

## Monitor pocitaca MATO

```

1/  0 : ;-----
2/  0 : ; Tento zdrojovy kod a vypis vznikol disassemblovanim obsahov ROM pocitaca MATO.
3/  0 : ; Pri komentovani kodu pomohli disassemblované vypisy od RaceSoft, Martin M1
4/  0 : ; a Komentovany vypis Monitora PMD 85-2 od Vita Libovickeho a Jiriho Olmera.
5/  0 : ; Vaccina mien navesti pisanych velkymi pismenami bola prevzata z Komentovaneho
6/  0 : ; vypisu Monitora PMD 85-2.
7/  0 : ; Ostatne povodne nepomenovane navestia su v tvare Camel-Case.
8/  0 :
9/  0 : ; V texte su pouzivane rozne skratky:
10/ 0 : ; DR - dialogovy riadok
11/ 0 : ; VRAM - Video RAM
12/ 0 : ; uR - mikroriadok
13/ 0 : ; zR - znakovy riadok
14/ 0 : ; MGF - magnetofon
15/ 0 :
16/ 0 : ; RomBor 12/2025
17/ 0 : ;-----
18/ 0 :
19/ 0 : ;-----
20/ 0 :
21/ 0 :         page      0
22/ 0 :         listing PURECODE
23/ 0 :         cpu       8080
24/ 0 :
25/ 0 : ;-----
26/ 0 : ; Kontrola hodnoty symbolu MVER, urcujuceho verziu kompilovaneho monitora.
27/ 0 : ; MVER=0 : mato-mb.rom
28/ 0 : ; MVER=1 : mato-lan.rom
29/ 0 : ; MVER=2 : mato-mb-mtst.rom
30/ 0 : ; MVER=3 : mato-mb-ru.rom
31/ 0 :
38/ 0 :
39/ 0 :         ; *****
43/ 0 :         ; * mato-lan.rom : MATO Monitor + BASIC + LAN *
49/ 0 :         ; *****
50/ 0 :
51/ 0 : ;-----
52/ 0 :
53/ 0 :         title     "Monitor pocitaca MATO"
54/ 0 :
66/ 0 :
67/ 0 : ;-----
68/ 0 : =8000H Begin          equ      8000h          ; Monitor zacina na adrese 8000h

```

```

69/  0 : =8000H      Stack      equ      8000h      ; zasobnik na koniec RAM
70/  0 : =0C000H     VRAM       equ      0C000h     ; adresa Video RAM
71/  0 :
72/  0 : =9800H      Basic      equ      9800h      ; adresa interpretera v ROM
73/  0 : =2400H      BasicLen   equ      2400h      ; dlzka BASICu
74/  0 : =0BCH       BasicEndHI equ      (Basic+BasicLen)>>8 ; vyssi byte konca BASICu
75/  0 : =400H       DemoLen    equ      0400h      ; dlzka DEMO programu
76/  0 : =0C0H       DemoEndHI  equ      (Basic+BasicLen+DemoLen)>>8 ; vyssi byte konca programu
77/  0 :
78/  0 : =2400H      BasPrgAdr   equ      2400h      ; adresa zaciatku Programu v BASICu
79/  0 : =27FDH      DemoPrgEnd  equ      27FDh      ; adresa konca DEMO programu v RAM
80/  0 : =5E7AH      BasPrgEnd   equ      5E7Ah      ; syst. premenna BASICu - koniec programu
81/  0 :
82/  0 :             ;-----
83/  0 :             ; porty Systemovej 8255
84/  0 : =0F4H       SYS55A     equ      0F4h      ; port A
85/  0 : =0F5H       SYS55B     equ      0F5h      ; port B
86/  0 : =0F6H       SYS55C     equ      0F6h      ; port C
87/  0 : =0F7H       SYS55M     equ      0F7h      ; nastaveni rezimu
88/  0 : =8AH        SYSCTRL    equ      8Ah       ; rezim: PA:OUT, PB:IN, PCH:IN, PCL:OUT
89/  0 :
90/  0 : =1H         X_MGF_OUT   equ      1         ; vystup na MGF
91/  0 : =2H         X_AM       equ      2         ; vystup na akusticky menic
92/  0 : =4H         X_RFU      equ      4         ; nepouzity vystup
93/  0 : =8H         X_POL      equ      8         ; nastavenie polarity signalu MGF
94/  0 : =10H        X_STOP     equ      16        ; vstup klavesu STOP - 1/0 - ne/stlaceny
95/  0 : =20H        X_SHF      equ      32        ; vstup klavesu SHF - 1/0 - ne/stlaceny
96/  0 : =40H        X_CNT      equ      64        ; vstup klavesu CNT - 1/0 - ne/stlaceny
97/  0 : =80H        X_MGF_IN   equ      128       ; vstup z MGF
98/  0 :
99/  0 :             ; porty 8255 v ROM Module
100/ 0 : =0F8H       ROM55A     equ      0F8h      ; citane data
101/ 0 : =0F9H       ROM55B     equ      0F9h      ; nizsi byte adresy
102/ 0 : =0FAH       ROM55C     equ      0FAh      ; vyssi byte adresy, aktivacia
103/ 0 : =0FBH       ROM55M     equ      0FBh      ; nastavenie rezimu
104/ 0 : =90H        ROMCTRL    equ      90h       ; rezim: PA:IN, PB:OUT, PC:OUT
105/ 0 :
106/ 0 :             ; kody pre vystup na obrazovku
107/ 0 : =8H         BS         equ      08h      ; backspace
108/ 0 : =0AH        LF         equ      0Ah      ; ignorovany kod
109/ 0 : =0DH        CR         equ      0Dh      ; koniec riadku/retazca
110/ 0 : =1CH        CLS        equ      1Ch      ; zmazanie obrazovky
111/ 0 :
112/ 0 :             ;=====
113/ 0 :             ; Kody riadiacich klavesov
114/ 0 :             ;=====
115/ 0 : =1H         KDELL      equ      01h      ; CNT + .

```

116/	0 : =3H	KENDL	equ	03h	; CNT + -->
117/	0 : =8H	KLEFT	equ	08h	; <--
118/	0 : =9H	KSUP	equ	09h	; SHF + sipka hore
119/	0 : =0AH	KINVR	equ	0Ah	; CNT + sipka dole
120/	0 : =0BH	KWRK	equ	0Bh	; CNT + Y
121/	0 : =0CH	KBGNL	equ	0Ch	; CNT + sipka hore
122/	0 : =0DH	KEOL	equ	0Dh	; EOL
123/	0 : =0FH	KPTL	equ	0Fh	; CNT + /
124/	0 : =10H	KBEEP	equ	10h	; CNT + <--
125/	0 : =11H	KUP	equ	11h	; sipka hore
126/	0 : =12H	KMON	equ	12h	; CNT + ,
127/	0 : =13H	KDOWN	equ	13h	; sipka dole
128/	0 : =14H	KSDOWN	equ	14h	; SHF + sipka dole
129/	0 : =17H	KRCL	equ	17h	; CNT + X
130/	0 : =18H	KRIGHT	equ	18h	; -->
131/	0 : =19H	KSRIGHT	equ	19h	; SHF + -->
132/	0 : =1AH	KSLEFT	equ	1Ah	; SHF + <--
133/	0 : =1BH	KCLL	equ	1Bh	; CNT + B
134/	0 : =1CH	KINST	equ	1Ch	; CNT + M
135/	0 : =1DH	KDELT	equ	1Dh	; CNT + N
136/	0 : =1EH	KCD	equ	1Eh	; CNT + C
137/	0 : =1FH	KCLS	equ	1Fh	; CNT + V
138/	0 :				
139/	0 :	; semigraficke znaky			
140/	0 : =80H	KSG0	equ	80h	; CNT + Q
141/	0 : =81H	KSG1	equ	81h	; CNT + W
142/	0 : =82H	KSG2	equ	82h	; CNT + E
143/	0 : =83H	KSG3	equ	83h	; CNT + R
144/	0 : =84H	KSG4	equ	84h	; CNT + T
145/	0 : =85H	KSG5	equ	85h	; CNT + Z
146/	0 : =86H	KSG6	equ	86h	; CNT + U
147/	0 : =87H	KSG7	equ	87h	; CNT + I
148/	0 : =88H	KSG8	equ	88h	; CNT + O
149/	0 : =89H	KSG9	equ	89h	; CNT + P
150/	0 : =8AH	KSG10	equ	8Ah	; CNT + @
151/	0 : =8BH	KSG11	equ	8Bh	; CNT + A
152/	0 : =8CH	KSG12	equ	8Ch	; CNT + S
153/	0 : =8DH	KSG13	equ	8Dh	; CNT + D
154/	0 : =8EH	KSG14	equ	8Eh	; CNT + F
155/	0 : =8FH	KSG15	equ	8Fh	; CNT + G
156/	0 : =90H	KSG16	equ	90h	; CNT + H
157/	0 : =91H	KSG17	equ	91h	; CNT + J
158/	0 : =92H	KSG18	equ	92h	; CNT + K
159/	0 : =93H	KSG19	equ	93h	; CNT + L
160/	0 : =94H	KSG20	equ	94h	; CNT + ;
161/	0 : =95H	KSG21	equ	95h	; CNT + :
162/	0 :				

```

163/ 0 : ; programove (klucove) klavesy
164/ 0 : =0D0H KK0 equ 0D0h ; CNT + 1
165/ 0 : =0D1H KK1 equ 0D1h ; CNT + 2
166/ 0 : =0D2H KK2 equ 0D2h ; CNT + 3
167/ 0 : =0D3H KK3 equ 0D3h ; CNT + 4
168/ 0 : =0D4H KK4 equ 0D4h ; CNT + 5
169/ 0 : =0D5H KK5 equ 0D5h ; CNT + 6
170/ 0 : =0D6H KK6 equ 0D6h ; CNT + 7
171/ 0 : =0D7H KK7 equ 0D7h ; CNT + 8
172/ 0 : =0D8H KK8 equ 0D8h ; CNT + 9
173/ 0 : =0D9H KK9 equ 0D9h ; CNT + 0
174/ 0 : =0DAH KK10 equ 0DAh ; CNT + -
175/ 0 : =0DBH KK11 equ 0DBh ; CNT + \
176/ 0 :
177/ 0 : =87H XRCL equ 87h ; kod kluca RCL
178/ 0 :
179/ 0 : ;=====
180/ 0 : ; Systemove premenne vedla VRAM
181/ 0 : ;=====
182/ 0 : =0C030H SYSVAR equ 0C030h ; zaciatok systemovych premennych
183/ 0 : =0C030H BUF equ 0C030h ; zaciatok buffra dialogoveho riadku
184/ 0 : =0C032H DIAL equ 0C032h ; adr. zac. vypisu dial. riadku na obrazovke
185/ 0 : =0C034H DLZRAD equ 0C034h ; koniec buffra dial. riadku
186/ 0 : =0C036H DCUR equ 0C036h ; pozicia kurzora v buffri
187/ 0 : =0C038H RPOS equ 0C038h ; prvý vypisovany znak z buffra na obrazovku
188/ 0 : =0C03AH COLOR equ 0C03Ah ; kod farvy
189/ 0 : =0C03BH BEEPPOS equ 0C03Bh ; pozicia v buffri pre pipnutie
190/ 0 : =0C03CH TAS00 equ 0C03Ch ; byvale TASCII
191/ 0 : =0C03EH CURSOR equ 0C03Eh ; adresa kurzora na obrazovke
192/ 0 :
193/ 0 : =0C070H SPVAL equ 0C070h ; adresa vrcholu zasobnika
194/ 0 : =0C072H CURCH equ 0C072h ; adresa nastavenia pri analyze buffra
195/ 0 : =0C074H MESS equ 0C074h ; adresa zaciatku textu pre vypis
196/ 0 : =0C076H RETCALL equ 0C076h ; navratova adresa pre ENTER
197/ 0 : =0C078H ODLOZ equ 0C078h ; adresa pre ulozenie textu po EOL v ENTER
198/ 0 : =0C07AH RETSP equ 0C07Ah ; adresa zasobnika pre navrat z ENTER
199/ 0 : =0C07CH SPEED equ 0C07Ch ; konstanta trvania 1 MGF pulzu
200/ 0 : =0C07EH ITABDIR equ 0C07Eh ; zaciatok tabulky prikazov Monitora
201/ 0 :
202/ 0 : =0C0B0H TASCII equ 0C0B0h ; tabulka adries predlohy znakov - nevyuzite
203/ 0 : =0C0BCH KeyLast equ 0C0BCh ; kod naposledy stlaceneho klavesu
204/ 0 : =0C0BDH KeyRepeat equ 0C0BDh ; zdrzanie pri opakovaní klavesu
205/ 0 : =0C0BFH SBEEP equ 0C0BFh ; priznak povolenia pipnutia
206/ 0 :
207/ 0 : =0C0F0H RAM equ 0C0F0h ; navratovy vektor po MGSV, MGLD a MGEND
208/ 0 : =0C0F4H IIROLL equ 0C0F4h ; pocet scrollovanych uR
209/ 0 : =0C0F5H RSIRRAD equ 0C0F5h ; vyska zR v uR

```

210/	0 : =0C0F6H	POROLL	equ	0C0F6h	; adresa posledneho zR
211/	0 : =0C0F8H	CURROL	equ	0C0F8h	; adresa pre vypis dalsieho riadku
212/	0 : =0C0FAH	VYSKA	equ	0C0FAh	; vyssi byte VRAM posledneho riadku
213/	0 : =0C0FBH	ENLNW	equ	0C0FBh	; pocet viditelnych znakov na riadok
214/	0 : =0C0FCH	RADSIR	equ	0C0FCh	; offset na dalsi zR
215/	0 : =0C0FEH	DLZR	equ	0C0FEh	; max. pocet zobrazonych znakov v riadku
216/	0 : =0C0FFH	DLZP	equ	0C0FFh	; max. pocet vysunutych znakov v riadku
217/	0 :				
218/	0 : =0C130H	BEEDT	equ	0C130h	; adresa tabulky pre BEEP
219/	0 : =0C132H	KDIR	equ	0C132h	; adresa tab. adres rutin pre kody 00h..1Fh
220/	0 : =0C134H	ASCII	equ	0C134h	; znak z klavesnice
221/	0 : =0C135H	SWRK	equ	0C135h	; priznak WRK
222/	0 : =0C136H	MGCHAR	equ	0C136h	; typ suboru
223/	0 : =0C137H	SPTL	equ	0C137h	; priznak PTL
224/	0 : =0C138H	MONX	equ	0C138h	; navratova adresa pre JUMP
225/	0 : =0C13AH	CDBUF	equ	0C13Ah	; adresa C-D buffer (text hlasenia)
226/	0 : =0C13CH	LSTR	equ	0C13Ch	; dlzka textu v DR
227/	0 : =0C13EH	STACK	equ	0C13Eh	; odlozenie adresy zasobnika
228/	0 :				
229/	0 : =0C170H	X1	equ	0C170h	; suradnica X pociatocneho bodu
230/	0 : =0C172H	Y1	equ	0C172h	; suradnica Y pociatocneho bodu
231/	0 : =0C173H	X2	equ	0C173h	; suradnica X koncového bodu
232/	0 : =0C174H	Y2	equ	0C174h	; suradnica Y koncového bodu
233/	0 : =0C175H	DCUROL	equ	0C175h	;
234/	0 : =0C177H	DELKA	equ	0C177h	; nepouzite
235/	0 : =0C178H	OLDKEY	equ	0C178h	; nepouzite
236/	0 : =0C17AH	BCUR	equ	0C17Ah	; adresa kurzora (nepouzite)
237/	0 : =0C17CH	KEYEND	equ	0C17Ch	; adresa vrcholu tabulky klucov
238/	0 : =0C17EH	KEYTOP	equ	0C17Eh	; radoby hranica pamati pre kluce
239/	0 :				
240/	0 : =0C1B0H	FINDNR	equ	0C1B0h	; pozadovane cislo suboru
241/	0 : =0C1B1H	FINDTP	equ	0C1B1h	; pozadovany typ suboru
242/	0 : =0C1B2H	HEAD	equ	0C1B2h	; tabulka hlavicky suboru
243/	0 : =0C1B2H	NUMFIL	equ	0C1B2h	; cislo suboru
244/	0 : =0C1B3H	TYPFIL	equ	0C1B3h	; typ suboru
245/	0 : =0C1B4H	ADRFIL	equ	0C1B4h	; adresa zaciatku dat suboru
246/	0 : =0C1B6H	LENFIL	equ	0C1B6h	; dlzka suboru-1
247/	0 : =0C1B8H	NAMFIL	equ	0C1B8h	; 8 znakov nazvu suboru, doplnene medzerami
248/	0 :				
249/	0 : =0C1F1H	OCHR	equ	0C1F1h	; '*' => nacten chraneni soubor
250/	0 : =0C1F4H	VYSTUP	equ	0C1F4h	; zaciatok rutiny pre vykreslenie bodu
251/	0 : =0C1F8H	PLOTCOL	equ	0C1F8h	; farba vykreslovaného bodu
252/	0 : =0C1FAH	MODIFY	equ	0C1FAh	; sposob vykreslenia bodu
253/	0 :				
254/	0 :				
255/	0 :	; -----			
256/	0 :	; - dm - pripravi text txt tak, ze za posledne pismeno ulozi kod CR			
		dm	macro	txt	

```

257/ 0 :                db    txt
258/ 0 :                db    CR
259/ 0 :                endm
260/ 0 :
261/ 0 :                ; - kbd - položka tabulky ovladacich/editacnych klavesov
262/ 0 :                ;                kod klavesu, adresa obsluhy funkcie klavesu
263/ 0 :                kbd      macro    kod,adresa
264/ 0 :                db      kod
265/ 0 :                dw      adresa
266/ 0 :                endm
267/ 0 :
268/ 0 :                ; - moncmd - položka tabulky prikazov Monitora
269/ 0 :                ;                nazov prikazu, adresa obsluhy prikazu
270/ 0 :                moncmd    macro    nazov,adresa
271/ 0 :                db      nazov
272/ 0 :                db      0
273/ 0 :                dw      adresa
274/ 0 :                endm
275/ 0 :
276/ 0 :                ;-----
277/ 0 :
278/ 0 :
279/ 0 :                ;-----
280/ 0 :
281/8000 :                org      Begin
282/8000 :
283/8000 :                ;-----
284/8000 :                ; Start Monitora
285/8000 :                ; Po Resete je ROM mapovana aj od adresy 0.
286/8000 :                ; Po inicializacii sytemoveho PIO uz bude od adresy 0 RAM.
287/8000 : 31 00 80      Start:    lxi      sp,Stack      ; zasobnik na koniec pamati
288/8003 : C3 06 80      jmp      MON              ; skok do oblasti od 8000h
289/8006 :
290/8006 :                MON:    mvi      a,SYSCTRL      ; nastav: PA:OUT, PB:IN, PCH:IN, PCL:OUT
291/8008 : D3 F7      out      SYS55M      ; a pripoj RAM od adresy 0
292/800A : CD 1C 8B      call     OCHRAN      ; pokus o utajenie programov
293/800D :                ; DE=INITAB      ; adresa inicializacnej tabulky
294/800D : 21 30 C0      MONLL:    lxi      h,SYSVAR      ; zaciatok oblasti systemovych premennych
295/8010 : 0E 30      mvi      c,48      ; offset datovych oblasti
296/8012 : 3E 08      mvi      a,8      ; pocet oblasti
297/8014 : F5      MONIL2:    push     psw      ; odpamataj pocitadlo oblasti
298/8015 : 06 10      mvi      b,16      ; dlzka oblasti
299/8017 : 1A      MONIL1:    ldax     d      ; presun byte
300/8018 : 77      mov      m,a
301/8019 : 23      inx      h      ; posun ukazatele
302/801A : 13      inx      d
303/801B : 05      dcr      b      ; opakuj pre jednu 16-ticu

