak mohou být přečteny najednou.

Z úrovně BASICu lze využít pouze první tři z uvedených kanálů; nepoužíváme jejich název, ale musíme použít číslo kanálu. Toto číslo je tzv. proud. Přiřazení kanálů a proudů je následující:

Proud	-3, 0, 1	Kanál	K
	-2, 2		S
	3		P
	-1		R

Proudy 4 až 15 jsou po startu systému uzavřeny, t.j. není jim přiřazen žádný kanál.

Všechny proudy se záporným číslem jsou z BASICu nepřístupné, ovšem k využívání kanálů K, S a P stačí proudy 0, 1, 2 a 3.

Příkaz PRINT tiskne v kanále S (proud 2). LPRINT, který tiskne na tiskárně, se od PRINT liší pouze tím, že používá kanál P (přes proud 3). Chceme-li uvnitř příkazu PRINT nebo LPRINT změnit kanál tisku, pak je nutné použít znak # (hash), za kterým následuje číslo námi zvoleného proudu. Napíšeme-li:

```
PRINT #3;"text"
```

pak je výsledek stejný jako LPRINT "text". Slovo se objeví na tiskárně. Kanál tisku lze měnit libovolně několikrát uvnitř jediného příkazu:

```
LPRINT "tisknu na tiskarne";#2;"ted v horni casti obrazovky";#0;"a nakonec v systemovem okne": PAUSE 0
```

Příkaz PAUSE 0 je použit proto, aby nápis v dolní části obrazovky nebyl okamžitě vymazán zprávou "0 O.K.".

Existuje teoretická možnost volby kanálu v instrukci INPUT, ale výběr je omezen na dva proudy: 0 a 1, tedy jen do kanálu K. V ostatních kanálech nelze INPUT použít, protože pouze kanál K může sloužit jako vstupní. Vstup z klávesnice je odesílán do dolní části obrazovky.

Kromě PRINT #, INPUT # a LIST # existují ještě příkazy umožňující otevírat nebo zavírat proudy, které může uživatel používat (proudy 4 až 15). Je to příkaz OPEN #, který proud otevírá a CLOSE #, který jej uzavře.

OPEN # číslo_kanálu, jméno_kanálu otevírá proud uvedeného čísla a přiřazuje mu kanál uvedeného názvu. Název je jednoznakový řetězec "K", "S" nebo "P" (velikost znaků nerozhoduje). Může být samozřejmě použita i řetězcová proměnná o délce 1 a obsahující platný znak (jinak zpráva "Invalid file name").

Chceme-li tedy například otevřít proud 4 pro kanál K, napíšeme:

```
OPEN #4,"K" nebo OPEN #4,"k"
```

a od toho okamžiku PRINT #4; ... tiskne v dolní části obrazovky. Děje se tak do okamžiku, kdy kanál uzavřeme pomocí CLOSE #4. Pak pokus o použití tohoto proudu vede k protestu počítače: "Invalid stream". Při zavírání kanálů je třeba myslet, protože díky chybě v ROM vede pokus o zavření kanálu, který už byl zavřen, nebo nebyl vůbec otevřen, k havárii systému.

V programech, které vůbec nepoužívají strojový kód mohou být kanály použity k tisku nějakých dat (např. jako v následujícím příkladě tabulky a b#) - podle potřeby na obrazovce nebo na tiskárně:

```
10 DIM a(20): DIM b$(20,16)
...
100 OPEN #4,"S": GOSUB 1000
110 OPEN #4,"P": GOSUB 1000
...
1000 REM program tisknouchi pole a() a b#()
1010 PRINT #4;"      Pole a","Pole b#"
1020 FOR n=1 TO 20
1030   PRINT #4;n;TAB 3;a(n),b$(n)
1040 NEXT n
1050 RETURN
```

A to je prakticky vše, k čemu je možné využít kanály na úrovni BASICu. Jejich dokonalejší využití je možné až při použití assembleru a podprogramů uložených v ROM. Neznamená to, že se v BASICu nedá už nic podnikat - podprogramy obsluhy kanálů napsané v assembleru můžeme vložit do příkazů DATA, nebo je natáhnout z pásky pomocí LOAD "" CODE. Provedením několika příkazů POKE "připojovacích" náš podprogram k existujícím kanálům, získáme možnost tento podprogram používat příkazy PRINT, LIST nebo INPUT. O tom si povíme později.

