

## Z K L A V E S N I C E N A

### J O Y S T I C K

Chcete-li, aby vam Spectrum slouzilo dlouho, k vasi plne spokojenosti musite pecovat o stav klavesnice.

Mechanicke pozkozeni klavesnice nebo pozkozeni "nakreslenych" vodicu pod klavesami je jednou z nejcastejsich zavad Spectra, ktera nebyla uplne odstranena ani v nejnovejsich verzich tohoto pocitace. Zvlaste vyrazne hrozi toto nebezpeci v pripade, ze pocitac je casto pouzivan pri hrani her, ve kterych se obvykle nektere klavesy pouzivaji castěji. Vsak to znate - v zapalu boje casto brzy zapomenete na to, ze klavesy je treba tisknout decentne a cele Spectrum se jen prohyba.

Znamena to snad, ze z nasich sbirek her mame vyradit ty, ktere jsou ovladany vyhradne z klavesnice?

Samozrejme, ze ne. Mnoho z nich muze smele konkurovat hitum poslednich let. K nekterym se radi vracime, i kdyz novejsi hry jsou obvykle lepe graficky zpracovany a maji i lepsi zvuk. K hromadne publikovanym upravam programu pomoci POKE, jako "nesmrtelnost", "nemenne energie" nebo "zastaveni casu", bych chtel prispet svou troskou do mlyna na tema "z klavesnice na joystick".

Otenar, který chce vyuzit mych uvah, bude potrebovat k praci dva programy (jsou obecne rozsireny), a to COPY COPY a MONS3M + GENS3M firmy HiSoft.

Stav jednotlivych portu cte instrukce BASICu IN W. Adresa W, je hodnotou z oblasti 0..65535 a odpovida ruznym vstupnim zarizenim. Zmeny stavu portu muzeme sledovat pomoci programu:

```
10 INPUT "Adresa portu = ";W
20 PRINT IN W;" ";
30 GOTO 20
```

Standardne se pouziva pouze nekolika vybranych adres, coz nam značne ulehci analyzu her. Na dalsi strance nasleduje prehled standardnich portu.

Pokud tyto adresy pouzijeme ve vyse uvedenem programu a neni v odpovidajici rade stisknuta zadna klavesa, pak bude nacten bajt o hodnotě 191, binarne 10111111. (Nebo 31, binarne 00011111 v pripade mikropocitace TIMEX 2048.) Po stisku kterekoliv klavesy teto rady dojde ke znulovani odpovidajiciho bitu (pozice bitu vzrusta ve smeru dovnitr rady). Napriklad po provedeni prikazu PRINT IN 65022 a stisku klavesy F dojde ke znulovani bitu 3 a precteny bajt bude mit hodnotu 187, po soucasnem stisku klaves A a G jsou znulovany bity 0 a 4 a bude ziskan bajt o hodnotě 174. (V pripade pocitace TIMEX 2048 to bude 23 a 14.)

ADRESA PORTU KLAVESNICE		OBSLUHUJE
DESITKOVE	SESTNACTKOVE	KLAVESY
65278	FEFE	CS - V
65022	FOFE	G - A
64510	FBFE	Q - T
64386	F7FE	I - S
61438	EFFE	O - E
57342	DFFE	P - Y
49150	BFFE	ENTER - H
32766	7FFE	SPACE - B

Podobne s pouzitim tehoz programu se muzeme podivat na hodnotu bajtu cteneho z portu joysticku. V pripade nejcasteji pouzivaneho joysticku typu KEMPSTON cte instrukce IN 21 hodnotu 0 je-li knipl v neutralni poloze, po stisku FIRE je posledni bit nastaven na 1 a dale "vpravo" - bit 0, "vlevo" bit 1, atd. Z portu joysticku jsou tedy ziskavany hodnoty 1, 2, atd. Pro joysticky jinych typu jsou jine adresy portu i odcitane hodnoty, coz je treba peclive zjistit pred tim, nez se pustite do uprav programu.