```

304/801C : C2 17 80	jnz	MONIL1	
305/801F : 09	dad	b	; prejsi na dalsiu oblast
306/8020 : F1	pop	psw	; obnov pocitadlo
307/8021 : 3D	dcr	a	; opakuje pre vsetky oblasti
308/8022 : C2 14 80	jnz	MONIL2	
309/8025 : 2A 7C C1	lhld	KEYEND	; adresa vrcholu tabulky klucov
310/8028 : 2B	dcx	h	
311/8029 : 77	mov	m,a	; nastav ukoncovaci znak
312/802A : CD 6B 8C	call	INIARA	; inicializuj PIO a zmaz obrazovku
313/802D : 21 89 81	lxi	h,M_Velcome	; adresa dat uvitacej melodie
314/8030 : CD A6 88	call	BELL	; zahraj melodiu
315/8033 :			; V tychto miestach je v povodnom Monitore PMD 85-2 nacitanie prvych 14 bytov
316/8033 :			; z ROM Modulu od adresy HEAD.
317/8033 :			; V Monitore Mata tato cast kodu chyba, takze nasledujuce tri instrukcie
318/8033 :			; stracaju zmysel.
319/8033 :			; Mato nema typicky ROM Modul, ale pripojit by sa na systemovy konektor dal.
320/8033 : 3A B2 C1	lda	HEAD	; prvý byte z ROM Modulu do A
321/8036 : FE CD	cpi	0CDh	; ak je to 0CDh - instrukcia CALL,
322/8038 : CA B2 C1	jz	HEAD	; ak ano, skoc vykonat tento kod
323/803B : 21 6D 83	lxi	h,T_OdReady	; priprav text "OS READY"
324/803E : 22 74 C0	shld	MESS	; na vypisanie v dialogovom riadku (DR)
325/8041 : 3E 3F	mvi	a,'?'	; typ suboru: binarny
326/8043 : 32 36 C1	sta	MGCHAR	
327/8046 : CD 13 81	call	CLR	; zmaz obsah buffra DR
328/8049 : CD 31 8B	call	PRBTXT	; pipni a vypis hlasenie
329/804C : CD A1 84	call	INKLAV	; test klavesnice
330/804F : FE 0D	cpi	CR	; stlaceny EOL
331/8051 : C4 43 8C	cnz	DOKEY	; ak nie, preved zmenu v DR
332/8054 : CD 55 88	call	WRBUFF	; vypis DR
333/8057 : 2A 70 C0	lhld	SPVAL	; nastav vrchol zasobnika
334/805A : F9	sphl		
335/805B : CD EE 8B	call	ENTER	; editacia v riadku
336/805E : 21 92 83	lxi	h,T_NoCmd	; priprav spravu "NO COMMAND"
337/8061 : 22 74 C0	shld	MESS	
338/8064 : 21 54 80	lxi	h,SYST	; priprav navratovu adresu
339/8067 : E5	push	h	; na zasobnik
340/8068 : 2A 7E C0	lhld	ITABDIR	; adresa tabulky prikazov -1
341/806B : EB	xchg		; do DE
342/806C : 2A 78 C0	lhld	ODLOZ	; adresa zadaneho textu v DR do HL
343/806F : 13	inx	d	; posun ukazatel v tabulke
344/8070 : 1A	ldax	d	; vezmi znak
345/8071 : A7	ana	a	; koniec mena prikazu?
346/8072 : CA 79 85	jz	SYST1	; ano, skoc prikaz vykonat
347/8075 : BE	cmp	m	; porovnaj znak prikazu
348/8076 : 23	inx	h	; posun ukazatel v DR
349/8077 : CA 6F 80	jz	FndCmdL	; zhodny znak, porovnaj dalsi
350/807A : 3C	inr	a	; koniec tabulky prikazov?

351/807B : CA 46 80	jz	SYST3	; ano, skoc vypisat chybove hlasiene
352/807E : 1A	ldax	d	; vyhľadaj koniec mena prikazu
353/807F : A7	ana	a	
354/8080 : 13	inx	d	
355/8081 : C2 7E 80	jnz	FndCmdM	
356/8084 : 13	inx	d	; preskoc adresu a prejdi na dalsi prikaz
357/8085 : C3 6C 80	jmp	FndCmdK	; skoc porovnat dalsi
358/8088 :			
359/8088 :			
360/8088 :			; -----
361/8088 :			; Rolovanie obrazovky hore o jeden zR.
362/8088 :			; Na rolovanie sa pouziva register SP, takze tato rutina sa neda pouzit
362/8088 :			; s prerusenim.
363/8088 : 21 00 00	ROLL:	lxi	h,0 ; adresu zasobnika
364/808B : 39		dad	sp ; do HL
365/808C : 22 3E C1		shld	STACK ; a odpamataj
366/808F : 2A FC C0		lhld	RADSIR ; offset na dalsi zR
367/8092 : 01 00 C0		lxi	b,VRAM ; adresa VRAM
368/8095 : 09		dad	b ; pripocitaj offset na dalsi riadok
369/8096 : F9		sphl	; zdrojova adresa VRAM do SP
370/8097 : 21 FF BF		lxi	h,VRAM-1 ; cielova adresa VRAM
371/809A : 3A F4 C0		lda	IIROLL ; pocet scrollovanych uR
372/809D : 06 08	RollK:	mvi	b,48/6 ; pocet cyklov
373/809F : D1	RollL:	pop	d ; vezmi 2 byty zo zdrojovej adresy
374/80A0 : 23		inx	h
375/80A1 : 73		mov	m,e ; uloz oba byty na cielove miesto
376/80A2 : 23		inx	h
377/80A3 : 72		mov	m,d
378/80A4 : D1		pop	d ; vezmi 2 byty zo zdrojovej adresy
379/80A5 : 23		inx	h
380/80A6 : 73		mov	m,e ; uloz oba byty na cielove miesto
381/80A7 : 23		inx	h
382/80A8 : 72		mov	m,d
383/80A9 : D1		pop	d ; vezmi 2 byty zo zdrojovej adresy
384/80AA : 23		inx	h
385/80AB : 73		mov	m,e ; uloz oba byty na cielove miesto
386/80AC : 23		inx	h
387/80AD : 72		mov	m,d
388/80AE : 05		dcr	b ; opakuj pre cely uR
389/80AF : C2 9F 80		jnz	RollL
390/80B2 : EB		xchg	; cielovu adresu na chvilu do DE
391/80B3 : 21 10 00		lxi	h,16 ; offset na dalsi uR
392/80B6 : 39		dad	sp ; posun zdrojovu adresu
393/80B7 : F9		sphl	; a opat uloz do SP
394/80B8 : 21 10 00		lxi	h,16 ; offset na dalsi uR
395/80BB : 19		dad	d ; posun cielovu adresu
396/80BC : 3D		dcr	a ; opakuj pre celu scrollovanu vysku
397/80BD : C2 9D 80		jnz	RollK



```

398/80C0 : 2A 3E C1          lhld    STACK          ; obnov povodnu adresu zasobnika
399/80C3 : F9                sphl                      ;
400/80C4 : 2A F6 C0          lhld    POROLL          ; adresa posledneho zR
401/80C7 : EB                xchg                      ; do DE
402/80C8 : 3A F5 C0          lda     RSIRRAD          ; vyska zR v uR
403/80CB : 47                mov     b,a              ; B
404/80CC : 2A F8 C0          lhld    CURROL          ; adresa pre vypis dalsieho riadku do HL
405/80CF : C3 B3 85          jmp     EraLines        ; skoc zmazat posledny riadok
406/80D2 :
407/80D2 :                  ;-----
408/80D2 :                  ; data pre vseobecne pipnutie
409/80D2 : 0A 14 0A 1E FF    M_Beep:      db      10,20,10,30,-1
410/80D7 :
411/80D7 :                  ;-----
412/80D7 :                  ; bitove masky pre vykreslenie bodu
413/80D7 :                  ; (masky 40h a 80h su tu zbytocne)
414/80D7 : 01 02 04 08 10 20 40 80 BitMasks: db      01h,02h,04h,08h,10h,20h,40h,80h
415/80DF :
416/80DF :                  ;-----
417/80DF :                  ; Vstupna sestnastkova konverzia 4 bity
418/80DF :                  ; I: HL=adresa znaku
419/80DF :                  ; O: CY=0, A=<0,15>
420/80DF :                  ; CY=1 - znak nie je sestnastkova cislica
421/80DF :                  ; M: AF
422/80DF : 7E                HEX1:      mov     a,m          ; znak zo vstupu
423/80E0 : FE 30              cpi      '0'          ; je to cislica?
424/80E2 : D8                rc          ; nie, navrat s CY=1
425/80E3 : FE 3A              cpi      '9'+1        ; je to cislica 0..9?
426/80E5 : DA F1 80          jc       HEX1D        ; ano, skoc
427/80E8 : D6 07              sui      7          ; uprav rozsah na A..F
428/80EA : FE 3A              cpi      '9'+1        ; je to pismeno A..F
429/80EC : D8                rc          ; nie, navrat s CY=1
430/80ED : FE 40              cpi      'F'+1-7      ;
431/80EF : 3F                cmc          ;
432/80F0 : D8                rc          ; nie, navrat s CY=1
433/80F1 : E6 0F              HEX1D:      ani      0Fh          ; odmaskuj iba platne bity
434/80F3 : C9                ret
435/80F4 :
436/80F4 :                  ;-----
437/80F4 :                  ; data pre pipnutie pri stlaceni klavesu
438/80F4 : 03 19 FF    M_Key:      db      3,25,-1
439/80F7 :
440/80F7 :                  ;-----
441/80F7 :                  ; Vstupna sestnastkova konverzia 8 bitov.
442/80F7 :                  ; I: HL=adresa 2 znakov
443/80F7 :                  ; O: CY=0, A=vysledna hodnota, HL=HL+2
444/80F7 :                  ; CY=1 - znak nie je sestnastkova cislica

```

```

445/80F7 : ; M: AF, B, HL
446/80F7 : CD DF 80 PAIRIN: call HEX1 ; precitaj a spracuj 1. znak
447/80FA : D8 rc ; chybný znak, navrat s CY=1
448/80FB : 07 rlc ; presun do horných styroch bitov
449/80FC : 07 rlc
450/80FD : 07 rlc
451/80FE : 07 rlc
452/80FF : 47 mov b,a ; a odlož do B
453/8100 : 23 inx h ; posun ukazateľ na ďalší znak
454/8101 : CD DF 80 call HEX1 ; precitaj a spracuj 2. znak
455/8104 : D8 rc ; chybný znak, navrat s CY=1
456/8105 : 23 inx h ; posun ukazateľ na ďalší znak
457/8106 : B0 ora b ; pripoj horné bity
458/8107 : C9 ret
459/8108 :
460/8108 : ;-----
461/8108 : FF db 0FFh
462/8109 :
463/8109 : ;-----
464/8109 : ; Vstupná šestnástková konverzia 16 bitov.
465/8109 : ; I: HL=adresa 2 znakov
466/8109 : ; O: CY=0, DE=výsledná hodnota, HL=HL+4
467/8109 : ; CY=1 - znak nie je šestnástková číslica
468/8109 : ; M: AF, B, DE, HL
469/8109 : CD F7 80 ADRIN: call PAIRIN ; precitaj a spracuj vyšší byte
470/810C : D8 rc ; chybný znak, navrat s CY=1
471/810D : 57 mov d,a ; ulož do D
472/810E : CD F7 80 call PAIRIN ; precitaj a spracuj nižší byte
473/8111 : 5F mov e,a ; a ulož do E
474/8112 : C9 ret
475/8113 :
476/8113 : ;-----
477/8113 : ; Obsluha klavesu CLL : Zmazanie DR.
478/8113 : ; klaves CLL (CNT + B)
479/8113 : CD F6 85 CLR: call HOME ; nastav kurzor na prvý znak DR
480/8116 : ; pokračuj v SHCLR
481/8116 :
482/8116 : ;-----
483/8116 : ; Obsluha klavesu DELL : Zmazanie DR vpravo od kurzora.
484/8116 : ; klaves DELL (CNT + .)
485/8116 : CD 5E 81 SHCLR: call CURPOS ; do HL adresu akt. znaku v buffri DR
486/8119 : 3A 34 C0 lda DLZRAD ; nižší byte koncové adresy buffra DR
487/811C : 3C inr a ; +1
488/811D : 36 20 ShClrL: mvi m, ' ' ; zapis medzeru
489/811F : 2C inr l ; ďalšia pozícia
490/8120 : BD cmp l ; koniec DR?
491/8121 : C2 1D 81 jnz ShClrL ; nie, opakuj do konca buffra