Ve strojovém kódu je tisk poněkud komplikovanější, protože ve strojovém kódu není příkaz, který by byl ekvivalentní příkazu PRINT. Běží-li BASICový program a interpreter narazí na PRINT je provedena celá řada strojových podprogramů. První z těchto podprogramů je CHAN-OPEN sídlící v ROM na adrese 5633 (#1601), který otevírá kanál pro tisk. Otevření zde nemá stejný význam jako v BASICu - tam šlo o přiřazení proudu ke kanálu - zde jde pouze o nalezení a zapamatování si čísla daného kanálu. Před voláním tohoto podprogramu je v akumulátoru umístěno číslo otevíraného kanálu. Hodnota je zapsána v kódu U2 (dvojkový doplněk).

Je-li kanál otevřen, můžeme se věnovat vlastnímu tisku. Základní podprogram vysílající jediný bajt informace v používaném kanále je v ROM na adrese 16 (#10). Podprogram můžeme zavolat pomocí CALL 16 (CALL #10), nebo ještě jednodušeji RST 16 (RST #10). Tento podprogram vyšle na otevřený kanál znak, jehož kód nalezneme v akumulátoru. Takže chceme-li např. v horní části obrazovky vytisknout písmenu "A" můžeme napsat následující program: (nemáte-li v počítači natažený assembler, použijte program, který je uveden dále.)

```
3E FE      LD    A,-2      ; číslo kanálu
CD 01 16   CALL 5633     ; otevření kanálu S
3E 41      LD    A,65     ; kód znaku "A"
C9         RET          ; návrat
```

Nejdříve otevřeme kanál přiřazený proudu 2 - tedy S, pak do něj pošleme kód znaku "A". Chceme-li nyní v kanále S vytisknout další znak, pak nemusíme kanál znovu otevírat. Každý kanál zůstává otevřen do okamžiku, kdy se otevře jiný kanál.

Chceme-li vytisknout delší text, pak není příliš pohodlné vysílat každý znak separátně. V ROM je naštěstí uloženo několik podprogramů, které tento problém řeší. První je na adrese 8252 (#203 PR-STRING). Tiskne text, jehož délka je uložena v registrovém páru BC a adresa prvního znaku v registrovém páru DE. Text může obsahovat libovolné znaky o kódech 0 - 255. Podprogram může pracovat v libovolném kanále, je třeba jen nezapomenout před jeho použitím otevřít požadovaný kanál.

```
3E 02      LD    A,2      ; číslo proudu
CD 01 16   CALL 5633     ; otevření kanálu S
11 72 09   LD    DE,2466  ; adresa řetězce
01 1E 00   LD    BC,30    ; délka
CD 3C 20   CALL 8252     ; tisk textu
C9         RET          ; návrat
```

Druhá procedura tisknoucí texty je na adrese 2082 (#0C0A PD-MSG). To tiskne pouze řetězce znaků s kódy 0 - 127, ale má tu výhodu, že můžeme mít v paměti uloženo za sebou až 256 řetězců a nestarat se ani o adresu ani o délku kteréhokoliv z nich.

V bajtu těsně před prvním řetězcem uložíme hodnotu větší než 127 a za ní text řetězce. Jeho konec označíme tak, že ke kódu posledního znaku přičteme 128 (nastavíme bit 7 na 1, číslo je v konvenci dvojkového doplňku záporné). Hned za tímto bajtem může být začátek dalšího řetězce, zapsaného tímtež způsobem. Chceme-li nyní některý z těchto řetězců vytisknout, vložíme do DE adresu bajtu, který je před prvním znakem prvního řetězce, do akumulátoru vložíme pořadové číslo textu, který se má tisknout. Je to hodnota osmibitová, tedy 0 - 255, kde 0 označuje první řetězec, 1 druhý ... 255 dvěšestpadesátýšestý. V ROMce byste našli několik oblastí textu uloženého tímto způsobem. Vyzkoušejte si jeden z nich - jsou v něm uloženy texty všech chybových zpráv:

```

                                3E 02          LD  A,2          ; číslo proudu
                                CD 01 16       CALL 5633        ; otevření kanálu 5
                                AF            XOR  A          ; znulování registru A
LOOP                            F5            PUSH AF         ; uschování stavu A
                                11 91 13      LD  DE,5009; adresa bajtu před
                                ; prvním textem
                                0D 0A 0C      CALL 2082        ; tisk textu o čísle A
                                3E 0D          LD  A,13         ; přechod na nový řádek
                                D7            RST 16         ; tisk <ENTER> (CR)
                                F1            POP  AF         ; obnovení obsahu A
                                3C            INC  A          ; další text
                                FE 1E         CP   30         ; návrat, byl-li již
                                D0            RET  NC         ; třicátý bajt
                                18 EF         JR   LOOP        ; (-17) opakuj