Je zrejme, ze uprava programu napsanych v BASICu nebude obtizna. Najdeme radky testujici stav klavesnice, napr.:

```
230 IF INKEY#="c" THEN GOTO 270
```

Potrebujeme dosahnout stejneho efektu pri poloze kniplu "vpred" - zmenime zapis na:

```
230 IF IN 21=0 THEN GOTO 270
```

nebo ponechame cteni klavesnice v platnosti a pridame:

```
230 IF INKEY#="c" OR IN=8 THEN GOTO 270
```

Bohuzel, zajimavych her napsanych v BASICu je malo (napr. MINED OUT). Prevažna vetsina je napsana ve strojovem kodu, který zajistuje vetsi dynamiku hry i mnohem kvalitnejsi vyuziti grafiky.

Proto nez zacneme hovorit o konkretnich podprogramech prechodu "z klavesnice na joystick" musime se jeste zastavit u zpusobu testovani stavu klavesnice pouzivanych strojovym kodem.

Nejcasteji jsou to instrukce 280, které ctou stav klavesnice pres standardni porty:

MNEMONIKA	KOD INSTRUKCE SESTNACTKOVE	ADRESA PORTU V REGISTRU		HODNOTA PRECTENA Z PORTU JE V REGISTRU
		SB	MD	
IN A, (dd)	DB dd	A	dd	A
IN A, (C)	ED 78	B	C	A
IN H, (C)	ED 60	B	C	H
IN L, (C)	ED 68	B	C	L
IN B, (C)	ED 40	B	C	B
IN C, (C)	ED 48	B	C	C
IN D, (C)	ED 50	B	C	D
IN E, (C)	ED 58	B	C	E

Ekvivalentem prikazu BASICu:

1) LET W=IN 65022                      muze byt

LD A,#FD                      LD BC,#FD

nebo

IN A,(#FE)                      IN A,(C)  
(hodnota precteneho bajtu bude v akumulatoru)

2) LET W=IN 21                      muze byt

XOR A                      LD BC,#0015

nebo

IN A,(#15)                      IN B,(C)  
(W v akumulatoru)      (W v registru B)

Jinak jsou take vyuzivany systemove promenne, ktere obsahuji ASCII kody odpovidajici danym klavesam. Na adrese 23556 (#5C04) najdeme pomoci PEEK kod aktualne stiskle klavesy v modu C. Obdobne muzeme ziskat kod posledni stiskle klavesy - je ulozen v systemove promenne LAST-K na adrese 23560 (#5C08). Obvykle to jsou kody klaves v modu L, ale soucasny stisk nektereho ze shiftu ma ze nasledek zmenu hodnoty promenne LAST-K.

Registry par IY standardne obsahuje bazovou adresu systemovych promennych 23610 (#5C3A). Diky tomu muzeme hodnotu kodu z promenne LAST-K zavest do akumulatoru bud instrukci LD A,(23560) nebo v rezimu indexoveho adresovani LD A,(IY-50).

Dalsi moznosti muze byt volani subrutiny KEY-SCAN v ROM, ktera je na adrese 654 (#028E). Rutina uklada kod stiskle klavesy do registroveho paru DE, pricemz bude registr D obsahovat hodnotu 0..39, a registr E hodnotu specifikujici s jakym shiftem byla klavesa stisknuta. Pri soucasnem stisku nekolika klaves je vynu- lovana vlajka Z (nula). Naproti tomu neni-li stisknuta zadna klavesa, je v registrovem paru DE hodnota #FEFE a vlajka Z je nastavena na 1.

Pro zjistení hodnoty v registrech D a E a stavu vlajky Z je možno použít následující program.

```
20 LET s=0: FOR i=0 TO 18: READ a: POKE 30003+i,a: NEXT i:
  IF s<>1447 THEN PRINT "Oprav DATA!": STOP
30 PRINT: PRINT INKEY#
40 RANDOMIZE USR 30002
50 PRINT "D=";PEEK 30001,"E=";PEEK 30000
60 PRINT "Vlajka Z=";PEEK 30002
70 PAUSE 100: POKE 23692,5: GOTO 30
80 DATA 205,142,2,237,83,49,117,40,4
90 DATA 62,0,24,2,61,1,50,50,117,201
```

Program volá rutinu KEY-SCAN v ROM a ukládá obsahy registru D a E a vlajky Z do pametových bunek 30000..30002.