```

```

492/8124 : C9                                ret
493/8125 :
494/8125 : ;-----
495/8125 : ; Vystupna sestnastkova konverzia 8 bitov s vypisom na obrazovku.
496/8125 : ; I: A=0..255
497/8125 : ; O: vola PRTOUT - vysledok vypisuje na obrazovku
498/8125 : ; M: B, AF
499/8125 : 47  PREV01:      mov      b,a          ; odloz byte do B
500/8126 : 0F                      rrc          ; vyssie 4 bity do nizsich
501/8127 : 0F                      rrc
502/8128 : 0F                      rrc
503/8129 : 0F                      rrc
504/812A : CD 6B 88      call     PRE00          ; preved na ASCII vyssi nibble
505/812D : CD 00 85      call     PRTOUT         ; a vypis
506/8130 : 78                      mov      a,b          ; povodny byte do A
507/8131 : CD 6B 88      call     PRE00          ; preved na ASCII nizsi nibble
508/8134 : C3 00 85      jmp      PRTOUT         ; a vypis
509/8137 :
510/8137 : ;-----
511/8137 : ; Fragment prikazu Monitora SUB, po vypisani zbytku adresy z DE, navrat do
512/8137 : ; slucky Monitora a cakanie na doplnenie dalsich dat pre ulozenie do RAM.
513/8137 : 32 36 C0  PREV21:      sta      DCUR          ; pozicia kurzora v buffri
514/813A : 7B                      mov      a,e          ; nizsi byte adresy pre ukladanie dat
515/813B :
516/813B : ;-----
517/813B : ; Vystupna sestnastkova konverzia 8 bitov so zapisom do buffra.
518/813B : ; I: A=0..255, HL=adresa buffra
519/813B : ; O: HL=HL+1
520/813B : ; M: HL, B, AF
521/813B : 47  PREV02:      mov      b,a          ; odloz byte do B
522/813C : 0F                      rrc          ; vyssie 4 bity do nizsich
523/813D : 0F                      rrc
524/813E : 0F                      rrc
525/813F : 0F                      rrc
526/8140 : CD 6B 88      call     PRE00          ; preved na ASCII vyssi nibble
527/8143 : 77                      mov      m,a          ; a uloz do buffra
528/8144 : 23                      inc      h          ; posun ukazatel
529/8145 : 78                      mov      a,b          ; povodny byte do A
530/8146 : CD 6B 88      call     PRE00          ; preved na ASCII nizsi nibble
531/8149 : 77                      mov      m,a          ; a uloz do buffra
532/814A : C9                      ret
533/814B :
534/814B : ;-----
535/814B : ; Vstupna sestnastkova konverzia 16 bitov s vypisom chyby.
536/814B : ; I: [CURCH]=adresa textu
537/814B : ; O: ak nedoslo ku chybe, [CURCH]=[CURCH]+4, DE=precitane cislo
538/814B : ; M: vsetky

```

```

539/814B : 2A 72 C0
540/814E : CD 09 81
541/8151 : 22 72 C0
542/8154 : D0
543/8155 : 21 46 83
544/8158 : 22 74 C0
545/815B : C3 49 80
546/815E :
547/815E :
548/815E :
549/815E :
550/815E :
551/815E :
552/815E : 2A 38 C0
553/8161 : EB
554/8162 : 2A 36 C0
555/8165 : 19
556/8166 : C9
557/8167 :
558/8167 :
559/8167 :
560/8167 :
561/8167 : 3A BF C0
562/816A : EE 01
563/816C : 32 BF C0
564/816F : C9
565/8170 :
566/8170 :
567/8170 :
568/8170 :
569/8170 :
570/8170 :
571/8170 : C5
572/8171 : 5A
573/8172 : DB F6
574/8174 : EE 06
575/8176 : D3 F6
576/8178 : 0B
577/8179 : 78
578/817A : B1
579/817B : CA 85 81
580/817E : 1D
581/817F : C2 78 81
582/8182 : C3 71 81
583/8185 :
584/8185 : C1
585/8186 : D3 F6

RADR:          lhld    CURCH          ; HL=adresa textu
               call    ADRIN          ; vyhodnot 16 bitove cislo HLL
               shld    CURCH          ; uloz novu poziciu
               rnc      ; navrat, ak nenastala chyba
               lxi      h,T_ErrAddr   ; inak, priprav spravu
               shld    MESS            ; "ERROR IN ADDRESS"
               jmp      PRBWTXT

;-----
; Vypocet adresy aktualneho znaku v buffri DR.
; I: [RPOS], [DCUR]
; O: HL=adresa aktualneho znaku v buffri DR
; M: HL, DE
CURPOS:         lhld    RPOS           ; adresa 1. znaku v buffri
CURPOSHL:       xchg
               lhld    DCUR           ; pozicia kurzora
               dad      d             ; HL=adresy aktualneho znaku v buffri DR
               ret

;-----
; Obsluha klavesu BEEP : Zap/vyp pipnutia pri stlaceni klavesu.
; klaves BEEP (CNT + <-->)
MODBEP:         lda      SBEEP         ; zmen stav zapnutia/vypnutia pipnutia
               xri      1
               sta      SBEEP
               ret

;-----
; Generovanie tonu.
; I: B=dlzka trvania, D=vyska tonu
; O: -
; M: AF, DE
Ton:            push    b              ; odpamataj BC
TonD:           mov     e,d            ; vyska tonu do E
               in       SYS55C        ; stav portu C
               xri      X_AM|X_RFU    ; invertuj bit(y) akustickeho menica
               out      SYS55C        ; a uloz na port C
TonB:           dcx     b              ; zniz pocitadlo dlzky
               mov     a,b
               ora      c
               jz       TonE           ; skoc, ak sa vynulovalo
               dcr      e              ; zniz pocitadlo vysky tonu
               jnz      TonB           ; opakuj pre celu polperiodu
               jmp      TonD           ; zacni novu polperiodu

TonE:           pop     b              ; obnov BC
               out      SYS55C        ; vynuluj port C

```

```

586/8188 : C9                                ret
587/8189 :
588/8189 : ;-----
589/8189 : ; Tabulka uvitacej melodie.
590/8189 : ; [dlzka tonu, vyska tonu]*,-1
591/8189 : 0F 23 0F 20 0F 1D 0F 1A M_Velcome: db 15,35,15,32,15,29,15,26,15,23,-1
8191 : 0F 17 FF
592/8194 :
593/8194 : ;-----
594/8194 : ; Vypis znaku pri editacii v DR.
595/8194 : CD 5E 81 CHAR: call CURPOS ; adresa aktualnej pozicie v DR
596/8197 : 77 mov m,a ; uloz znak
597/8198 : 3A 3B C0 lda BEEPPOS ; pozicia pre pipnutie
598/8198 : BD cmp l ; kurzor je na tejto pozicii?
599/819C : CC A3 88 cz BEEP ; pipni, ak ano
600/819F : ; pokracuj v RIGHT
601/819F :
602/819F : ;-----
603/819F : ; Obsluha klavesu --> : posun kurzora vpravo
604/819F : ; klaves -->
605/819F : 21 36 C0 RIGHT: lxi h,DCUR ; pozicia kurzora v buffri
606/81A2 : 34 inr m ; +1
607/81A3 : 3A FE C0 lda DLZR ; max. pocet zobrazenyh znakov v riadku
608/81A6 : BE cmp m ; dosiahli sme danu poziciu?
609/81A7 : C0 rnz ; nie, navrat
610/81A8 : 35 dcr m ; vrat predoslu hodnotu
611/81A9 : ; pokracuj v LLEFT
612/81A9 :
613/81A9 : ;-----
614/81A9 : ; Obsluha klavesu SHF + <-- : posun zobrazovanej casti DR dolava
615/81A9 : ; klaves SHF + <--
616/81A9 : 21 38 C0 LLEFT: lxi h,RPOS ; prvvy vypisovany znak
617/81AC : 3A FF C0 lda DLZP ; max. pocet vysunutych znakov v riadku
618/81AF : BE cmp m ; uz je vysunutych max. pocet znakov?
619/81B0 : C8 rz ; ak ano, navrat
620/81B1 : 34 inr m ; inak, zasun dalsi znak
621/81B2 : C9 OnlyRet: ret
622/81B3 :
623/81B3 : ;-----
624/81B3 : ; Obsluha klavesu <-- : posun kurzora vlavo
625/81B3 : ; klaves <--
626/81B3 : 21 36 C0 LEFT: lxi h,DCUR ; pozicia kurzora v buffri
627/81B6 : 35 dcr m ; -1
628/81B7 : F0 rp ; OK, navrat
629/81B8 : 34 inr m ; vrat na 0
630/81B9 : ; pokracuj v RRIGHT
631/81B9 :

```

```

632/81B9 : ;-----
633/81B9 : ; Obsluha klavesu SHF + --> : posun zobrazovanej casti DR doprava
634/81B9 : ; klaves SHF + -->
635/81B9 : 21 38 C0 RRIGHT: lxi h,RPOS ; prvvy vypisovany znak
636/81BC : 35 dcr m ; -1
637/81BD : F0 rp ; OK, navrat
638/81BE : 34 inr m ; vysun dalsi znak
639/81BF : C9 ret
640/81C0 :
641/81C0 : ;-----
642/81C0 : ; Prenos mena suboru do hlavicky suboru s doplnenim medzerami na max. 8 znakov.
643/81C0 : ; Zdrojove meno suboru moze byt ukoncene CR.
644/81C0 : ; I: HL=zdvojova adresa mena suboru
645/81C0 : ; O: -
646/81C0 : ; M: HL, DE, B, AF
647/81C0 : 11 B8 C1 TEXTHEAD: lxi d,NAMFIL ; buffer mena suboru
648/81C3 : ; pokracuj v TEXTTH
649/81C3 :
650/81C3 : ;-----
651/81C3 : ; Prenos mena suboru s doplnenim medzerami na 8 znakov.
652/81C3 : ; Zdrojove meno suboru moze byt ukoncene CR.
653/81C3 : ; I: HL=zdvojova adresa, DE=cielova adresa
654/81C3 : ; O: -
655/81C3 : ; M: HL, DE, B, AF
656/81C3 : 06 08 TEXTTH: mvi b,8 ; dlzka mena suboru
657/81C5 : ; pokracuj v TEXTSP
658/81C5 :
659/81C5 : ;-----
660/81C5 : ; Prenos textu s doplnenim medzerami na B znakov.
661/81C5 : ; Text moze byt ukonceny CR.
662/81C5 : ; I: HL=zdvojova adresa, DE=cielova adresa, B=maximalny pocet znakov
663/81C5 : ; O: -
664/81C5 : ; M: HL, DE, B, AF
665/81C5 : 7E TEXTSP: mov a,m ; vezmi znak
666/81C6 : FE 0D cpi CR ; je to koniec riadku?
667/81C8 : C2 CE 81 jnz TEXTC ; ak nie, skoc
668/81CB : 3E 20 mvi a,' ' ; inak, dopln medzeru
669/81CD : 2B dcx h ; vrat ukazatel
670/81CE : 12 TEXTC: stax d ; uloz znak
671/81CF : 23 inx h ; posun oba ukazatele
672/81D0 : 13 inx d
673/81D1 : 05 dcr b ; a opakuj B krat
674/81D2 : C2 C5 81 jnz TEXTSP
675/81D5 : C9 ret
676/81D6 :
677/81D6 : ;-----
678/81D6 : ; Obsluha prikazu JUMP : skok na vykonanie programu.

```

```

679/81D6 : ; Format: JUMP aaaa
680/81D6 : 2A 38 C1 JUMP:      lhld    MONX      ; navratova adresa
681/81D9 : E3          xthl      ; na zasobnik
682/81DA : CD 4B 81      call    RADR      ; precitaj adresu pre skok
683/81DD : D5          push    d      ; uloz na zasobnik
684/81DE : 21 7D 82      lxi     h,T_Exec  ; vypis text "EXECUTIVE"
685/81E1 : CD 39 8B      call    PRHTXT
686/81E4 : FB          ei          ; povol prerusenie - PRECO???
687/81E5 : C9          ret         ; skoc nepriamo na danu adresu
688/81E6 :
689/81E6 : ;-----
690/81E6 : ; Obsluha prikazu DUMP : vypis obsahu pamate hexadecimalne a znakov.
691/81E6 : ; V jednom riadku sa vypisuje adresa, 8 bytov hexadecimalne a 8 znakov.
692/81E6 : ;   AAAA HH HH HH HH HH HH HH HH   CCCCCCCC
693/81E6 : ; Format: DUMP aaaa
694/81E6 : CD 4B 81 DUMP:      call    RADR      ; precitaj adresu vypisu
695/81E9 : EB          xchg      ; uloz do HL
696/81EA : E5 DumpL:      push    h      ; odpamataj aktualnu adresu
697/81EB : 7C          mov     a,h      ; vypis vyssi byte
698/81EC : CD 25 81      call    PREV01
699/81EF : 7D          mov     a,l      ; vypis nizsi byte
700/81F0 : CD 25 81      call    PREV01
701/81F3 : CD FB 84      call    SPACE      ; vypis medzeru
702/81F6 : 0E 08          mvi     c,8      ; 8 bytov na riadok
703/81F8 : CD FB 84 DumpH:      call    SPACE      ; vypis medzeru
704/81FB : 7E          mov     a,m      ; prevezmi byte
705/81FC : CD 25 81      call    PREV01      ; vypis ho
706/81FF : 23          inx     h      ; posun ukazatel
707/8200 : 0D          dcr     c      ; opakuje 8x
708/8201 : C2 F8 81      jnz     DumpH
709/8204 : 0E 05          mvi     c,5      ; vypis 5 medzier
710/8206 : CD FB 84 DumpS:      call    SPACE
711/8209 : 0D          dcr     c
712/820A : C2 06 82      jnz     DumpS
713/820D : 0E 08          mvi     c,8      ; 8 znakov
714/820F : E1          pop     h      ; obnov zdrojovu adresu
715/8210 : 7E DumpC:      mov     a,m      ; prevezmi znak
716/8211 : E5          push    h
717/8212 : CD 42 85      call    PRT02      ; vypis znak
718/8215 : E1          pop     h
719/8216 : 23          inx     h      ; posun adresu
720/8217 : 0D          dcr     c      ; opakuje 8x
721/8218 : C2 10 82      jnz     DumpC
722/821B : DB F6 DumpW:      in      SYS55C      ; otestuj stav klavesu SHF
723/821D : E6 20          ani     X_SHF
724/821F : CA 1B 82      jz      DumpW      ; cakaj, kym je stlaceny
725/8222 : CD FE 84      call    PRTCR      ; odriadkuj

```

```

726/8225 : CD 74 8C      call    STOP          ; otestuj stav klavesu STOP
727/8228 : C2 EA 81      jnz     DumpL          ; pokracuj, kym nebol stlaceny
728/822B : CD 13 81      call    CLR            ; zmaz DR
729/822E : C3 54 80      jmp     SYST           ; navrat do hlavnej slucky Monitora
730/8231 :
731/8231 :
732/8231 :                ;-----
732/8231 :                ; Tabulka inicializacnych dat pre systemove premenne.
733/8231 :                INITAB:                ; 0C030h
734/8231 : 00 7F          dw      7F00h          ; BUF
735/8233 : C0 FF          dw      0FFC0h         ; DIAL
736/8235 : 4F 7F          dw      7F00h+79       ; DLZRAD
737/8237 : 00 00          dw      0             ; DCUR
738/8239 : 00 7F          dw      7F00h         ; RPOS
739/823B : 00             db      0             ; COLOR
740/823C : 45             db      69            ; BEEPPOS
741/823D : 08 86          dw      A_Font_1+8     ; TAS00
742/823F : 00 00          dw      0             ; CURSOR
743/8241 :
744/8241 :                ; 0C070h
745/8241 : FF 7F          dw      7FFFh         ; SPVAL
746/8243 : 00 00          dw      0             ; CURCH
747/8245 : 00 00          dw      0             ; MESS
748/8247 : 00 00          dw      0             ; RETCALL
749/8249 : 82 7F          dw      7F82h         ; ODLOZ
750/824B : 00 00          dw      0             ; RETSP
751/824D : 20             db      32            ; SPEED (32 => ~311 us)
752/824E : 00             db      0             ; LanFlag
753/824F : EB 82          dw      CmdTab-1       ; ITABDIR
754/8251 :
755/8251 :                ; 0C0B0h
756/8251 : 00 FF          dw      0FF00h         ; TASCII - nevyuzite
757/8253 : 08 86          dw      A_Font_1+8     ; nevyuzite
758/8255 : 08 87          dw      A_Font_1+100h+8 ; nevyuzite
759/8257 : C8 88          dw      88C8h         ; nevyuzite
760/8259 : 00 FF          dw      0FF00h         ; nevyuzite
761/825B : 00 FF          dw      0FF00h         ; nevyuzite
762/825D : 00             db      0             ; KeyLast
763/825E : 32             db      50            ; KeyRepeat
764/825F : 00 01          dw      0100h
765/8261 :
766/8261 :                ; 0C0F0h
767/8261 : 00             nop                     ; RAM
768/8262 : C3 B2 81      jmp     OnlyRet
769/8265 : EA             db      234           ; IIROLL
770/8266 : 09             db      9             ; RSIRRAD
771/8267 : B0 FA          dw      0FAB0h         ; POROLL
772/8269 : 00 FD          dw      0FD00h         ; CURROL