```

Proanalyzujte činnost tohoto programu a porovnejte vytisklé zprávy se skutečným obsahem oblasti paměti 5009 - 5460 (pomocí PEEK ... a CHR# PEEK ...).

Pokud je v A hodnota 0 až 9, můžete ji vytisknout pomocí CALL 5615 - je to totéž jako ADD A, (kód znaku "0") přes RST 16. Chcete-li vytisknout hodnotu větší než jednocifernou ale menší než 10000, můžete ji uložit do BC a provést podprogram na adrese 6683 (#1A1B OUT-NUM-1).

Pokud nám nestačí ani tento rozsah, pak musíte použít subrutiny tzv. "kalkulátoru". Je to část interpreteru BASICu, která se zabývá všemi druhy výpočtů a k ukládání hodnot používá oblast paměti označovanou jako "CALCULATOR STACK" - zásobník kalkulátoru. Pomocí podprogramů kalkulátoru je možné hodnoty uložené v zásobníku kalkulátoru sčítat, odčítat, násobit, dělit, umocňovat, používat je jako argumenty různých funkcí atd. Je také možné vytisknout hodnotu čísla, které se nachází na vršku zásobníku. Stačí použít pouze dva podprogramy: první - který zapiše hodnotu z registrů mikroprocesoru na vršek zásobníku a druhý, který tuto hodnotu vytiskne.

Abychom zapsali hodnotu na zásobník, uložíme ji do páru BC a zavoláme podprogram na adrese 11563 (#2D28 STACK-BC). Pak otevřeme příslušný kanál a vytiskneme hodnotu zapsanou na zásobník pomocí CALL 11747 (#2DE3 PRINT-FP).

```

01 39 30      LD    BC,12345      ; zápis hodnoty
CD 2B 2D      CALL 11563        ; 12345 na zásobník
                                ; kalkulátoru
3E 02        LD    A,2          ; číslo proudu
CD 01 16      CALL 5633        ; otevření kanálu 5
CD E3 2D      CALL 11747        ; tisk hodnoty z vráku
                                ; zásobníku
C9           RET              ; návrat

```

Tyto podprogramy umožňují vytisknutí prakticky libovolného textu nebo hodnoty a lze je s výhodou používat programy napsanými v assembleru. Na úrovni strojového kódu v podstatě odpovídající příkazu PRINT. Umožňují využívat kanály a jejich studiem se naučíte využívat kanály ke speciálním účelům - např. k vylepšení příkazu PRINT, k zabezpečení programů před vylistováním atp. Následuje výpis jednoduchého programu, který vám umožní spustit mc rutiny, které jsme probrali:

```

1 REM vyzkoušení příkladu v assembleru
10 CLEAR 59999: LET x=60000
20 READ a,b,c,d,e,f
30 DATA 10,11,12,13,14,15
40 READ a#
50 FOR n=1 TO LEN a# STEP 2
60   POKE x,VAL a#(n)*16+VAL a#(n+1)
65   LET x=x+1
70 NEXT n
80 RANDOMIZE USR 60000
85 umístete šestnáctkovy zápis mc programu v řádce 90 DATA .

```

Např. náš první příklad by vypadal následovně:

```
90 DATA "3EFECD01163E41D7C9"
```

Probereme si systém kanálu Spectra a pokusíme se proniknout do tajemství jejich činnosti. Základní instrukci vysílající znaky otevřeným kanálem je RST 16 (RST #10). Využívá ji většina rutin v ROM. Např. rutina na adrese 8252 (#203C) s názvem PR-STRING postupně umísťuje do akumulátoru znaky tisteneho retezce a posílá je přes RST 16.

Jak to, že RST 16 tiskne jednou v kanále S a jindy v kanále K? V čem spočívá otevření kanálu?