Je třeba říci, že jsou to nejčastěji používané rutiny ve strojovém kódu. Je nutné vyhledat ve strojovém kódu tu část, která se zabývá "čtením klávesnice" a její rozbor patří k nejobtížnějším etapám práce.

Doporučuji následující postup:

- 1) Zjistit startovní adresy a délky jednotlivých bloků programu (analýzou LOADERU nebo programy COPY COPY, Q-COPY, TASMAG, tape apod.)
- 2) Seznamit se s činností jednotlivých kláves užívaných během hry. Rozhodnout, které z činností budeme simulovat pomocí joysticku (pohybem kniplotu, stiskem FIRE, eventuelně současným výkonem obou činností).
- 3) Zjistit zda se bloky programu realizujícího hru nespouštějí automaticky. Jsou-li bloky příliš dlouhé je možné je pomocí COPY COPY rozdělit.
- 4) Nahrát MONSSM a do volné oblasti RAM bloků programu. Spustit monitor a nastavit paměťový ukazatel na začátek bloku.
- 5) Příkazem G najít sekvence instrukcí ctoucích stav klávesnice např.:  
G:DB:FE G:ED:78 G:8:5C atp.  
a zapsat si adresy těchto instrukcí.
- 6) Nastavit paměťový ukazatel na první z nalezených adres. Je-li to nutné - posunout paměťový ukazatel tak, aby bylo možné přesně určit významnější bajt adresy portu i registru do kterého se bude číst, aniž by se při tom změnila desasemblovací program. Pochopit činnost desasemblované části strojového kódu a pečlivě analyzovat činnost programu odcitávajícího stav klávesnice.
- 7) Prohlédnout, je-li to nutné, zbyvající bloky programu podle bodu 4-6.

8) Najit volne misto v RAM a zapsat sem podprogram obsluhy joysticku bez vyloucení původního řízení z klavesnice v duchu následujícího vzoru (napr. funkci klavesy "KL" simulujeme pohybem kniplu "vpravo"):

```

                ORG    NN                ;program zacina na adrese NN
                XOR    A                ;cte do akumulatoru
                IN     A, (#15)         ;hodnotu portu z joysticku
B1             BIT    0,A              ;testuje zda je knipl vychylen
                ;"vpravo" (zda je bit 0 roven
                ;jedna)
                JR     Z,B2            ;skok do dalsi casti rutiny,
                ;podminka neni splena,
                ;bit 0 je nulovy
                LD     A,dd            ;do A hodnota kodu, ktera byla
                ;prenesena procedurou v hlavnim
                ;programu po stisku klavesy KL
                RET                    ;navrat do hlavni rutiny a pro-
                ;vedeni podprogramu obsluhy
                ;klavesy KL
B2             BIT    1,A              ;je-li knipl vychylen "vlevo"
                ;(zacatek obsluhy dalsi
                ;klavesy)
                "
                "
                "
C              ;radky hlavniho programu, ctou-
                ;ciho stav klavesnice, nahraze-
                ;ne instrukci CALL NN
                RET                    ;navrat do hlavniho programu

```

Pri zapisu teto rutiny pouzijte GEN88M nebo jiny assembler. Generujte strojovy kod a nahrajte jej na pasek.

9) V analyzovanem bloku nahradte vsechny sekvence cteni stav klavesnice instrukci CALL NN. Stav klavesnice muze byt ve hre testovan na nekolika ruznych mistech programu.

10) Zamontujte rutiny obsluhy joysticku do upraveného programu pomoci COPY COPY nebo z BASICu prikazem POKE. Nesmi samozrejme kolidovat s oblastí hlavního programu a musí byt nacteny do pameti drive, nez program odstartuje.