```



773/826B : FB		db	251		; VYSKA
774/826C : 30		db	48		; ENLNW
775/826D : 40 02		dw	9*40h		; RADSIR
776/826F : 30		db	48		; DLZR
777/8270 : 20		db	32		; DLZP
778/8271 :					
779/8271 :				; 0C130h	
780/8271 : D2 80		dw	M_Beep		; BEEDT
781/8273 : B0 82		dw	KbdTab-2		; KDIR
782/8275 : 00		db	0		; ASCII
783/8276 : 00		db	0		; SWRK
784/8277 : 00		db	0		; MGCHAR
785/8278 : 00		db	0		; SPTL
786/8279 : 3B 80		dw	MON1		; MONX
787/827B : 52 7F		dw	7F52h		; CDBUF
788/827D :					
789/827D : (MACRO)	T_Exec:	dm	"++ EXECUTIVE ++"		; LSTR, STACK, X1, Y1, X2, Y2
789/827D : 2B 2B 20 45 58 45 43 55		db	"++ EXECUTIVE ++"		
8285 : 54 49 56 45 20 2B 2B					
789/828C : 0D		db	CR		
790/828D :					; DCUOLD, DELKA, OLDKEY, BCUR
791/828D :					
792/828D :				; 0C17Ch	
793/828D : 00 7F		dw	7F00h		; KEYEND
794/828F : 00 7D		dw	7D00h		; KEYTOP
795/8291 :					
796/8291 :				; 0C1B0h	
797/8291 : (MACRO)	T_ErrData:	dm	"++ ERROR IN DATA ++"		; FINDNR, FINDTP, HEAD, NUMFIL
797/8291 : 2B 2B 20 45 52 52 4F 52		db	"++ ERROR IN DATA ++"		
8299 : 20 49 4E 20 44 41 54 41					
82A1 : 20 2B 2B					
797/82A4 : 0D		db	CR		
798/82A5 :					; TYPFIL, ADRFIL, LENFIL, NAMFIL, OCHR
799/82A5 :					; LanSP, LanID, LanStat
800/82A5 :					
801/82A5 :				; 0C1F4h	
802/82A5 : 7E		mov	a,m		; VYSTUP
803/82A6 : FE 3F		cpi	3Fh		
804/82A8 : FE 00		cpi	0		; PLOTCOL
805/82AA : 00		nop			
806/82AB : A8		xra	b		; MODIFY
807/82AC : 00		nop			
808/82AD : 77		mov	m,a		
809/82AE : C9		ret			
811/82AF : 10		db	16		; LANTM1
812/82B0 : 21		db	33		; LANTM2
816/82B1 :					

```

817/82B1 : ;-----
818/82B1 : 00          db      0
819/82B2 :
820/82B2 : ;-----
821/82B2 : ; tabulka editacnych klavesov
822/82B2 : (MACRO) KbdTab: kbd    KPTL,PTL      ; zap/vyp vypisu obsahu DR na obrazovku
822/82B2 : 0F          db    KPTL
822/82B3 : C7 83      dw    PTL
823/82B5 : (MACRO) kbd    KMON,MON1      ; skok do Monitora
823/82B5 : 12          db    KMON
823/82B6 : 3B 80      dw    MON1
824/82B8 : (MACRO) kbd    KWRK,WRK      ; zapis do klucovych klavesov Kx
824/82B8 : 0B          db    KWRK
824/82B9 : C1 83      dw    WRK
825/82BB : (MACRO) kbd    KCD,CD        ; vypisanie posledneho hlasenia do DR
825/82BB : 1E          db    KCD
825/82BC : 9C 88      dw    CD
826/82BE : (MACRO) kbd    KRCL,RCL      ; vyvolanie naposledy odoslaného riadku
826/82BE : 17          db    KRCL
826/82BF : 33 88      dw    RCL
827/82C1 : (MACRO) kbd    KINST,INS     ; vsunutie medzery v mieste kurzora
827/82C1 : 1C          db    KINST
827/82C2 : 0D 88      dw    INS
828/82C4 : (MACRO) kbd    KDEL,DEL      ; vymazanie znaku v mieste kurzora
828/82C4 : 1D          db    KDEL
828/82C5 : 4D 8D      dw    DEL
829/82C7 : (MACRO) kbd    KCLL,CLR      ; zmazanie DR
829/82C7 : 1B          db    KCLL
829/82C8 : 13 81      dw    CLR
830/82CA : (MACRO) kbd    KLEFT,LEFT    ; posun kurzora vľavo
830/82CA : 08          db    KLEFT
830/82CB : B3 81      dw    LEFT
831/82CD : (MACRO) kbd    KBGNL,HOME    ; nastavenie kurzora na začiatok DR
831/82CD : 0C          db    KBGNL
831/82CE : F6 85      dw    HOME
832/82D0 : (MACRO) kbd    KRIGHT,RIGHT  ; posun kurzora vpravo
832/82D0 : 18          db    KRIGHT
832/82D1 : 9F 81      dw    RIGHT
833/82D3 : (MACRO) kbd    KSLEFT,LLEFT  ; posun zobrazovanej časti DR doľava
833/82D3 : 1A          db    KSLEFT
833/82D4 : A9 81      dw    LLEFT
834/82D6 : (MACRO) kbd    KENDL,ENDL    ; posun kurzora za posledný znak
834/82D6 : 03          db    KENDL
834/82D7 : 90 8B      dw    ENDL
835/82D9 : (MACRO) kbd    KSRIGHT,RRIGHT ; posun zobrazovanej časti DR doprava
835/82D9 : 19          db    KSRIGHT
835/82DA : B9 81      dw    RRIGHT

```

836/82DC : (MACRO)	kbd	KEOL,EOL	; ukoncenie editacie v DR
836/82DC : 0D	db	KEOL	
836/82DD : B6 8B	dw	EOL	
837/82DF : (MACRO)	kbd	KDELL,SHCLR	; zmazanie DR vpravo od kurzora
837/82DF : 01	db	KDELL	
837/82E0 : 16 81	dw	SHCLR	
838/82E2 : (MACRO)	kbd	KCLS,ERASE	; vymazanie obrazovky okrem DR
838/82E2 : 1F	db	KCLS	
838/82E3 : A7 85	dw	ERASE	
839/82E5 : (MACRO)	kbd	KINVR,MODNEG	; zap/vyp inverzneho zobrazenia
839/82E5 : 0A	db	KINVR	
839/82E6 : 9F 85	dw	MODNEG	
840/82E8 : (MACRO)	kbd	KBEEP,MODBEP	; zap/vyp pipnutia pri stlaceni klavesu
840/82E8 : 10	db	KBEEP	
840/82E9 : 67 81	dw	MODBEP	
841/82EB : 00	db	0	
842/82EC :			
843/82EC :			
844/82EC :			
845/82EC : (MACRO)			
845/82EC : 53 55 42	moncmd	"SUB",SUB	; zapis dat do pamate
845/82EF : 00	db	"SUB"	
845/82F0 : 8F 8A	db	0	
846/82F2 : (MACRO)	dw	SUB	
846/82F2 : 4D 45 4D	moncmd	"MEM",MEM	; vypisanie obsahu pamate do DR
846/82F5 : 00	db	"MEM"	
846/82F6 : 87 85	db	0	
847/82F8 : (MACRO)	dw	MEM	
847/82F8 : 4A 55 4D 50	moncmd	"JUMP",JUMP	; spustenie programu
847/82FC : 00	db	"JUMP"	
847/82FD : D6 81	db	0	
848/82FF : (MACRO)	dw	JUMP	
848/82FF : 44 55 4D 50	moncmd	"DUMP",DUMP	; vypisanie obsahu pamate na obrazovku
848/8303 : 00	db	"DUMP"	
848/8304 : E6 81	db	0	
849/8306 : (MACRO)	dw	DUMP	
849/8306 : 4D 47 53 56	moncmd	"MGSV",MGSV	; zapis suboru na MGF
849/830A : 00	db	"MGSV"	
849/830B : 7F 8E	db	0	
850/830D : (MACRO)	dw	MGSV	
850/830D : 4D 47 4C 44	moncmd	"MGLD",MGLD	; nacitanie suboru z MGF
850/8311 : 00	db	"MGLD"	
850/8312 : 73 88	db	0	
851/8314 : (MACRO)	dw	MGLD	
851/8314 : 4D 47 45 4E 44	moncmd	"MGEND",MGEND	; kontrola zaznamu na MGF
851/8319 : 00	db	"MGEND"	
851/831A : 79 88	db	0	
	dw	MGEND	

852/831C : (MACRO)		moncmd	"BASIC",BASIC	; spustenie interpretera BASICu
852/831C : 42 41 53 49 43		db	"BASIC"	
852/8321 : 00		db	0	
852/8322 : 11 8F		dw	BASIC	
853/8324 : (MACRO)		moncmd	"JOB",JOB	; nacitanie bloku z Rom Modulu a spustenie
853/8324 : 4A 4F 42		db	"JOB"	
853/8327 : 00		db	0	
853/8328 : D7 8F		dw	JOB	
873/832A : (MACRO)		moncmd	"DEMO",DEMO	; spustenie DEMO programu BASICu
873/832A : 44 45 4D 4F		db	"DEMO"	
873/832E : 00		db	0	
873/832F : 00 90		dw	DEMO	
874/8331 : (MACRO)		moncmd	"READ",READ	; citanie dat/suboru zo siete
874/8331 : 52 45 41 44		db	"READ"	
874/8335 : 00		db	0	
874/8336 : 92 91		dw	READ	
875/8338 : (MACRO)		moncmd	"SEND",SEND	; odoslanie dat/suboru do siete
875/8338 : 53 45 4E 44		db	"SEND"	
875/833C : 00		db	0	
875/833D : 1D 90		dw	SEND	
876/833F : (MACRO)	MgLdR:	moncmd	"SET",SET	; prihlasenie/odhlasenie do/zo siete
876/833F : 53 45 54		db	"SET"	
876/8342 : 00		db	0	
876/8343 : A6 95		dw	SET	
877/8345 : FF		db	-1	
879/8346 :				
880/8346 :				
881/8346 :				
882/8346 : (MACRO)				
882/8346 : 2B 2B 20 45 52 52 4F 52		dm	"++ ERROR IN ADDRES ++"	
834E : 20 49 4E 20 41 44 44 52		db	"++ ERROR IN ADDRES ++"	
8356 : 45 53 20 2B 2B				
882/835B : 0D		db	CR	
883/835C : (MACRO)				
883/835C : 2B 2B 20 46 49 4C 45 20	T_FileErr:	dm	"++ FILE ERROR ++"	
8364 : 45 52 52 4F 52 20 2B 2B		db	"++ FILE ERROR ++"	
883/836C : 0D		db	CR	
884/836D : (MACRO)				
884/836D : 2A 2A 20 4F 53 20 52 45	T_OdReady:	dm	"** OS READY **"	
8375 : 41 44 59 20 2A 2A		db	"** OS READY **"	
884/837B : 0D		db	CR	
886/837C : (MACRO)				
886/837C : 2B 2B 20 4D 45 4D 4F 52	T_MemOF:	dm	"++ MEMORY OVERFLOW ++"	
8384 : 59 20 4F 56 45 52 46 4C		db	"++ MEMORY OVERFLOW ++"	
838C : 4F 57 20 2B 2B				
886/8391 : 0D		db	CR	
890/8392 : (MACRO)				
	T_NoCmd:	dm	"++ NO COMMAND ++"	

```

890/8392 : 2B 2B 20 4E 4F 20 43 4F          db    "++ NO COMMAND ++"
      839A : 4D 4D 41 4E 44 20 2B 2B
890/83A2 : 0D                                db    CR
891/83A3 :
892/83A3 : ;-----
893/83A3 : ; Pokracovanie obsluhy prikazu SUB.
894/83A3 : ; I: DE=nasledujuca adresa
895/83A3 : D5      SUB03:      push    d          ; odpamataj nasledujucu adresu
896/83A4 : CD 13 81      call    CLR          ; zmaz obsah buffra DR
897/83A7 : D1          pop     d          ; obnov nasledujucu adresu
898/83A8 : 2A 30 C0      SUBTXT:  lhld    BUF          ; adresa buffra DR
899/83AB : 36 53          mvi     m,'S'        ; vypln opat prikazom SUB pre
900/83AD : 23          inx     h          ; pokracovanie vo vkladani dat do pamate
901/83AE : 36 55          mvi     m,'U'
902/83B0 : 23          inx     h
903/83B1 : 36 42          mvi     m,'B'
904/83B3 : 23          inx     h
905/83B4 : 36 20          mvi     m,' '
906/83B6 : 23          inx     h
907/83B7 : 7A          mov     a,d          ; vypis do buffra vyssi byte
908/83B8 : CD 3B 81      call    PREV02
909/83BB : 23          inx     h          ; posun ukazatel v Buffri
910/83BC : 3E 09          mvi     a,9        ; kuror sa nastavi za adresu
911/83BE : C3 37 81      jmp     PREV21      ; skoc vypisat do buffra este nizsi byte
912/83C1 :
913/83C1 : ;-----
914/83C1 : ; Obsluha klavesu WRK : zapis do klucovych klavesov Kx
915/83C1 : ; klaves WRK (CNT + Y)
916/83C1 : 3E FF      WRK:      mvi     a,-1      ; nastav priznak, ze sa bude zapisovat
917/83C3 : 32 35 C1      sta     SWRK        ; do klucovych klavesov
918/83C6 : C9          ret
919/83C7 :
920/83C7 : ;-----
921/83C7 : ; Obsluha klavesu PTL : zap/vyp vypisu obsahu DR na obrazovku
922/83C7 : ; klaves PTL (CNT + /)
923/83C7 : 21 37 C1      PTL:      lxi     h,SPTL      ; adresa priznaku PTL
924/83CA : 7E          mov     a,m          ; aktualny stav
925/83CB : 2F          cma     a          ; invertuj priznak
926/83CC : 77          mov     m,a          ; uloz novy stav
927/83CD : C9          ret
928/83CE :
929/83CE : ;-----
930/83CE : ; Vypocet adresy predlohy znaku. Predloha jedneho znaku ma 8 bytov.
931/83CE : ; Kedze sa znak vykresluje odspodu, vratena adresa ukazuje na nasledujuci znak.
932/83CE : ; Na rozdiel od Monitora PMD 85-2, nepouziva sa systemova premenna TASCII.
933/83CE : ; I: A=kod znaku
934/83CE : ; O: HL=predloha znaku+8

```

```

935/83CE : ; M: HL, DE, AF
936/83CE : 26 00      ADRAS:      mvi      h,0          ; kod znaku do HL
937/83D0 : 6F          mov      l,a
938/83D1 : 29          dad      h          ; x2
939/83D2 : 29          dad      h          ; x4
940/83D3 : 29          dad      h          ; x8
941/83D4 : EB          xchg
942/83D5 : FE 20      cpi      ' '          ; je to tlacitelny znak?
943/83D7 : DA EE 83    jc      ADRASU      ; ak nie, skoc
944/83DA : FE 60      cpi      ' '          ; znaky od kodu 60h?
945/83DC : D2 E4 83    jnc      ADRASM     ; ak ano, skoc
946/83DF : 21 08 85    lxi      h,A_Font_1-100h+8 ; font znakov 20h az 5Fh
947/83E2 : 19          dad      d          ; pripocitaj offset
948/83E3 : C9          ret
949/83E4 :
950/83E4 : FE 97      ADRASM:      cpi      97h          ; platny znak? (tu malo byt 96h)
951/83E6 : D2 EE 83    jnc      ADRASU      ; ak nie, skoc
952/83E9 : 21 C6 85    lxi      h,A_Font_2-300h+8 ; font znakov 60h az 95h
953/83EC : 19          dad      d          ; pripocitaj offset
954/83ED : C9          ret
955/83EE :
956/83EE : 21 6E 8A    ADRASU:      lxi      h,UNDEF      ; znak pre nedefinovane znaky
957/83F1 : C9          ret
958/83F2 :
959/83F2 : ;-----
960/83F2 : ; Tabulka mapovania klavesnice.
961/83F2 : ; Tabulka je v tvare:
962/83F2 : ; maska
963/83F2 : ; zoznam 7 kodov klavesov stlacenyh samostatne, ukonceny 0
964/83F2 : ; zoznam 7 kodov klavesov stlacenyh so SHF, ukonceny 0
965/83F2 : ; zoznam 7 kodov klavesov stlacenyh so CNT, ukonceny 0
967/83F2 : 01          KEYMAP:      db      01h
968/83F3 : 31 51 41 59 2D 50 13 00 db      '1','Q','A','Y','-', 'P',KDOWN,0
969/83FB : 21 71 61 79 3D 70 14 00 db      '!','q','a','y','=','p',KSDOWN,0
970/8403 : D0 80 8B 0B DA 89 0A 00 db      KK0,KSG0,KSG11,KWRK, KK10,KSG9,KINVR,0
971/840B : 02          db      02h
972/840C : 32 57 53 58 30 2F 11 00 db      '2','W','S','X','0','/',KUP,0
973/8414 : 22 77 73 78 5F 3F 09 00 db      '"','w','s','x','_','?',K SUP,0
974/841C : D1 81 8C 17 D9 0F 0C 00 db      KK1,KSG1,KSG12,KRCL, KK9,KPTL,KBGNL,0
975/8424 : 04          db      04h
976/8425 : 33 45 44 43 5C 3B 20 00 db      '3','E','D','C','\\',';',' ',0
977/842D : 23 65 64 63 5E 2B 20 00 db      '#','e','d','c','^','+', ' ',0
978/8435 : D2 82 8D 1E DB 94 20 00 db      KK2,KSG2,KSG13,KCD, KK11,KSG20, ' ',0
979/843D : 08          db      08h
980/843E : 34 52 46 56 38 39 40 00 db      '4','R','F','V','8','9','@',0
981/8446 : 24 72 66 76 28 29 60 00 db      '$','r','f','v','(',' )','`',0
982/844E : D3 83 8E 1F D7 D8 8A 00 db      KK3,KSG3,KSG14,KCLS, KK7, KK8,KSG10,0