V oblasti rezervované pro operační systém Spectra existuje několik oblastí sloužících k obsluze kanálu. K dispozici jsou také určité systémové proměnné a oblast CHANNEL INFORMATION, ležící mezi systémovými proměnnými a BASICem, obvykle na adrese 23734. Pro ujistění, zda tam skutečně je staci zjistit hodnotu systémové proměnné CHANS, v níž je uložena adresa počátku této oblasti.

Zde jsou uloženy adresy rutin obsluhy kanálu. Pro informace pro jednotlivý kanál je určeno 5 bajtů.

ADRESA	HODNOTA BAJTU	
	HEX	DEC
23734	F4	244 \
35	09	9 / 2548
36	A8	168 \
37	10	16 / 4264
38	4B	75 kanál K
39	F4	244 \
23740	09	9 / 2548
41	C4	196 \
42	15	21 / 5572
43	53	83 kanál S
44	81	129 \
45	0F	15 / 3969
46	C4	196 \
47	15	21 / 5572
48	52	82 kanál R
49	F4	244 \
23750	09	9 / 2548
51	C4	196 \
52	15	21 / 5572
53	50	80 kanál P
23754	80	128 konec

OBŘ.1 TABULKA "CHANNEL INFORMATION"

První dva bajty obsahují adresu rutiny obsluhy kanálu v případě, že je používán jako výstupní (OUTPUT), dva následující pro adresu rutiny obsluhující kanál jako vstupní (INPUT). Poslední pátý bajt je kód jednopísmenného názvu kanálu. Pro kanály S, P a R, které jsou pouze výstupními kanály (nemohou být použity pro příjem informací od vnějších zařízení), ukazuje druhá adresa na rutinu generující zprávu "Invalid I/O device".

Za informacemi o všech existujících kanálech se nachází koncový znak - bajt s hodnotou 128 (#80), oddělující CHANNEL INFORMATION od následujícího programu v BASICu.

V době otevření kanálu jsou používány systémové proměnné STRMS a CURCHL. Proměnná STRMS je na adrese 23568 a má délku 38 bajtů. Obsahuje informace o přiřazení kanálu proudům, t.j. informaci o tom, že kanál K můžeme používat přes proudy -3, 0 a 1 a kanál P pouze přes proud 3. Viz obr.2 na příští stránce. Pro každý proud jsou určeny dva bajty používané rutinou otevírající

kanaly (CHAN-OPEN na adrese 5633 [#16011]) jako jedna dvoubajtova hodnota. Je-li rovna 0, pak to znamena, ze proud je uzavren (neni mu prirazen zadny kanal). Pokus otevrit tento kanal skonci zpravou "Invalid stream". Je-li zde hodnota ruzna od nuly, pak je pouzita jako offset pro adresu oteviraneho kanalu v CHANNEL INFORMATION.

Hodnota i prectena ze STRMS ukazuje na prvni bajt oblasti CHANNEL INFORMATION, cili na adresu prectenou v promenne CHANS. Rutina otevirajici kanaly prida k hodnote CHANS hodnotu STRMS zmensenou o 1. Adresa ziskana timto zpusobem je adresou peti bajtu popisujicich prave otevirany kanal. Tato adresa zustane ulozena v systemove promenne CURCHL (CURrent CHaneL - aktualni kanal) na adrese 23633. Na zaklade hodnoty teto promenne identifikuje RST 16 kanal i adresu rutiny jeji obsluhy.

Zkusme neco zmenit ve STRMS: v datech tykajicich se proudu 3 je hodnota 16 (16+256*0), ktera ukazuje na kanal P (23734+16-1)=23749 - viz obr.1. Kdyz tam umistime misto 16 cislo 6, pak bude proud 3 nastaven do kanalu S - LPRINT bude fungovat stejne jako PRINT - tedy vystup nepujde na tiskarnu, ale na obrazovku:

POKE 23580,6: LPRINT "To melo byt na tiskarne."

Napis se na obrazovce skutečne objevi, presteze bylo pouzito prikazu LPRINT.

Je cas pristoupit ke konkretnim prikladum. Podivejme si na proceduru zajistujici vysilani znaku kanalem S, ktera ponekud meni cinnost prikazu PRINT: v case tisku kazdeho znaku (krome mezery a ridicich znaku) je generovan kratky zvuk a nasleduje kratka pauza. Dosahneme tim efektu, jako by tisknul dalnopic.