11) Overte funkcnost upraveného programu.

Zkusme nyní vyuzit dosud uvedenyh faktů k uprave velmi pekneho programu "POOL", ktery v roce 1988 napsal Mike LAMB. Tento program se sklada ze dvou casti:

- prvni, ktera je soucasne zavadecim programem, obsahuje jednak cast strojoveho kodu (radky 1-130) a BASIC (140-1140) a tato cast je dlouha 6975 bajtu
- druha cast (bajty bez hlavicky) ma delku 1536 bajtu a je ukladana od adresy 30976 procedurou, ktera je v prvni casti programu ma adrese #65A7.

Tyto udaje nam tedy rikaji, ze cely program POOL je zavaden na adresy 22755 az 32512.

Program obsluhy joysticku bude zalozen na nasledujici logice:

- cinnost klaves "s" a "a" (pohyb zamerovace) budeme simulovat pohybem kniplu "doleva" a "doprava"
- cinnost klavesy "I" (sila uderu) pohybem kniplu "vpred" a klavesa ENTER (pal) stiskem "FIRE".

Ostatni klavesy ("1", "2", nebo "ENTER" po "GAME OVER" ve volbe 2), pouzivane behem hry zridka, ponechame beze zmeny.

Prvni segment nahrany prikazem LOAD se odstartuje po stisku BREAK. Pri sledovani jeho efektu obratime svou pozornost na nastavovani zamerovace (zvukovy efekt) nebo provedeni uderu. Privadi nas to na myslenku, ze funkce cteni klaves je treba hledat v teto casti programu. V radcich BASICu nejsou, tam jsou jen prikazy tykajici se funkci klaves, které menit nechceme. K prozkoumani tedy zbyvaji radky 1 az 130, které obsahují strojový kód. Skutecne pomocí programu MONS a jeho příkazu G (G: DB: FE nebo G: FD: 7E: CE) najdeme část programu, která nás zajímá:

```
#6460 LD    A,#FD          ;cteni stavu
      IN    A,(#FE)       ;rady klaves A-G
      LD    HL,(#7A0E)    ;
      RRA                ;presun bitu 0 (stav "a"
                        ;do CARRY)
      JR    C,#646D       ;skok, neni-li "a"
                        ;stisknuto

      DEC   HL
      JR    #64E7         ;skok na obsluhu "a"
#646D RRA                ;presouva bit 1 (stav "s"
                        ;do CARRY)
      JR    NC,#64E6      ;skok na obsluhu "s"
#6470 LD    A,(IY+50)     ;cte do A ASCII kod
                        ;stiskle klavesy
      CP    #6C           ;jde o klavesu "1" ?
      EI
      JR    NZ,#64BA      ;skok neni-li to "1"
#64BA CP    #0D           ;jde o klavesu ENTER ?
      JR    NZ,#64DE      ;skok, neni-li ENTER
```

Zdurazname jeste jednou, ze v programu dochazi ke cteni stavu klavesnice bud testovanim bitu (klavesy "a" a "s"), nebo kodu ASCII (klavesy "1" a "ENTER"). Na to musime brat zreteľ pri vytvareni programu pro obsluhu joysticku.