```

```

983/8456 : 10                                db      10h
984/8457 : 35 54 47 42 2C 2E 08 00           db      '5','T','G','B',' ',' ','.',KLEFT,0
985/845F : 25 74 67 62 3C 3E 1A 00           db      '%','t','g','b','<','>',KSLEFT,0
986/8467 : D4 84 8F 1B 12 01 10 00           db      KK4,KSG4,KSG15,KCLL,KMON,KDELL,KBEEP,0
987/846F : 20                                db      20h
988/8470 : 36 5A 48 4E 4B 4C 18 00           db      '6','Z','H','N','K','L',KRIGHT,0
989/8478 : 26 7A 68 6E 6B 6C 19 00           db      '&','z','h','n','k','l',KSRIGHT,0
990/8480 : D5 85 90 1D 92 93 03 00           db      KK5,KSG5,KSG16,KDELT,KSG18,KSG19,KENDL,0
991/8488 : 40                                db      40h
992/8489 : 37 55 4A 4D 49 4F 3A 00           db      '7','U','J','M','I','O',':',0
993/8491 : 27 75 6A 6D 69 6F 2A 00           db      "','','u','j','m','i','o','*',0
994/8499 : D6 86 91 1C 87 88 95 00           db      KK6,KSG6,KSG17,KINST,KSG7,KSG8,KSG21,0
1025/84A1 :
1026/84A1 : ;-----
1027/84A1 : ; Test klavesnice s cakanim. Rutina vola rutinu SCAN.
1028/84A1 : ; I: [SBEEP]=priznak povolenia pipnutia
1029/84A1 : ; O: A=ASCII kod klavesu
1030/84A1 : ; M: AF
1031/84A1 : C5                                INKLAV:    push    b                ; odpamataj pouzite registre
1032/84A2 : E5                                push    h
1033/84A3 : 21 BD C0                          lxi      h,KeyRepeat    ; adresa hodnoty pre opakovanie klavesu
1034/84A6 : 4E                                mov      c,m            ; prevezmi do C
1035/84A7 : 36 32                          mvi      m,50           ; a inicializuj na 50
1036/84A9 : 2B                                dcx      h              ; posun sa na posledny stlaceny klaves
1037/84AA : 3E 03                          InKlavR:  mvi      a,3      ; casova konstanta medzi testami
1041/84AC : CD 20 92                        call     ProcLanReq     ; spracuj poziadavku siete
1043/84AF : CD 03 8C                        call     SCAN           ; otestuj klavesnicu
1044/84B2 : 46                                mov      b,m            ; posledne stlaceny klaves do B
1045/84B3 : 77                                mov      m,a            ; uloz aktualny kod klavesu
1046/84B4 : CA AA 84                        jz       InKlavR        ; ak nebolo nic stlacene, cakaj dalej
1047/84B7 : B8                                cmp      b              ; je to rovnaky klaves, ako naposledy?
1048/84B8 : C2 27 8C                        jnz      InKlavN        ; nie, skoc dalej
1049/84BB : 0D                                dcr      c              ; zniz pocitadlo pre opakovanie
1050/84BC : C2 AA 84                        jnz      InKlavR        ; dalsi test, kym bude klaves akcept.
1051/84BF : 23                                inx      h              ; posun sa na zdrzanie
1052/84C0 : C3 25 8C                        jmp      InKlavS        ; a skoc dalej
1053/84C3 :
1054/84C3 : ;-----
1055/84C3 : ; Pokracovanie testu klavesnice - rutiny SCAN.
1056/84C3 : ; I: A=bitova maska stlpca stlaceneho klavesu, B=cislo riadku
1057/84C3 : ; O: A=kod stlaceneho klavesu
1058/84C3 : 4F                                ScanM:    mov      c,a      ; odloz masku do C
1059/84C4 : 17                                ral                          ; je to klaves EOL
1060/84C5 : 3E 0D                          mvi      a,KEOL        ; A=kod klavesu EOL
1061/84C7 : DA F5 84                        jc       ScanE1         ; ak ano, skoc
1062/84CA : 78                                mov      a,b            ; invertuj cislo stlpca v rozsahu <7,1>
1063/84CB : 2F                                cma

```

```

1064/84CC : E6 07          ani    07h          ; na rozsah <0,6>
1065/84CE : 47             mov    b,a          ; a uloz
1066/84CF : E5             push   h            ; odpamataj HL a DE
1067/84D0 : D5             push   d
1068/84D1 : 11 19 00       lxi    d,25          ; offset na data pre jeden riadok tabulky
1069/84D4 : 21 D9 83       lxi    h,KEYMAP-25    ; HL=adresa klavesnicovej tabulky
1070/84D7 : 19             ScanC: dad    d          ; posun adresu na dalsi riadok
1071/84D8 : 7E             mov    a,m          ; maska stlpca do A
1072/84D9 : B7             ora     a            ; radoby test konca tabulky - zbytocne
1073/84DA : CA F3 84       jz     ScanE3        ; skoc, ak je koniec tabulky
1074/84DD : B9             cmp    c            ; je to spravny stlpec?
1075/84DE : C2 D7 84       jnz    ScanC          ; nie, hladaj dalej
1076/84E1 : 23             inx     h            ; preskoc masku v tabulke
1077/84E2 : DB F6             in     SYS55C        ; precitaj stav CNT a SHF
1078/84E4 : 2F             cma          ; invertuj
1079/84E5 : 1F             rar          ; posun ich stav do bitov 4 (CNT)
1080/84E6 : 1F             rar          ; a 3 (SHF)
1081/84E7 : E6 18          ani    18h          ; odmaskuj, tym vznikne offset 0, 8, 16
1082/84E9 : FE 18          cpi    18h          ; ak nie su stlacene oba naraz,
1083/84EB : C2 EF 84       jnz    ScanV          ; skoc dalej
1084/84EE : AF             xra     a            ; ak su stlacene oba, ignoruje sa to
1085/84EF : 80             ScanV: add    b            ; pripocitaj upravene cislo riadku
1086/84F0 : 5F             mov    e,a          ; DE=offset na dany kod klavesu
1087/84F1 : 19             dad    d            ; pripocitaj ho
1088/84F2 : 7E             mov    a,m          ; a prevezmi kod klavesu do A
1089/84F3 : D1             ScanE3: pop    d          ; obnov DE
1090/84F4 : E1             ScanE2: pop    h          ; obnov HL
1091/84F5 : C1             ScanE1: pop    b          ; obnov BC
1092/84F6 : 32 34 C1       sta    ASCII        ; uloz kod stlaceneho klavesu do ASCII
1093/84F9 : B7             ora     a            ; Z=0
1094/84FA : C9             ret
1095/84FB :
1096/84FB : ;-----
1097/84FB : ; Vypis medzery na obrazovku.
1098/84FB : ; I: -
1099/84FB : ; O: -
1100/84FB : ; M: AF
1101/84FB : 3E 20       SPACE: mvi    a,' '          ; vypis medzeru
1102/84FD : 11             db     11h          ; lxi d,NN - preskoc nasl. instrukciu
1103/84FE :             ; pokracuj PRTOUT
1104/84FE :
1105/84FE : ;-----
1106/84FE : ; Odriadkovanie.
1107/84FE : ; Ak je kurzor na najspodnejšiom riadku, vykona sa odrolovanie obrazovky.
1108/84FE : ; I: -
1109/84FE : ; O: -
1110/84FE : ; M: AF

```



```

1111/84FE : 3E 0D
1112/8500 :
1113/8500 :
1114/8500 :
1115/8500 :
1116/8500 :
1117/8500 :
1118/8500 :
1119/8500 : C5
1120/8501 : D5
1121/8502 : E5
1122/8503 : 21 75 85
1123/8506 : E5
1124/8507 : FE 0D
1125/8509 : C2 23 85
1126/850C : 2A 3E C0
1127/850F : 3A FA C0
1128/8512 : BC
1129/8513 : DA 88 80
1130/8516 : 7D
1131/8517 : E6 C0
1132/8519 : 6F
1133/851A : EB
1134/851B : 2A FC C0
1135/851E : 19
1136/851F : 22 3E C0
1137/8522 : C9
1138/8523 :
1139/8523 : FE 1C
1140/8525 : CA A7 85
1141/8528 : FE 0A
1142/852A : C8
1143/852B : A7
1144/852C : C8
1145/852D : 21 3E C0
1146/8530 : 35
1147/8531 : FE 08
1148/8533 : C8
1149/8534 :
1150/8534 :
1151/8534 :
1152/8534 :
1153/8534 : 34
1154/8535 : 4F
1155/8536 : 7E
1156/8537 : E6 3F
1157/8539 : 47

PRTCR:          mvi      a,CR          ; odriadkuj
                ; pokracuj PRTOUT

;-----
; Vypis znaku na obrazovku.
; I: A=kod znaku
; O: -
; M: AF
PRTOUT:          push     b            ; odpamataj registre
                push     d
                push     h
                lxi      h,ADRRET      ; navratova adresa, ktora obnovi
                push     h            ; registre, na zasobnik
                cpi      CR           ; je to CR - novy riadok?
                jnz      PrtOutA      ; skoc, ak nie
                lhld     CURSOR       ; adresa kurzora
                lda      VYSKA        ; vyssi byte VRAM posledneho zR
                cmp      h            ; je kurzor na poslednom zR?
                jc       ROLL         ; ak ano, bude sa scrollovat
                mov      a,l          ; inak, uprav adresu na zaciatok riadku
                ani      0C0h
                mov      l,a
                xchg      ; presun do DE
                lhld     RADSIR       ; offset na dalsi zR
                dad      d            ; pripocitaj
                shld     CURSOR       ; a uloz novu adresu kurzora
                ret

PrtOutA:         cpi      CLS         ; je to CLS - vymazanie obrazovky?
                jz       ERASE        ; ak ano, skoc ju vymazat
                cpi      LF           ; je to LF - Line-Feed?
                rz       ; ak ano, ignoruj ho
                ana      a            ; je to NUL?
                rz       ; ak ano, ignoruj ho
                lxi      h,CURSOR     ; adresa kurzora
                dcr      m            ; zniz adresu
                cpi      BS           ; je to BS - o znak nazad?
                rz       ; ak ano, vrat sa
                ; BACKSPACE na zaciatku riadku umiestni kurzor "vedla" VRAM. kde su systemove
                ; premenne Monitora, pripadne Basicu.
                ; Pri vypisani znaku do tejto casti pamati je nedefinovatelny a s najvacsou
                ; pravdepodobnostou skonci krachom!
                inr      m            ; vrat kurzor nazad
                mov      c,a          ; kod znaku do C
                mov      a,m          ; pozicia v riadku
                ani      3Fh
                mov      b,a          ; do B

```

```

1158/853A : 3A FB C0          lda    ENLNW          ; pocet viditelnych znakov na riadok
1159/853D : B8                cmp    b              ; uz zme na konci riadku?
1160/853E : CC FE 84          cz      PRTCR          ; ak ano, najprv odriadkuj
1161/8541 : 79                mov    a,c            ; kod znaku opat do A
1162/8542 : 21 3E C0          PRT02: lxi    h,CURSOR      ; adresa kurzora
1163/8545 : 34                inr    m              ; posun na dalsiu poziciu
1164/8546 : 2A 3E C0          lhld   CURSOR          ; adresa VRAM kurzora
1165/8549 : 2B                dcx    h              ; vrat povodnu poziciu
1166/854A :                  ; pokracuj v WRCHAR
1167/854A :
1168/854A :                  ;-----
1169/854A :                  ; Vypisanie znaku na pozicii v HL.
1170/854A :                  ; Rutinu nie je mozne pouzit s prerusenim, pretoze pouziva SP ako bezny register!
1171/854A :                  ; I: A=kod znaku, HL=adresa VRAM (2 uR pod znakom), [COLOR]=atribut
1172/854A :                  ; O: -
1173/854A :                  ; M: AF
1174/854A : C5                WRCHAR: push   b          ; odpamataj registre
1175/854B : D5                push   d
1176/854C : E5                push   h
1177/854D : CD CE 83          call   ADRAS          ; vypocitaj adresu predlohy znaku
1178/8550 : EB                xchg                    ; do DE
1179/8551 : 21 00 00          lxi    h,0            ; odpamataj adresu zasobnika
1180/8554 : 39                dad    sp
1181/8555 : 22 3E C1          shld   STACK
1182/8558 : 3A 3A C0          lda    COLOR          ; A=atribut
1183/855B : E1                pop    h              ; obnov adresu VRAM
1184/855C : E5                push   h              ; a zase uloz
1185/855D : 31 C0 FF          lxi    sp,-64          ; offset o 1 uR hore
1186/8560 : 01 80 FF          lxi    b,-128          ; posun adresu VRAM o 2 uR hore
1187/8563 : 09                dad    b
1188/8564 : 47                mov    b,a            ; B=atribut
1189/8565 : 0E 08          mvi    c,8            ; znak ma vysku 8
1190/8567 : 1B                dcx    d              ; posun ukazatel na predlohu znaku
1191/8568 : 1A                ldax   d              ; vezmi byte predlohy
1192/8569 : A8                xra    b              ; pripoj atribut
1193/856A : 77                mov    m,a            ; a zapis do VRAM
1194/856B : 39                dad    sp              ; posun adresu VRAM o 1 uR hore
1195/856C : 0D                dcr    c              ; opakuj 8x
1196/856D : C2 67 85          jnz    WrCharL
1197/8570 : 70                mov    m,b            ; vymaz byte nad znakom
1198/8571 : 2A 3E C1          lhld   STACK          ; obnov adresu zasobnika
1199/8574 : F9                sphl
1200/8575 : E1                ADRRET: pop    h          ; obnov vsetky registre
1201/8576 : D1                pop    d
1202/8577 : C1                ADRR1T: pop    b
1203/8578 : C9                ret
1204/8579 :                  ; a navrat

```

```

1205/8579 : ;-----
1206/8579 : ; Spustenie prikazu Monitora
1207/8579 : CD 24 8F SYST1:      call    SSKIP          ; preskoc medzery za prikazom
1208/857C : 22 72 C0          shld    CURCH          ; uloz novy ukazatel na zadany text
1209/857F : 13              inx     d              ; preskoc ukonc. 0 za prikazom v tabul.
1210/8580 : EB              xchg                    ; presun do DE
1211/8581 : C3 66 8C          jmp     JmpIndHL          ; skoc na adresu obsluhy prikazu
1212/8584 : ;-----
1213/8584 : ; Povodna adresa pre vypisanie znaku z Monitora PMD 85-1.
1214/8584 : ; (8584h)
1215/8584 : ;-----
1216/8584 : C3 4A 85 WRCHAROLD:  jmp     WRCHAR
1217/8587 : ;-----
1218/8587 : ; Obsluha prikazu MEM : vypisanie obsahu pamate do DR
1219/8587 : ; Format: MEM aaaa
1220/8587 : ;-----
1221/8587 : CD 4B 81 MEM:      call    RADR          ; precitaj adresu pre vypis
1222/858A : CD A8 83          call    SUBTXT        ; priprav text "SUB adresa"
1223/858D : 06 10          mvi     b,16          ; vypise sa 16 bytov pamate
1224/858F : C5          MemL:    push    b              ; odpamataj pocitadlo
1225/8590 : 23              inx     h              ; posun ukazatel buffra DR
1226/8591 : 36 20          mvi     m,' '          ; zapis medzeru
1227/8593 : 23              inx     h              ; posun ukazatel buffra DR
1228/8594 : 1A              ldax    d              ; byte z pamate
1229/8595 : CD 3B 81          call    PREV02        ; priprav do DR
1230/8598 : 13              inx     d              ; dalsi byte
1231/8599 : C1              pop     b              ; obnov pocitadlo
1232/859A : 05              dcr     b              ; opakuj 16x
1233/859B : C2 8F 85          jnz     MemL
1234/859E : C9              ret
1235/859F : ; po navrate sa vypise pripraveny text
1236/859F : ;-----
1237/859F : ; Obsluha klavesu INVR : Zap/vyp inverzneho zobrazenia a zmazanie obrazovky
1238/859F : ; klaves INVR (CNT + sipka dole)
1239/859F : 3A 3A C0 MODNEG:    lda     COLOR          ; atribut
1240/85A2 : EE 3F          xri     3Fh          ; invertuj
1241/85A4 : 32 3A C0          sta     COLOR          ; a uloz
1242/85A7 : ; pokracuj v ERASE
1243/85A7 : ;-----
1244/85A7 : ; Vymazanie obrazovky.
1245/85A7 : ; Rutina vyuziva register SP ako cielovy ukazatel do VRAM a tak tato rutina
1246/85A7 : ; sa nesmie pouzivat s povoleny prerusenim.
1247/85A7 : ; Obsluha klavesu CLS : Vymazanie obrazovky okrem DR.
1248/85A7 : ; klaves CLS (CNT + V)
1249/85A7 : ;-----
1250/85A7 : 01 80 C2 ERASE:      lxi     b,VRAM+0280h    ; adresa prveho znaku vo VRAM
1251/85AA : 11 30 C0          lxi     d,VRAM+48          ; adresa konca uR+1