ADRESA	HODNOTA BAJTU		
	HEX	PROUD	--> KANAL
23568	01		
69	00	-8	--> K
23570	06		
71	00	-2	--> S
72	0B		
73	00	-1	--> R
74	01		
75	00	0	--> K
76	01		
77	00	1	--> K

78	06			
79	00	2	-->	S
23500	10			
81	00	3	-->	P
82	00			
83	00	4	-->	uzavren
84	00			
85	00	5	-->	uzavren
23604	00			
23605	00	15	-->	uzavren

OBR. 2 TABULKA "STRMS"

Jak funguje program z listingu 2? V informacich o kanale S umistime adresu naseho programu. Nyni bude RST 16 v tomto kanale smerovat do naseho programu na adrese 65311 a ne do standardni rutiny v ROM.

Zde si zapamatujeme stav akumulatoru (obsahuje kod znaku, který ma byt vytisten), pak si zkontrolujeme zda pripadne není zablokovano preruseni (instrukce LD A,I s JP PO,65319). Je-li tomu tak, vyhneme se instrukci HALT, která by mela za následek "kousnutí" pocitace (normalne dočasne zastavuje práci pocitace). Dale obnovime obsah akumulatoru a podivame se, co se ma tisknout. Jsou-li to ridici znaky nebo mezera (kody mensi nez 39), pak preskocime instrukce generujici zvuk, ne-li schovame si znovu akumulator na zasobnik i s obsahem IX, nastavime do DE delku zvuku a do HL jeho vysku a pomoci CALL 949 zavolame proceduru BEEP v ROM. V dobe obsluhy kanalu je zakazano menit obsahy alternativnich registru BC' a DE' a registru IX a IY. Rutina BEEP pouziva registr IX, proto si jej musime uložit na zasobnik a na zaver opet jeho stav obnovit. Po BEEP vytiskneme znak

```

10 ;      dalnopsis
20
30 CURCHL EQU 23633
40      ORG 65309
50
60
70 CUR1   DEFW 2548
80
90 START  PUSH AF
100      LD  A,I
110      JP  PO,NOHALT
120      HALT
130      NOP
140 NOHALT POP  AF
150      CP  39
160      JR  C,NOBEEP
170      PUSH AF
180      PUSH IX
190      LD  DE,30
200      LD  HL,150
210      CALL 949
220      POP  IX
230      POP  AF
240 NOBEEP LD  HL,(CUR1)
250      CALL 5676
260      LD  HL,(CURCHL)
270      LD  BC,START
280      LD  E,(HL)
290      LD  (HL),C
300      INC HL
310      LD  D,(HL)
320      LD  (HL),B

```



```

330      LD      A,B      na obrazovku "obvyklou"  rutinou
340      CP      D      v ROM. Nemuzeme to vsak provest
350      JR      NZ,ZMENA obvyklým CALL 2548, ale trochu
360      LD      A,C      oklikou. V informacich o kanale
370      CP      E      S (a take K a P) nejsou vzdy
380      RET     Z      adresy 2548. V dobe tisku ridicich
390 ZMENA LD      (CUR1),DE cich znaku AT a TAB i vsech,
400      RET     ktere se zabývaji atributy zde

```

dochazi k dočasným zmenam. Jde o dva az tri bajty, které se netisknou. První bajt je uložen v proměnné TVDATA a adresa 2548 je změněna na 2669 (AT a TAB) nebo 2695 (INK, PAPER, OVER, BRIGHT atd.). Je-li tisknut znak barvy je až po odeslání druhého bajtu změněn papír, inkoust nebo něco jiného a pak je teprve vrácena "normální" adresa 2548. V případě AT a TAB zůstane druhý bajt také uložen a teprve třetí tiskne znak a vrátí adresu v CHANNEL INFORMATION na 2548.

Aby nedošlo k porušení tohoto mechanismu při výskytu řídicích znaků, musíme v našem programu zohlednit změny, které v této oblasti nastanou. Nejjednodušejší to lze udelat tak, že vždy po zavolání PRINT-OUTPUT v ROM (t.j. od adresy 2548, 2669 nebo 2695) otestujeme adresu nacházející se v informacích kanálu S, kterou tam rutina tisku mohla umístit a jestli je jiná než 65311 (což by znamenalo, že PRINT-OUTPUT ji nezmenila), pak si ji uložíme na adrese 65309. Když následně je RST 16 nasmerována do našeho programu, přečteme odtud adresu volání PRINT-OUTPUT a skočíme tam pomocí CALL 5676. Na tomto místě je instrukce JP (HL), tedy něco na způsob v assembleru neexistující instrukce CALL (HL) - vyvolání podprogramu, jehož adresa je v registru HL.