Zjistili jsme, ze puvodni program POOL konci na adrese 32512 a proto nas program umistime na adrese 33000 (#80E8):

```
ORG    33000             ;pohyb kniplem "doleva",
                        ;"doprava"
XOR    A                 ;cteni stavu
IN     A,(#15)           ;portu joysticku
BIT    0,A               ;test zda "doprava"
```

```

        JR      Z,E1          ;ne-li, jdi na E1
        LD      A,#BD        ;do akumulatoru 189, (binarne
                               ;1011 1101), simuluje stisk
                               ;klavesy "s"
E1:     RET
        BIT     1,A          ;navrat do programu
        JR      Z,E2        ;test zda "doleva"
                               ;a jestli ano, tak simuluje
                               ;stisk klavesy "a"
E2:     RET
        LD      A,#FD        ;neni-li pouzit joystick, pak
                               ;podprogram
        IN      A,(#FE)      ;prectete stav klaves "a" a "s"
        RET
        ORG    33050        ;a vrati se do hlavniho prog.
        XOR    A            ;pohyb kniplu "vpred" a FIRE
        IN      A,(#15)
        BIT     3,A          ;knipl "vpred" ?
        JR      Z,F1
        PUSH   BC
        LD      BC,5000      ;smycka zpozdujici provedeni
A1:     DEC     BC          ;procedurey obsluhujici
                               ;klavesy "1"
        LD      A,B
        OR     C
        JR      NZ,A1
        POP    BC
        LD      A,#6C        ;simuluje stisk klavesy "1"
                               ;ASCII kod je 108
F1:     RET
        BIT     4,A          ;bylo FIRE ?
        JR      Z,F2        ;ano, simuluj stisk "ENTER"
                               ;ASCII kod je 13
F2:     RET
        LD      A,(IY-50)    ;neni-li pouzit joystick, pre-
                               ;ctete podprogram stav klaves "1"
                               ;a "ENTER" a vraci se do
                               ;hlavniho programu
        RET

```

Program napíšeme pomocí vhodného assembleru (např. GENS 3M) a na pásek jej nahrajeme příkazem SAVE "POOL" CODE 33000,85

Dále musíme provést změny v první části hlavního programu. Změny způsobují ve vepsání instrukcí CALL v místě instrukcí čtení stavu klávesnice. Pomocí MONSU zapíšeme do paměťových buněk 6460-6462 instrukci CALL 33000 (#CD, #E8, #80), a do bajtu 6463, který zůstává po parametru instrukce IN A,(#FE), instrukci NOP (#00). Stejně tak do bajtu 6470-6472 napíšeme CALL 33050 (#CD, #1A, #80). Upravený segment nahrajeme na pásek příkazem SAVE "POOL" LINE 170.

Nyní napíšeme vlastní zaváděcí program např.:

```
10 LOAD ""CODE: LOAD ""
```

(může eventuelně obsahovat popis funkce joysticku a další poznámky), nahrajeme jej na pásku a nakonec pomocí vhodného

kopiraku sestavime vyslednou sekvenci bloku predstavujicich opravenou verzi programu POOL, ktera se sklada z naseho zavadeciho programu, bloku obsluhy joysticku, opraveneho prvnio bloku a dat bez hlavicky.

Uvedene zmeny muzeme provest i jinak. Pomoci programu COPY-COPY, provedeme v prvni segmentu nasledujicimi "pouky":

25254,205	25255,232	25256,128
25257,0	25270,205	25271,26
25272,129		

a napiseme nasledujici zavadeci program:

```
10 LET S=0
   FOR I=33000 TO 33021
     READ A
     POKE I,A
     LET S=S+A
   NEXT I
20 FOR I=33050 TO 33080
   READ A
   POKE I,A
   LET S=S+A
   NEXT I
30 IF SK>6561 THEN PRINT "UPRAV DATA"
   STOP
40 LOAD ""
50 DATA 175,219,21,203,71,40,3,62,189,201,203
60 DATA 79,40,3,62,190,201,62,253,219,254,201
70 DATA 175,219,21,203,95,40,13,197,1,0
80 DATA 80,11,120,177,32,251,193,62,108,201
90 DATA 203,103,40,3,62,13,201,253,126,206,201
```

Nahrajeme opraveny program POOL a pokud jsme nespachali zadnou chybu, muzeme si kulecnik zahrat pomoci joysticku.

Na zaver uvedu upravy nekolika dalsich programu, ktere ziskali popularitu. Dukladne si je prostudujte, porovnejte upravene verze s originalnimi programy a ziskate tim dostatek zkusenosti pro upravy dalsich programu.