```

1252/85AD : 2A FC C0	lhld	RADSIR	; offset na dalsi zR
1253/85B0 : 09	dad	b	; HL=adresa kurzora po zmazani obrazovky
1254/85B1 : 06 00	mvi	b,0	; 256 uR
1255/85B3 :			
1256/85B3 :			
1257/85B3 :			;-----
1258/85B3 :			; Vymazanie casti obrazovky.
1259/85B3 :			; I: HL=adresa kurzora, DE=adresa konca najvyssieho uR, B=pocet mazanych uR
1260/85B3 :			; O: -
1261/85B3 : 22 3E C0			; M: vsetky
1262/85B6 : 3A 3A C0	EraLines:	shld	CURSOR ; uloz novu adresu kurora
1263/85B9 : 21 00 00		lda	COLOR ; atribut
1264/85BC : 39		lxi	h,0 ; odpamataj adresu zasobnika
1265/85BD : EB		dad	sp
1266/85BE : F9		xchg	; do DE
1267/85BF : 4F		sphl	; SP=cielova adresa VRAM
1268/85C0 : 78		mov	c,a ; C=atribut
1269/85C1 : 41		mov	a,b ; A=pocet mazanych uR
1270/85C2 :	EraLinesL:	mov	b,c ; B=atribut
1271/85C2 :		rept	24 ; 24x2 mazanych bytov v uR
1272/85C2 :		push	b
1271/85C2 : C5		endm	
1271/85C3 : C5		push	b
1271/85C4 : C5		push	b
1271/85C5 : C5		push	b
1271/85C6 : C5		push	b
1271/85C7 : C5		push	b
1271/85C8 : C5		push	b
1271/85C9 : C5		push	b
1271/85CA : C5		push	b
1271/85CB : C5		push	b
1271/85CC : C5		push	b
1271/85CD : C5		push	b
1271/85CE : C5		push	b
1271/85CF : C5		push	b
1271/85D0 : C5		push	b
1271/85D1 : C5		push	b
1271/85D2 : C5		push	b
1271/85D3 : C5		push	b
1271/85D4 : C5		push	b
1271/85D5 : C5		push	b
1271/85D6 : C5		push	b
1271/85D7 : C5		push	b
1271/85D8 : C5		push	b
1271/85D9 : C5		push	b
1273/85DA : 31 40 00		lxi	sp,64 ; SP na dalsi uR
1274/85DD : 39		dad	sp

```

1275/85DE : F9          sphl
1276/85DF : 3D          dcr      a          ; opakuj pre A uR
1277/85E0 : C2 C2 85    jnz      EraLinesL
1278/85E3 : EB          xchg
1279/85E4 : F9          sphl          ; obnov povodnu adresu zasobnika
1280/85E5 : C9          ret
1281/85E6 :
1282/85E6 : ;-----
1283/85E6 : ; Test rozsvietenia bodu.
1284/85E6 : ; I: [X1], [Y1]=suradnice bodu
1285/85E6 : ; O: A=0/1 - bod ne/svieti
1286/85E6 : ; M: vsetky
1287/85E6 : ; (85E6h)
1288/85E6 : 2A 71 C1    RPOINT:      lhld      Y1-1          ; H=Y1
1289/85E9 : 3A 70 C1    lda      X1
1290/85EC : 6F          mov      l,a          ; L=X1
1291/85ED : CD 94 8C    call     POSPOINT      ; vypocitaj adresu a poziciu bodu
1292/85F0 : 7E          mov      a,m          ; vezmi byte VRAM
1293/85F1 : A0          ana      b          ; uplatni masku
1294/85F2 : C8          rz          ; navrat, ak nesvieti
1295/85F3 : 3E 01      mvi      a,1          ; A=1 - bod svieti
1296/85F5 : C9          ret
1297/85F6 :
1298/85F6 : ;-----
1299/85F6 : ; Obsluha klavesu BGNL : Nastavenie kurzora na zaciatok DR.
1300/85F6 : ; klaves BGNL (CNT + sipka hore)
1301/85F6 : AF          HOME:      xra      a          ; vunuluj
1302/85F7 : 32 38 C0      sta      RPOS          ; poziciu v buffri DR
1303/85FA : 32 36 C0      sta      DCUR          ; poziciu kurzora v DR
1304/85FD : C9          ret
1305/85FE :
1306/85FE : ;-----
1307/85FE : 00 00          dw      0
1308/8600 :
1309/8600 : ;-----
1310/8600 : ; Font znakov s kodmi 20h az 5Fh.
1312/8600 : A_Font_1:      binclude "font-1.bin"
1316/8800 :
1317/8800 : ;-----
1318/8800 : ; Vykonanie zmien v editacnom riadku.
1319/8800 : C5          KEYOUT:      push     b          ; odpamataj registre
1320/8801 : D5          push     d
1321/8802 : E5          push     h
1322/8803 : 21 75 85      lxi      h,ADRRET      ; navratova adresa, ktora obnovi
1323/8806 : E5          push     h          ; registre, na zasobnik
1324/8807 : CD 43 8C      call     DOKEY          ; vykonaj zmeny v DR
1325/880A : C3 55 88      jmp      WRBUFF          ; vypis novy stav do DR

```

```

1326/880D :
1327/880D :
1328/880D : ;-----
1329/880D : ; Obsluha klavesu INST : Vsunutie medzery v mieste kurzora
1330/880D : ; klaves INST (CNT + M)
1331/8810 : CD 5E 81 INS:      call  CURPOS      ; adresa kurzora do HL
1332/8811 : CD 46 8B      mov    c,l      ; pozicia X do L
1333/8814 : 54          call  ENDLN      ; HL=adresa posledneho znaku v DR
1334/8815 : 5D          mov    d,h      ; skopiruj aj do DE
1335/8816 : 13          mov    e,l      ;
1336/8817 : 79          inx    d      ; +1
1337/8818 : BB          mov    a,c      ; vojde sa este medzera?
1338/8819 : D0          cmp    e      ;
1339/881A : 7E          rnc    e      ; ak nie, navrat
1340/881B : 12          InsL:  mov    a,m      ; posunutie casti textu o 1 znak doprava
1341/881C : 79          stax    d      ;
1342/881D : BD          mov    a,c      ; az po aktualnu poziciu
1343/881E : 2B          cmp    l      ;
1344/881F : 1B          dcx    h      ; zniz ukazatele
1345/8820 : C2 1A 88    dcx    d      ;
1346/8823 : 23          jnz    InsL      ;
1347/8824 : 36 20      inx    h      ; vrat sa na miesto kurzora
1348/8826 : C9          mvi    m,' '      ; a zapis tam medzeru
1349/8827 : ret
1350/8827 :
1351/8827 : ;-----
1352/8827 : ; Obsluha programovych klucov K0 az K11.
1353/8827 : ; klaves K0 (CNT + 1) az K11 (CNT+\)
1354/882A : 4F          KEY:      lda    ASCII      ; kod programoveho klavesu
1355/882B : 3A 35 C1      mov    c,a      ; do C
1356/882E : B7          lda    SWRK      ; bol stlaceny pred tym WRK?
1357/882F : C2 C2 8A      ora    a      ; ak ano,
1358/8832 : 32          jnz    SETWRK      ; skoc ulozit obsah DR do progr. kluca
1359/8833 :          db    32h      ; STA nn - preskoc nasl. instrukciu
1360/8833 :          ; (tu by bolo spravnejsie LDA nn)
1361/8833 :          ; pokracuj v RCL
1362/8833 :
1363/8833 : ;-----
1364/8833 : ; Obsluha klavesu RCL : Vyvolanie naposledy odoslaného riadku
1365/8833 : ; klaves RCL (CNT + X)
1366/8833 : ; Do programoveho kluca mozno ulozit aj text so semigrafickymi znakmi, ale
1367/8833 : ; vyvolanie takehoto textu nebude uspesne. Vkladanie textu do DR skonci po
1368/8833 : ; narazení na prvý semigrafický znak, ktorý sa už nevyíše.
1369/8835 : 2A 7C C1 RCL:      mvi    c,XRCL      ; kod kluca RCL
1370/8838 : 2B          lhld    KEYEND      ; adresa vrcholu tabulky klucov
1371/8839 : 7E          RclS:  dcx    h      ; -1
1372/883A : B7          mov    a,m      ; prevezmi byte
          ora    a      ; koniec tabulky?

```

```

1373/883B : C8                rz                ; ano, navrat - nenajdeny, ignorovat
1374/883C : B9                cmp      c                ; je to hladany kluc?
1375/883D : C2 38 88          jnz      RclS            ; nie, hladaj dalej
1376/8840 :                   ; kluc najdeny
1377/8840 : 2B                RclL:   dcx      h                ; posun ukazatel
1378/8841 : 7E                mov      a,m            ; vezmi znak z textu kluca
1379/8842 : B7                ora      a                ; koniec kluca?
1380/8843 : C8                rz                ; ano, navrat
1381/8844 : 32 34 C1          sta      ASCII           ; uloz znak
1382/8847 : F2 4D 88          jp       RclC            ; skoc pre ASCII znak
1383/884A :                   ; tato podmienka zamedzi vloženie semigrafických znakov z programoveho kluca
1384/884A : FE E0            cpi      0E0h            ; je to kod kluca?
1385/884C : D8                rc                ; ak ano, navrat
1386/884D : E5                RclC:   push     h                ; odpamataj ukazatel v progr. klucoch
1387/884E : CD 43 8C          call    DOKEY           ; vlož dany znak do buffra DR
1388/8851 : E1                pop      h                ; obnov ukazatel v progr. klucoch
1389/8852 : C3 40 88          jmp     RclL            ; pokračuj ďalším znakom v kluci
1390/8855 :
1391/8855 :                   ;-----
1392/8855 :                   ; Vypis obsahu buffra DR.
1393/8855 : 2A 38 C0          WRBUFF:  lhld     RPOS                ; prvý vypisovaný znak
1394/8858 : EB                WRLINE:  xchg                     ; do DE
1395/8859 : 2A 32 C0          lhld     DIAL                ; adresa VRAM vypisu obsahu DR
1396/885C : 3A FE C0          lda      DLZR                ; max. počet zobrazených znakov v riadku
1397/885F : 47                mov      b,a                ; do B
1398/8860 : 1A                WrLineL: ldax     d                ; prevezmi znak
1399/8861 : CD 4A 85          call    WRCHAR           ; vypis ho
1400/8864 : 23                inx      h                ; posun ukazatele
1401/8865 : 13                inx      d
1402/8866 : 05                dcr      b                ; opakuj N krát
1403/8867 : C2 60 88          jnz     WrLineL
1404/886A : C9                ret
1405/886B :
1406/886B :                   ;-----
1407/886B :                   ; Vystupna sestnastkova konverzia 4 bity.
1408/886B :                   ; Cislo 0 az 15 sa transformuje na znaky 0-9A-F.
1409/886B :                   ; I: A=cislo <0,15>
1410/886B :                   ; O: A=znak 0-9A-F
1411/886B :                   ; M: AF
1412/886B : E6 0F            PRE00:  ani      0Fh                ; odmaskuj iba dolne 4 bity
1413/886D : FE 0A            cpi      10                ; je cislo <10?
1414/886F : DE 2F            sbi      2Fh                ; uprav hodnotu podľa príznakov
1415/8871 : 27                daa                     ; a transformuj na znak 0-9A-F
1416/8872 : C9                ret
1417/8873 :
1418/8873 :                   ;-----
1419/8873 :                   ; Obsluha príkazu MGLD : nahratie suboru z MGF.

```

```

1420/8873 : ; Format: MGLD nn
1421/8873 : 3A 36 C1 MGLD:          lda      MGCHAR          ; typ suboru
1422/8876 : C3 7A 88          jmp      MgLdT
1423/8879 :
1424/8879 : ;-----
1425/8879 : ; Obsluha prikazu MGEND : kontrola suboru z MGF.
1426/8879 : ; Format: MGEND nn
1427/8879 : AF MGEND:          xra      a          ; akykolvek typ suboru
1428/887A : 32 B1 C1 MgLdT:          sta      FINDTP          ; uloz hladany typ suboru
1429/887D : CD 3E 8E          call    DECPAR          ; nacistaj zadane cislo suboru
1430/8880 : DA 49 80          jc      PRBWTXT          ; ak bolo chybne, vypis "ERROR IN DATA"
1431/8883 : 32 B0 C1          sta      FINDNR          ; uloz hladane cislo
1432/8886 : CD E2 8D          call    SHEAD          ; vyhľadaj a nacistaj subor
1433/8889 : C2 46 80          jnz     SYST3          ; skoc pri chybe
1434/888C : DA 46 80          jc      SYST3          ; skoc pri chybe (zbytocne)
1435/888F : 3A B1 C1          lda      FINDTP          ; ak bol pozadovany akykolvek typ,
1436/8892 : B7          ora      a          ; teda to bol prikaz MGEND,
1437/8893 : CA 46 80          jz      SYST3          ; tak sa hned vrat do slucky Monitora
1438/8896 : C3 6E 8A          jmp      MgLdA          ; inak este over pripadny Autostart
1439/8899 :
1440/8899 : ;-----
1441/8899 : 00 00          db      0,0
1442/889B :
1443/889B : ;-----
1444/889B : E5 PRPTXT:          push     h
1445/889C :          ; pokračuj v CD
1446/889C :
1447/889C : ;-----
1448/889C : ; Obsluha klavesu C-D : Vypisanie posledneho hlasenia do DR
1449/889C : ; klaves C-D (CNT + C)
1450/889C : E1 CD:          pop      h
1451/889D : 2A 3A C1          lhld    CDBUF          ; HL=adresa textu
1452/88A0 : C3 58 88          jmp      WRLINE          ; vypis text
1453/88A3 :
1454/88A3 : ;-----
1455/88A3 : ; Vseobecne pipnutie.
1456/88A3 : 2A 30 C1 BEEP:          lhld    BEEDT          ; HL=adresa dat pipnutia
1457/88A6 :          ; pokračuj v BELL
1458/88A6 :
1459/88A6 : ;-----
1460/88A6 : ; Zahratie "melodie" podľa dat.
1461/88A6 : ; Format: [dlzka,vyska]+,-1
1462/88A6 : ; I: HL=adresa dat
1463/88A6 : ; O: -
1464/88A6 : ; M: HL, DE, B, AF
1465/88A6 : 7E BELL:          mov      a,m          ; prevezmi byte dlzky tonu
1466/88A7 : FE FF          cpi      -1          ; koniec tabulky?