Proceduru obsluhující kanál si můžete napsat v některém assembleru. Chcete-li ji použít v nějakém vlastním BASICovém programu, pak je nejvýhodnější ji uložit v příkazech DATA a v době činnosti programu ji přenést do RAM pomocí POKE. Pomocí POKE změnit také hodnoty v CHANNEL INFORMATION, nebo obe metody kombinovat. Vyrobíte-li si rutinu strojového kódu, stačí pro její použití provést:

```

CLEAR 65309
LOAD "" CODE 65309
POKE 23739,31
POKE 23740,255

```

Program na obrázku 4 na následující stránce umísťuje rutinu na konec paměti od adresy 65309. V posledním řádku DATA je umístěn kontrolní součet, což by mělo upozornit na eventuelne vzniklé chyby při zápisu. Po umístění rutiny do paměti zbyva ještě ji spustit. Potřebujeme nasmerovat kanál S na tuto rutinu t.j. na adresu 65311, cíli umístit tuto adresu do informací pro kanál S. To udeláme v řádku 53 pomocí:

```
POKE 23739,31: POKE 23740, 255
```

V dalším řádku vyskytující se LIST předvádí jak rutina funguje.

Vsechno funguje spravne, rodinni prislusnici reaguji podrazdene na ramuseni pocitace, kdyz tu ejhle - po CLS nebo ENTER a automaticke vylistovani programu vse zmizi - PRINT a LIST funguje zase normalne. Je to proto, ze CLS vraci do kanalu S (a take K) adresu obsluhy 2548. Je to velmi vyhodne v pripade, ze bychom omylem umistili do CHANNEL INFORMATION nejakou nejakou pomylenou hodnotu, ktera by znemoznila spravnou cinnost techto kanalu v pravde stezejnich. Pak staci jen ENTER a vse je OK.

```
10 CLEAR 65308: RESTORE
20 READ A,B,C,D,E,F
30 DATA 10, 11, 12, 13, 14, 15
40 LET X=65309: RESTORE 200: GOSUB 100
50 POKE 22739,31: POKE 23740,255
90 LIST: STOP
100 LET S=0
110 READ A#: IF A#(1)="." THEN GOTO 170
120 FOR N=1 TO LEN A# STEP 2
130 POKE X,16*VAL A#(N)+VAL A#(N+1)
140 LET S=S+PEEK X
150 LET X=X+1: NEXT N
160 GOTO 110
170 IF S=VAL A#(2 TO ) THEN RETURN
180 PRINT FLASH 1; "OPRAV DATA V RADCICH DATA !!!"
190 STOP
195 REM
200 DATA "F409F5E057E227FF7600F1FE21380FF5DDE5111E00
219600CDB503DDE1F1"
210 DATA "2A1DFFCD2C162A515C011FFF5E7123567079BA2003
798BC8ED531DFFC9"
220 DATA ".7253"
```

OBR. 3 LISTING PROGRAMU "DALNOPIS"

V tomto pripade se nam to ovsem nehodi a musime tuto cinnost eliminovat. Je mozne nasi adresu vracet po kazdem CLS pomoci dvou instrukci POKE ale to neni nejvyhodnejsi reseni. Lepsi je nasledujici metoda.

CLS vraci standardni adresy kanalu S a K. Aby CLS nalezl v CHANNEL INFORMATION adresu kam se tyto hodnoty umisti pouzije proudy - informace o kanale S ziska s kanalu -2 a k nalezeni kanalu K pouzije proud -2. Jestlize otevreme dva kanaly S a jeden priradime proudu -2 a druhy proudu +2, pak do druhého nebude CLS zasahovat. V tomto kanale umistime adresu nasi rutiny.

Proc se kanal, který vytvorime musi jmenovat S ? Vznikly nam dva kanaly a oba maji stejne jmeno.

Kanaly K, S a P jsou obsluhovany spolecnou rutinou PRINT-OUT (na adrese 2548). Ta musi nejakym zpusobem zvedet na který kanal je informace vysilana, jinak receno kam znak vytisknout. K tomu se vyuzivaji nektere bity v urcitych systemovych promennych nastavene odpovidajicim zpusobem procedurou otevirajici kanaly (adr. 5633), v zavislosti na nazvu oteviraneho kanalu.