## >>> JUMPING JACK <<<

-----

1. Zavadeci program:

```
10 LET S=0
   FOR I=33000 TO 33022
     READ A
     POKE I,A
     LET S=S+A
   NEXT I
```



```

20 FOR I=33050 TO 33065
  READ A
  POKE I,A
  LET S=S+A
NEXT I
30 IF S<>4698 THEN PRINT "OPRAV DATA !"
  STOP

40 LOAD ""
50 DATA 175,219,21,208,71,40,3,62,30,201,203,79
60 DATA 40,2,62,29,201,1,254,127,237,120,201
70 DATA 175,219,21,208,95,40,3,62,30,201
80 DATA 1,254,254,237,120,201

```

2. POKE do segmentu (bajty bez hlavicky) o delce 8818 provedeme programem COPY-COPY:

```

29700,205          29701,26          29702,129
29703,0           29704,0           29714,205
29715,232        29716,128        29717,0
29718,0

```

>>> SKIING <<<

Misto prvnich dvou BASICovych segmentu napiseme vlastni zavadecci program:

```

10 CLEAR 24575
  LET S=0
  FOR I=33000 TO 33034
    READ A
    POKE I,A
    LET S=S+A
  NEXT I
20 IF S<>3674 THEN PRINT "UPRAV DATA"
  STOP

30 LOAD "" SCREEN$
  LOAD "" CODE
40 FOR I=1 TO 4
  READ A
  POKE A,205
  POKE A+1,232
  POKE A+2,128
NEXT I
50 RANDOMIZE USR 24576
60 DATA 175,219,21,208,71,40,3,62,23,201
70 DATA 208,79,40,3,62,15,201,203,87,40,3,62,27,201
80 DATA 203,95,40,3,62,29,201,123,219,254,201
90 DATA 23393,23423,23432,23471

```

>>> GULFMAN <<<

Do zavadečního programu dopiseme následující řádky:

```

30 GOSUB 80
80 LET S=0
   FOR I=33000 TO 33041
     READ A
     POKE I,A
     LET S=S+A
   NEXT I
90 IF SK>4029 THEN PRINT "OPRAV DATA"
   STOP
100 POKE 25932,205
    POKE 25933,232
    POKE 25934,128
110 RETURN
120 DATA 175,219,21,203,103,40,3,62,57,201,203,71,
    40,3,62,56
130 DATA 201,203,79,40,3,62,53,201,203,87,40,3,62,
    54,201
140 DATA 203,95,40,3,62,55,201,58,8,92,201

```

>>> CHECQUERED FLAG <<<

1. Program o délce 38 bajtu, který hru načítá upravíme následujícím způsobem:

```

10 LET S=0
   FOR I=64000 TO 64087
     READ A
     POKE I,A
     LET S=S+A
   NEXT I
20 IF SK>9716 THEN PRINT "UPRAV DATA"
   STOP
30 LOAD "*"
40 DATA 245,62,253,184,32,43,175,219,21,203,103
50 DATA 40,18,203,71,40,5,241,62,23,24,64
60 DATA 203,79,40,57,241,62,30,24,55,203,71
70 DATA 40,5,241,62,27,24,46,203,79,40,39
80 DATA 241,62,29,24,37,62,239,184,32,12,175
90 DATA 219,21,203,95,40,22,241,62,30,24,20
100 DATA 62,223,184,32,12,175,219,21,203,87,40
110 DATA 5,241,62,30,24,3,241,237,120,19,201

```

2. Následují "pouky" do čtvrtého segmentu o délce 38400 bajtu pomocí programu COPY-COPY:

54545,205	54546,0	54547,250
-----------	---------	-----------

Čtenář určitě zjistí, že při těchto úpravách je v přidávaných podprogramech získány kód znovu testován hlavním programem. Tato nevýhoda je kompenzována univerzálností použití a v praxi nezpůsobuje snížení rychlosti programu.

TRANSLATE & TASWORD DATLING  
ADAM GAWLOWSKI, BAJTEK 5/88  
NOT (C)1988 TOPSOFT KRALUPY