```



```

1467/88A9 : C8                rz                ; ano, navrat
1468/88AA : 47                mov      b,a        ; uloz dlzku do B
1469/88AB : 23                inx      h          ; posun ukazatel
1470/88AC : 56                mov      d,m        ; vyska tonu do D
1471/88AD : 23                inx      h          ; posun ukazatel
1472/88AE : CD 70 81          call    Ton        ; zahraj ton
1473/88B1 : C3 A6 88          jmp     BELL       ; opakuje pre dalsi ton
1474/88B4 :
1475/88B4 : ;-----
1476/88B4 : 00                db      0
1477/88B5 :
1478/88B5 : ;-----
1479/88B5 : ; Zdrzanie podla DE.
1480/88B5 : ; I: -
1481/88B5 : ; O: DE=0
1482/88B5 : ; M: DE, F
1483/88B5 : ; T: D * 1,88 ms + E * 7,32 us
1484/88B5 : 1D                DelayDE: dcr      e
1485/88B6 : C2 B5 88          jnz     DelayDE
1486/88B9 : 15                dcr      d
1487/88BA : C2 B5 88          jnz     DelayDE
1488/88BD : C9                ret
1489/88BE :
1490/88BE : ;-----
1491/88BE : ; Font znakov s kodmi 60h az 95h.
1493/88BE : A_Font_2:      binclude "font-2.bin"
1497/8A6E : UNDEF:                ; adresa pre nedefinovany znak
1498/8A6E :
1499/8A6E : ;-----
1500/8A6E : ; Test na Auto start nahraného programu.
1501/8A6E : ; I: [NAMFIL]=prvy znak mena suboru
1502/8A6E : ; O: Z=auto start, NZ=bez startu
1503/8A6E : 3A B8 C1          MgLdA:   lda      NAMFIL      ; prvý znak mena suboru
1504/8A71 : FE 2A                cpi      '*'        ; je to '*'?
1505/8A73 : C3 3F 83          jmp     MgLdR      ; skoc dalej
1506/8A76 :
1507/8A76 : ;-----
1508/8A76 : ; Priprava textu "FILE ERROR" pre prikaz MGLD a MGEND.
1509/8A76 : 21 5C 83          MgLdMsg:  lxi      h,T_FileErr    ; priprav spravu "FILE ERROR"
1510/8A79 : 22 74 C0          shld    MESS
1511/8A7C : C9                ret
1512/8A7D :
1513/8A7D : ;-----
1514/8A7D : ; Vykreslenie/zmazanie kurzora.
1515/8A7D : 2A 3E C0          XCHCUR:  lhld    CURSOR      ; adresa kurzora
1516/8A80 : 01 C0 FF          XCCUR:  lxi      b,-64        ; offset -1 uR
1517/8A83 : 09                dad     b

```

```

1518/8A84 : 7E          mov     a,m          ; byte VRAM
1519/8A85 : EE 3E        xri      3Eh          ; vykresli/zmaz kurzor
1520/8A87 : 77          mov     m,a          ; a zapis do VRAM
1521/8A88 : C9          ret
1522/8A89 :
1523/8A89 : ;-----
1524/8A89 : ; Pipnutie a vypis hlasenia.
1525/8A89 : ; I: [MESS]=adresa textu hlasenia
1526/8A89 : ; O: -
1527/8A89 : ; M: vsetky
1528/8A89 : CD A3 88      PRTTEXT:  call    BEEP
1529/8A8C : C3 36 8B      jmp     PRMTXT
1530/8A8F :
1531/8A8F : ;-----
1532/8A8F : ; Obsluha prikazu SUB : zapis dat do pamate.
1533/8A8F : ; Format: SUB aaaa data
1534/8A8F : ; Data mozu byt dvojice hexadecimalnych cislic alebo text v apostrofoch.
1535/8A8F : 21 91 82      SUB:      lxi      h,T_ErrData  ; priprav spravu "ERROR IN DATA"
1536/8A92 : 22 74 C0      shld     MESS
1537/8A95 : CD 4B 81      call    RADR          ; precitaj adresu do DE
1538/8A98 : CD 24 8F      SubL:    call    SSKIP         ; preskoc medzery
1539/8A9B : FE 0D        cpi      CR          ; koniec textu?
1540/8A9D : CA A3 83      jz       SUB03        ; ano, skoc vypisat dalsiu adresu
1541/8AA0 : FE 27        cpi      ""         ; je to text v apostrofoch?
1542/8AA2 : CA B0 8A      jz       SubA         ; ano, skoc
1543/8AA5 : CD F7 80      call    PAIRIN        ; precitaj sestnastkove cislo
1544/8AA8 : DA 49 80      jc       PRBWTXT      ; bolo chybne, skoc vypisat chybu
1545/8AAB : 12          stax     d          ; uloz hodnotu
1546/8AAC : 13          inx      d          ; posun ukazatel
1547/8AAD : C3 98 8A      jmp     SubL          ; pokračuj s dalsim vstupom
1548/8AB0 :
1549/8AB0 : 23          SubA:    inx      h          ; preskoc apostrof
1550/8AB1 : 7E          SubS:    mov     a,m          ; vezmi dalsi byte
1551/8AB2 : FE 0D        cpi      CR          ; koniec textu?
1552/8AB4 : 23          inx      h
1553/8AB5 : CA A3 83      jz       SUB03        ; ano, skoc vypisat dalsiu adresu
1554/8AB8 : FE 27        cpi      ""         ; ukoncovaci apostrof?
1555/8ABA : CA 98 8A      jz       SubL         ; ano, dalsi vstup
1556/8ABD : 12          stax     d          ; uloz znak
1557/8ABE : 13          inx      d          ; posun cielovy ukazatel
1558/8ABF : C3 B1 8A      jmp     SubS          ; a pokračuj dalsim znakom
1559/8AC2 :
1560/8AC2 : ;-----
1561/8AC2 : ; Vlozenie textu do tabulky klucov.
1562/8AC2 : ; I: C=kod kluca
1563/8AC2 : ; O: -
1564/8AC2 : ; M: vsetky

```

1565/8AC2 : CD A3 88	SETWRK:	call	BEEP	; pipni
1566/8AC5 : CD 5E 81		call	CURPOS	; adresa kurzora v editacnom riadku
1567/8AC8 : 45	SetWrkI:	mov	b,l	; dlzka textu do B
1568/8AC9 : 21 7C 83		lxi	h,T_MemOF	; priprav spravu "MEMORY OVERFLOW"
1569/8ACC : 22 74 C0		shld	MESS	
1570/8ACF : 2A 7C C1		lhld	KEYEND	; adresa vrcholu tabulky klucov
1571/8AD2 : 2B	SetWrkS:	dcx	h	; zniz ukazatel
1572/8AD3 : 7E		mov	a,m	; vezmi byte tabulky klucov
1573/8AD4 : B7		ora	a	; konice tabulky?
1574/8AD5 : CA F6 8A		jz	SetWrkJ	; skoc, ak ano
1575/8AD8 : B9		cmp	c	; je to hladany kluc?
1576/8AD9 : C2 D2 8A		jnz	SetWrkS	; ak nie, hladaj dalej
1577/8ADC : 54		mov	d,h	; adresa najdeného kluca do DE
1578/8ADD : 5D		mov	e,l	
1579/8ADE : 2B	SetWrkK:	dcx	h	; zniz ukazatel
1580/8ADF : 7E		mov	a,m	; a hladaj koniec textu kluca
1581/8AE0 : B7		ora	a	; koniec tabulky?
1582/8AE1 : CA F7 8A		jz	SetWrkH	; ano, skoc dalej
1583/8AE4 : F2 DE 8A		jp	SetWrkK	; je to znak, preskoc
1584/8AE7 : FE E0		cpi	0E0h	; je znak s kodom >= 0E0h?
1585/8AE9 : D2 DE 8A		jnc	SetWrkK	; je, preskocit
1586/8AEC : 7E	SetWrkM:	mov	a,m	; prenes zbytok tabulky hore
1587/8AED : 12		stax	d	
1588/8AEE : 2B		dcx	h	
1589/8AEF : 1B		dcx	d	
1590/8AF0 : B7		ora	a	; koniec tabulky?
1591/8AF1 : C2 EC 8A		jnz	SetWrkM	; nie, pokračuj v presuvani
1592/8AF4 : 13		inx	d	; posun sa na ukoncovaciu 0
1593/8AF5 : EB		xchg		
1594/8AF6 : EB	SetWrkJ:	xchg		; DE=aktualny koniec tabulky (byte 0)
1595/8AF7 :				; Nasledujuci test overuje, ci kluce neprekracuju spodnu hranicu miesta urceneho
1596/8AF7 :				; pre programove kluce. Neberie vsak do uvahy aktualnu velkost vkladaneho textu.
1597/8AF7 :				; Takze adresa KEYTOP sa moze prekrocit aj o 80 bytov, co velkost editacneho
1598/8AF7 :				; riadku.
1599/8AF7 : 2A 7E C1	SetWrkH:	lhld	KEYTOP	; adresa hranice pamate pre kluce
1600/8AFA : 7B		mov	a,e	; porovanie s aktualnym koncom
1601/8AFB : 95		sub	l	
1602/8AFC : 7A		mov	a,d	
1603/8AFD : 9C		sbb	h	
1604/8AFE : DA 32 8B		jc	ERROR	; DE < HL => ohlas "MEMORY OVERFLOW"
1605/8B01 : 2A 7C C1		lhld	KEYEND	; adresa vrcholu tabulky klucov
1606/8B04 : EB		xchg		; do DE
1607/8B05 : 05		dcr	b	; bol prazdny riadok?
1608/8B06 : FA 14 8B		jm	SetWrkF	; ano, skoc - nebude sa vkladat
1609/8B09 : 04		inr	b	; vrat dlzku textu
1610/8B0A : 71		mov	m,c	; uloz kod kluca
1611/8B0B : 2B		dcx	h	; zniz ukazatel

1612/8B0C : 1A	SetWrkG:	ldax	d	; prenes text kluca
1613/8B0D : 77		mov	m,a	
1614/8B0E : 2B		dcx	h	; posun ukazatele
1615/8B0F : 13		inx	d	
1616/8B10 : 05		dcr	b	; opakuj pre cely text
1617/8B11 : C2 0C 8B		jnz	SetWrkG	
1618/8B14 : 36 00	SetWrkF:	mvi	m,0	; pridaj ukoncovaci 0
1619/8B16 : 21 35 C1	SetWrkO:	lxi	h,SWRK	; vypni priznak WRK
1620/8B19 : 36 00		mvi	m,0	
1621/8B1B : C9		ret		
1622/8B1C :				
1623/8B1C :				
1624/8B1C :				
1625/8B1C :				
1626/8B1C :				
1627/8B1C :				
1628/8B1C :				
1629/8B1C :				
1630/8B1C :				
1631/8B1C : 21 00 00	OCHRAN:	lxi	h,0	; bude sa nulovat pamat
1632/8B1F : E5	OCHRANL:	push	h	; vynuluj 2 byty
1633/8B20 : 3A F1 C1		lda	OCHR	; ma sa pamat vymazat?
1634/8B23 : FE 2A		cpi	"*"	
1635/8B25 :				
1636/8B25 : CA 1F 8B		jz	OCHRANL	; ak ano, pokracuj v mazani
1637/8B28 : 31 00 80		lxi	sp,Stack	; pamat je vymazana, nastav opat zasobnik
1638/8B2B : 11 31 82		lxi	d,INITAB	; DE=adresa inicializacnej tabulky
1639/8B2E : C3 0D 80		jmp	MONLL	; pokracuj v inicializacii
1640/8B31 :				
1641/8B31 :				
1642/8B31 :				
1643/8B31 :				
1644/8B31 : E5	PRBTXT:	push	h	
1645/8B32 : CD A3 88	ERROR:	call	BEEP	; pipni
1646/8B35 : E1		pop	h	
1647/8B36 : 2A 74 C0	PRMTXT:	lhld	MESS	; adresa textu spravy
1648/8B39 : EB	PRHTXT:	xchg		; do DE
1649/8B3A : 2A 3A C1		lhld	CDBUF	; HL=adresa buffra posledneho hlasenia
1650/8B3D : 06 30		mvi	b,48	; max. 48 znakov
1651/8B3F : EB		xchg		; prehod ukazatele
1652/8B40 : CD C5 81		call	TEXTSP	; skopiruj spravu do buff. posl. hlasenia
1653/8B43 : C3 9B 88		jmp	PRPTXT	; skoc vypisat samotne hlasenie
1654/8B46 :				
1655/8B46 :				
1656/8B46 :				
1657/8B46 :				
1658/8B46 :				

```

1659/8B46 : ; [LSTR]=dlzka textu v DR
1660/8B46 : ; M: HL, AF
1661/8B46 : 2A 34 C0 ENDLN:      lhld    DLZRAD      ; koncova adresa buffra DR do HL
1662/8B49 : 3E 20 EndLnS:      mvi     a,' '      ; vyhľadaj prvý nemedzerový znak
1663/8B4B : BE          cmp     m          ; je to medzera?
1664/8B4C : C2 53 8B          jnz     EndLnE      ; ak nie, skoc dalej - mame koniec
1665/8B4F : 2D          dcr     l          ; zniz adresu v DR
1666/8B50 : C2 49 8B          jnz     EndLnS      ; a hľadaj dalej
1667/8B53 : 7C EndLnE:      mov     a,h      ; odpamataj H do A
1668/8B54 : 26 00          mvi     h,0      ; HL=dlzka textu v DR
1669/8B56 : 22 3C C1          shld    LSTR      ; uloz dlzku textu v DR
1670/8B59 : 67          mov     h,a      ; obnov H
1671/8B5A : C9          ret
1672/8B5B :
1673/8B5B : ;-----
1674/8B5B : ; Pokracovanie editacie v riadku.
1675/8B5B : 22 7A C0 ENTER1:      shld    RETSP      ; uloz si povodnu adresu zasobnika
1676/8B5E : 2A 70 C0          lhld    SPVAL      ; nastav novu adresu zasobnika
1677/8B61 : F9          sphl
1678/8B62 : 2A 32 C0 ENTER2:      lhld    DIAL      ; adresa VRAM vypisu do DR
1679/8B65 : CD 61 81          call   CURPOSHL    ; vypocet adresy kurzora
1680/8B68 : 22 75 C1          shld    DCUROL      ; odloz si ju
1681/8B6B : CD 80 8A          call   XCCUR      ; zobraz kurzor
1682/8B6E : CD A1 84          call   INKLAV      ; cakaj na klaves
1683/8B71 : 2A 75 C1          lhld    DCUROL      ; adresa kurzora
1684/8B74 : CD 80 8A          call   XCCUR      ; zmaz kurzor
1685/8B77 : CD 00 88          call   KEYOUT     ; modifikuj DR
1686/8B7A : 3A 34 C1          lda     ASCII      ; posledne stlaceny klaves
1687/8B7D : FE 0D          cpi     CR          ; bol to EOL?
1688/8B7F : C2 62 8B          jnz     ENTER2     ; ak nie, pokracuj v editacii
1689/8B82 : ; bol stlaceny EOL
1690/8B82 : CD 13 81          call   CLR          ; zmaz obsah buffra DR
1691/8B85 : CD 55 88          call   WRBUFF      ; zmaz DR
1692/8B88 : 2A 7A C0          lhld    RETSP      ; obnov SP
1693/8B8B : F9          sphl
1694/8B8C : 2A 76 C0          lhld    RETCALL     ; a nepriamo sa vrat
1695/8B8F : E9          pchl
1696/8B90 :
1697/8B90 : ;-----
1698/8B90 : ; Pokracovanie obsluhy klavesu ENDL : Posun kurzora za posledn nemedzerov znak v DR
1699/8B90 : ; klaves ENDL (CNT + sipka vpravo)
1700/8B90 : CD 46 8B ENDL:      call   ENDLN      ; adresa posledneho znaku do HL
1701/8B93 : 3A 34 C0          lda     DLZRAD      ; nizsi byte koncovej adresy buffra DR
1702/8B96 : BD          cmp     l          ; sme na konci riadku?
1703/8B97 : CA 9B 8B          jz     EndlA      ; skoc, ak ano
1704/8B9A : 2C          inr     l          ; posun za posledny znak
1705/8B9B : 3A FE C0 EndlA:      lda     DLZR      ; max. pocet zobrazonych znakov v riadku

```

```

1706/8B9E : 3D          dcr      a          ; -1
1707/8B9F : BD          cmp      l          ; bude treba zmenit posunutie textu?
1708/8BA0 : 7D          mov      a,l        ; A=pozicia posledneho znaku (kurzora)
1709/8BA1 : 21 38 C0     lxi      h,RPOS    ; prvý vypisovaný znak
1710/8BA4 : D2 B0 8B     jnc      EndlB      ; skoc, ak netreba posuvat
1711/8BA7 : 4F          mov      c,a        ; C=pozicia posledneho znaku
1712/8BA8 : 3A FF C0     lda      DLZP      ; max. pocet vysunutych znakov v riadku
1713/8BAB : 47          mov      b,a        ; do B
1714/8BAC : 77          mov      m,a        ; a uloz do RPOS
1715/8BAD : 79          mov      a,c        ; vypocitaj poziciu kurzora
1716/8BAE : 90          sub      b          ;
1717/8BAF : 21          db      21h        ; lxi h,NN - preskoc nasl. instrukciu
1718/8BB0 : 36 00          mvi      m,0        ; vynuluj RPOS
1719/8BB2 : 32 36 C0     sta      DCUR      ; uloz poziciu kurzora
1720/8BB5 : C9          ret          ;
1721/8BB6 :
1722/8BB6 : ;-----
1723/8BB6 : ; Obľuha klavesu EOL : ukoncenie editacie v DR.
1724/8BB6 : CD 46 8B     EOL:      call    ENDLN      ; adresa konca textu do HL
1725/8BB9 : E5          push    h          ; odpamataj
1726/8BBA : 45          mov      b,l        ; pocet znakov do B
1727/8BBB : 2A 30 C0     lhld    BUF          ; adresa textu v DR
1728/8BBE : EB          xchg      ; do DE
1729/8BBF : 2A 78 C0     lhld    ODLOZ      ; adresa pre ulozenie zadaneho textu
1730/8BC2 : 3A 37 C1     lda      SPTL      ; priznak vypisu na obrazovku
1731/8BC5 : 4F          mov      c,a        ; do C
1732/8BC6 : 1A          EoLL:      ldax    d          ; vezmi znak z DR
1733/8BC7 : 77          mov      m,a        ; uloz do cieloveho buffra
1734/8BC8 : 23          inx      h          ; posun ukazatele
1735/8BC9 : 13          inx      d          ;
1736/8BCA : 0C          inr      c          ; vypisovat na obrazovku?
1737/8BCB : C4 00 85     cnz      PRTOUT    ; ao, vypis
1738/8BCE : 0D          dcr      c          ;
1739/8BCF : 05          dcr      b          ; opakuj pre cely text v DR
1740/8BD0 : F2 C6 8B     jp      EoLL      ;
1741/8BD3 : 36 0D          mvi      m,CR      ; a este odriadkuj
1742/8BD5 : 0C          inr      c          ; ak je to povolene
1743/8BD6 : C4 FE 84     cnz      PRTCR      ;
1744/8BD9 : E1          pop      h          ; obnov adresu konca textu do HL
1745/8BDA : 0E 87          mvi      c,XRCL      ; kod kluca RCL
1746/8BDC : 23          inx      h          ; posun ukazatel za posledny znak
1747/8BDD : C3 C8 8A     jmp      SetWrkI    ; a uloz text do tabulky klucov
1748/8BE0 :
1749/8BE0 : ;-----
1750/8BE0 : ; Pokracovanie obsluhy klavesu DELT : Vymazanie znaku v mieste kurzora
1751/8BE0 : ; klaves DELT (CNT + N)
1752/8BE0 : 1A          DeltM:      ldax    d          ; presun znak