Bit 1 promenne na adrese 23611 rika zda znak pujde na tiskarnu, nebo na obrazovku (1=tiskarna, kanal P, 0=obrazovka, kanal K nebo S. Bit 0 promenne TV-FLAG (adr. 23612) informuje o tom, zda pouzivame horni nebo dolni cast obrazovky (1=dolni, kanal K, 0=horni, kanal S). Jestlize procedura CHAN-OPEN v dobe otevirani kanalu zjistí jiný název kanalu než K, S nebo P, pak ponechá stav těchto ukazatelů beze změny. Chceme-li tedy, aby náš nový kanál používal rutinu PRINT-OUTPUT k tisku na horní část obrazovky, pak jej musíme nazvat S přesto, že už jeden takový kanál existuje.

Pro informace o tomto novém kanále musíme vytvořit v paměti pět bajtů místa počínaje adresou, na které se nacházel konec oblasti CHANNEL INFORMATION. K tomu můžeme využít rutiny na adrese 5717 (#1655, MAKE-ROOM), která přesouvá obsah paměti počínaje adresou uloženou v HL a končící posledním bajtem prostoru využívaného BASICem o 80 bajtů směrem nahoru a která současně provede odpovídající změny v příslušných systémových proměnných jako PROG, VARS a pod. Dale umístíme do uvolněného místa informace o vytvořeném kanále, tedy adresu výstupu 65811, adresu vstupu 5572 (protože naše rutina, stejně jako standardní v ROM není schopna přijímat informace od vnějších zařízení) a název kanálu, cílí kód znaku 'S'. Na závěr už stačí jen připojit náš kanál na proud 2 pomocí POKE 23758,21 (informace o našem kanále jsou 21, 22 .. 25-ty bajt oblasti CHANNEL INFORMATION).

A v tomto případě už ani CLS nebo ENTER nezruší náš způsob tisku.

Rozdíl od předchozí verze je pouze ve způsobu spuštění procedury v assembleru, takže jen stačí upravit program z obrázku 4 podle obrázku 5 a program znovu spustit pomocí RUN.

```
50 RESTORE 300: LET X=23300: GOSUB 100
60 RANDOMIZE USR 23300
70 RESTORE 350: LET X=23754: GOSUB 100
80 POKE 23758,21
300 DATA "010500210950025516"
310 DATA ".634"
350 DATA "1FFFC41553", ".596"
```

OBR. 5 UPRAVA PROGRAMU Z OBR. 4

Mejte na paměti, že není třeba provádět znovu RANDOMIZE USR 23300, neboť by to mělo za následek vytvoření dalších pěti bajtů (nepotřebných) a přesunutí BASICu nahoru.

A to je prakticky vsechno. Tyto znalosti o kanalech a proudcech staci k tomu, abyste je mohli pouzivat k libovolnemu rozumnemu ucelu. Je treba mit na pameti, ze pri vsech experimentech s kanaly a assemblerem musite byt maximalne opatrní. Drobna chyba muze smaznout celou pamet a dostanete zpravu " 1982 Sinclair Research Ltd", coz je v tomto pripade totez jako jirankovske "Sorry, vole, error!". Radeji nekolikrat si proverte, zda jste se nedopustili chyby. Nevytvarejte v proudcech a kanalech zbytecne chaos, protoze pak se v tom pak nevyznate ani vy, ani Spectrum. Pokud k necemu takovemu dojde, pak nezbyva nez provest v primam rezimu takovou sekvenci, která obnovi normalni hodnoty v prvnich 21 bajtech CHANNEL INFORMATION:

```
FOR N=0 TO 20: POKE 23794+N,PEEK 5551+N: NEXT N
```

a provest

```
RANDOMIZE USR 4720
```

coz navrati normalni stav zevnejsek i promenne STRMS. V tomto pripade se nemusite napisem, který se objeví v dolni casti obrazovky vzrusovat - program, který byl v pameti tam zustal beze zmen a lze jej nadale pouzivat.

TOMASZ SURMACZ
BAJTEK 8/87

TRANSLATE & TASWORD DATLING NOT (C) TOPSOFT KRALUPY/VLT

TRANSLATION & TASWORD DATLING (C) 1988 TOPSOFT KRALUPY

Natištěno pro ZO Svazarm Karolinka - srpen 1989.