```

```

1753/8BE1 : 77          mov     m,a
1754/8BE2 : 79          mov     a,c          ; posledny znak?
1755/8BE3 : BB          cmp     e
1756/8BE4 : 23          inx     h          ; posun ukazatele
1757/8BE5 : 13          inx     d
1758/8BE6 : C2 E0 8B        jnz     DeltM          ; nebol posledny, pokracuj
1759/8BE9 : 36 20          mvi     m,' '          ; vymaz posledny znak
1760/8BEB : C9          ret
1761/8BEC :
1762/8BEC : ;-----
1763/8BEC : 00 00          db      0,0
1764/8BEE :
1765/8BEE : ;-----
1766/8BEE : ; Editacia v riadku.
1767/8BEE : CD 55 88        ENTER:   call    WRBUFF          ; vypis editacny riadok
1768/8BF1 : E1             pop     h          ; vyber navratovu adresu
1769/8BF2 : 22 76 C0        shld    RETCALL          ; a uloz si ju na neskor
1770/8BF5 : 21 00 00        lxi     h,0          ; SP do HL
1771/8BF8 : 39             dad     sp
1772/8BF9 : C3 5B 8B        jmp     ENTER1
1773/8BFC :
1774/8BFC : ;-----
1775/8BFC : 00 00 00 00      db      0,0,0,0
1776/8C00 :
1777/8C00 : ;-----
1778/8C00 : ; Odskok do rutiny pre prenos obsahu ROM Modulu do pamate.
1779/8C00 : ; (8C00h)
1780/8C00 : C3 A4 8F        ROMIN:   jmp     TRANSF
1781/8C03 :
1782/8C03 : ;-----
1783/8C03 : ; Letmy test klavesnice.
1784/8C03 : ; I: -
1785/8C03 : ; O: NZ, A=kod stlaceneho klavesu;
1786/8C03 : ;      Z, A=0 - ak nebolo nic stlacene
1787/8C03 : ; M: AF
1788/8C03 : C5             SCAN:   push    b          ; odpamataj BC
1789/8C04 : 06 07          mvi     b,7          ; 7 riadkov klavesnice
1790/8C06 : 0E 7F          mvi     c,7Fh          ; pociatocna maska riadku
1791/8C08 : 79             ScanR:  mov     a,c          ; odrotuj masku
1792/8C09 : 07             rlc
1793/8C0A : 4F             mov     c,a
1794/8C0B : D3 F4          out     SYS55A          ; posli masku riadku na port
1795/8C0D : DB F5          in      SYS55B          ; precitaj stav riadku
1796/8C0F : 2F             cma          ; invertuj
1797/8C10 : B7             ora     a          ; bolo nieco stlacene?
1798/8C11 : C2 C3 84        jnz     ScanM          ; ak ano, skoc vyhodnotit klaves
1799/8C14 : 05             dcr     b          ; opakuj pre vsetky riadky

```

```

1800/8C15 : F2 08 8C      jp      ScanR
1801/8C18 : C1            pop     b          ; obnov BC
1802/8C19 : AF            xra     a          ; A=0 - nic nebolo stlacene
1803/8C1A : C9            ret
1804/8C1B :
1805/8C1B : ;-----
1806/8C1B : ; Zdrzanie podla hodnoty v A.
1807/8C1B : ; I: A=velkost zdrzania.
1808/8C1B : ; O: -
1809/8C1B : ; M: AF
1810/8C1B : ; T: ~(A-1)*5.5ms
1811/8C1B : C5      DelayA:      push    b
1812/8C1C : 47              mov     b,a
1813/8C1D : 0B      DelayB:      dcx     b
1814/8C1E : 79              mov     a,c
1815/8C1F : B0              ora     b
1816/8C20 : C2 1D 8C      jnz     DelayB
1817/8C23 : C1              pop     b
1818/8C24 : C9            ret
1819/8C25 :
1820/8C25 : ;-----
1821/8C25 : ; Pokracovanie testu klavesnice - rutiny INKLAV.
1822/8C25 : ; I: A=kod klavesu
1823/8C25 : ; O: A=kod klavesu
1824/8C25 : ; M: AF
1825/8C25 : 36 06      InKlavS:      mvi     m,6          ; nastav mensie zdrzanie pri opakovaní
1826/8C27 : F5      InKlavN:      push    psw          ; odpamataj kod klavesu
1827/8C28 : 3A BF C0      lda     SBEEP          ; je povolené pipnutie
1828/8C2B : B7              ora     a
1829/8C2C : CA 35 8C      jz      InKlavB          ; ak nie, skoc dalej
1830/8C2F : 21 F4 80      lxi     h,M_Key          ; adresa dat pipnutia
1831/8C32 : CD A6 88      call   BELL             ; pipni
1832/8C35 : F1      InKlavB:      pop     psw          ; obnov kod klavesu
1833/8C36 : C3 F4 84      jmp     ScanE2          ; skoc ukoncit rutinu
1834/8C39 :
1835/8C39 : ;-----
1836/8C39 : ; Dvojnásobne dlhe čakanie (2*[SPEED]) pred načítaním bitu z MGF.
1837/8C39 : ; I: -
1838/8C39 : ; O: CY=úroveň precítaného bitu
1839/8C39 : ; M: AF
1840/8C39 : 3A 7C C0      WaitBitD:      lda     SPEED          ; dĺžka pulzu do A
1841/8C3C : 17              ral              ; *2
1842/8C3D : C3 62 8D      jmp     WaitBitA          ; skoc do rutiny čakania
1843/8C40 :
1844/8C40 : ;-----
1845/8C40 : ; Standardný vstupný bod návratu do Monitora.
1846/8C40 : ; (8C40h)

```



```

1847/8C40 : C3 3B 80
1848/8C43 :
1849/8C43 :
1850/8C43 :
1851/8C43 : 3A 34 C1
1852/8C46 : 47
1853/8C47 : FE E0
1854/8C49 : D2 94 81
1855/8C4C : FE D0
1856/8C4E : D2 27 88
1857/8C51 : FE 20
1858/8C53 : CD 16 8B
1859/8C56 : D2 94 81
1860/8C59 :
1861/8C59 : 2A 32 C1
1862/8C5C : 23
1863/8C5D : 23
1864/8C5E : 7E
1865/8C5F : A7
1866/8C60 : C8
1867/8C61 : 23
1868/8C62 : B8
1869/8C63 : C2 5C 8C
1870/8C66 : 46
1871/8C67 : 23
1872/8C68 : 66
1873/8C69 : 68
1874/8C6A : E9
1875/8C6B :
1876/8C6B :
1877/8C6B :
1878/8C6B : 3E 8A
1879/8C6D : D3 F7
1880/8C6F : C3 A7 85
1881/8C72 :
1882/8C72 :
1883/8C72 : 00 00
1884/8C74 :
1885/8C74 :
1886/8C74 :
1887/8C74 :
1888/8C74 :
1889/8C74 :
1890/8C74 :
1891/8C74 : DB F6
1892/8C76 : E6 10
1893/8C78 : C0

```

```

Monit:          jmp      MON1

;-----
; Prevedenie zmien v DR.
DOKEY:          lda      ASCII          ; kod stlaceneho klavesu
                mov      b,a           ; do B
                cpi      0E0h          ; kody >= 0E0h
                jnc      CHAR          ; sa povazuju za tlacitelne
                cpi      KK0           ; kody <0D0h, 0DBh>
                jnc      KEY           ; su programove kluce
                cpi      20h           ; riadiaci kod?
                call     SetWrk0        ; zrus priznak WRK
                jnc      CHAR          ; je to znak, vypisat
                ; B=kod riadiaceho znaku
                lhld     KDIR           ; adresa tabulky obsluhy riadich kodov-2
DoKeyS:         inx      h             ; preskoc adresu
                inx      h
                mov      a,m           ; prevezmi kod
                ana      a             ; je to koniec tabulky?
                rz        ; ano, navrat
                inx      h             ; preskoc kod v tabulke
                cmp      b             ; je to hladany kod?
                jnz      DoKeyS        ; nie, hladaj dalej
JmpIndHL:       mov      b,m           ; je to on, vyber adresu obsluhy dp HL
                inx      h
                mov      h,m
                mov      l,b
                pchl                    ; a skoc do rutiny obsluhy

;-----
; Inicializuj systemovy PIO a zmaz obrazovku.
INIARA:         mvi      a,SYSCTRL     ; nastav: PA:OUT, PB:IN, PCH:IN, PCL:OUT
                out      SYS55M        ; a pripoj RAM od adresy 0
                jmp      ERASE         ; skoc na zmazanie obrazovky

;-----
                db        0,0

;-----
; Test klavesu STOP.
; I: -
; O: Z, A=3 - klaves STOP stlaceny
;     NZ, A=40h - klaves STOP nestlaceny
; M: AF
STOP:           in       SYS55C        ; stav portu C
                ani      X_STOP        ; odmaskuj bit klavesu STOP
                rnz                    ; STOP nie je stlaceny, navrat

```

```

1894/8C79 : 3E 03          mvi    a,3          ; STOP je stlaceny
1895/8C7B : C9            ret
1896/8C7C :
1897/8C7C : ;-----
1898/8C7C : 00          db      0
1899/8C7D :
1900/8C7D : ;-----
1901/8C7D : ; Vykreslenie bodu v danom grafickom mode.
1902/8C7D : ; I: [X1], [Y1] = suradnice bodu
1903/8C7D : ; O: -
1904/8C7D : ; M: AF
1905/8C7D : ; (8C7Dh)
1906/8C7D : C5          POINT:    push    b          ; odpamataj registre
1907/8C7E : D5            push    d
1908/8C7F : E5            push    h
1909/8C80 : 21 75 85        lxi     h,ADRRET      ; navratova adresa, ktora obnovi
1910/8C83 : E5            push    h          ; registre, na zasobnik
1911/8C84 :
1912/8C84 : ;-----
1913/8C84 : ; Vykreslenie bodu v danom grafickom mode.
1914/8C84 : ; I: [X1], [Y1] = suradnice bodu
1915/8C84 : ; O: E=suradnica X, D=suradnica Y
1916/8C84 : ; M: vsetky
1917/8C84 : 2A 71 C1        PointZ:    lhld    Y1-1          ; suradnica Y do H
1918/8C87 : 3A 70 C1        lda      X1          ; suradnica X do L
1919/8C8A : 6F            mov     l,a
1920/8C8B : E5            push    h          ; odpamataj suradnice
1921/8C8C : CD 94 8C        call    POSPOINT      ; vypocitaj adresu VRAM a masku bodu
1922/8C8F : CD F4 C1        call    VYSTUP        ; vykresli bod
1923/8C92 : D1            pop     d          ; obnov suradnice do DE
1924/8C93 : C9            ret
1925/8C94 :
1926/8C94 : ;-----
1927/8C94 : ; Vypocet adresy a masky bodu vo VRAM.
1928/8C94 : ; Pracuje sa iba s rozlisenim 256x256 bodov.
1929/8C94 : ; Najlavejsi 0. pixel zacina v 2. znakovom stlpci.
1930/8C94 : ; I: L=suradnica X, H=suradnica Y
1931/8C94 : ; O: HL=adresa VRAM bodu, B=maska bodu
1932/8C94 : ; M: vsetky
1933/8C94 : E5          POSPOINT:    push    h          ; odpamataj suuradnice
1934/8C95 : 3A 3A C0        lda      COLOR        ; atribut do A
1935/8C98 : E6 C0          ani      0C0h          ; ponechaj iba farbove bity
1936/8C9A : 32 F8 C1        sta      PLOTCOL      ; a uloz do rutiny vykreslenia bodu
1937/8C9D : 26 00          mvi     h,0          ; H=0
1938/8C9F : 4D            mov     c,l          ; C=X
1939/8CA0 : 44            mov     b,h          ; B=0
1940/8CA1 : 23            inx     h          ; X+1

```

```

1941/8CA2 : 54      mov     d,h
1942/8CA3 : 5D      mov     e,l          ; DE=1x
1943/8CA4 : 29      dad     h          ; *2
1944/8CA5 : 29      dad     h          ; *4
1945/8CA6 : 19      dad     d          ; *5
1946/8CA7 : 29      dad     h          ; *10
1947/8CA8 : 29      dad     h          ; *20
1948/8CA9 : 19      dad     d          ; *21
1949/8CAA : 29      dad     h          ; *42
1950/8CAB : 29      dad     h          ; *84
1951/8CAC : 19      dad     d          ; *85 <=> /3
1952/8CAD : 3E FE    mvi     a,0FEh
1953/8CAF : A4      ana     h
1954/8CB0 : 0F      rrc          ; /2
1955/8CB1 : 5F      mov     e,a          ; E=X DIV 6 - znakovy stlpec
1956/8CB2 : 7B      mov     a,e          ; (zbytocne)
1957/8CB3 : 87      add     a          ; *2
1958/8CB4 : 83      add     e          ; *3
1959/8CB5 : 87      add     a          ; *6
1960/8CB6 : 91      sub     c          ; -(X mod 6)
1961/8CB7 : 2F      cma          ; zneguj
1962/8CB8 : 3C      inr     a
1963/8CB9 : 4F      mov     c,a          ; C = X MOD 6 - index do tabulky masiek
1964/8CBA : 16 C0    mvi     d,(VRAM >> 8) ; vyssi byte VRAM do D
1965/8CBC : 1C      inr     e
1966/8CBD : 1C      inr     e          ; 0. pixel zacina 2. znakovom stlpci
1967/8CBE : 21 D7 80 lxi     h,BitMasks ; tabulka bitovych masiek
1968/8CC1 : 09      dad     b          ; pripocitaj index do tabulky
1969/8CC2 : 46      mov     b,m          ; B=maska bodu
1970/8CC3 : E1      pop     h          ; obnov suradnice
1971/8CC4 : 6C      mov     l,h
1972/8CC5 : 26 00    mvi     h,0          ; HL=Y
1973/8CC7 : 29      dad     h          ; *2
1974/8CC8 : 29      dad     h          ; *4
1975/8CC9 : 29      dad     h          ; *8
1976/8CCA : 29      dad     h          ; *16
1977/8CCB : 29      dad     h          ; *32
1978/8CCC : 29      dad     h          ; *64
1979/8CCD : 19      dad     d          ; + VRAM + X DIV 6 + 2
1980/8CCE : C9      ret
1981/8CCF :
1982/8CCF :          ;-----
1983/8CCF : 00      db     0
1984/8CD0 :
1985/8CD0 :          ;-----
1986/8CD0 :          ; Linearny interpolator.
1987/8CD0 :          ; I: [X1], [Y1]=suradnice pociatocneho bodu

```

```

1988/8CD0 : ; [X2], [Y2]=suradnice koncového bodu
1989/8CD0 : ; O: [X1]=[X2], [Y1]=[Y2]
1990/8CD0 : ; M: všetky
1991/8CD0 : ; (8CD0h)
1992/8CD0 : CD 84 8C INPOL: call PointZ ; vykresli pociatocny bod
1993/8CD3 : ; DE=suradnice pociatocného bodu
1994/8CD3 : 2A 73 C1 lhld X2 ; suradnice koncového bodu
1995/8CD6 : 7D mov a,l ; zapisat do X1 a Y1
1996/8CD7 : 32 70 C1 sta X1
1997/8CDA : 7C mov a,h
1998/8CDB : 32 72 C1 sta Y1
1999/8CDE : ; usecka sa kresli vzdy zprava do lava
2000/8CDE : 7D mov a,l ; porovnaj X-ove suradnice
2001/8CDF : 93 sub e ; X2-X1
2002/8CE0 : D2 E6 8C jnc InpolDX ; X2 >= X1
2003/8CE3 : 7B mov a,e
2004/8CE4 : 95 sub l
2005/8CE5 : EB xchg ; prehodenie suradnic
2006/8CE6 : 47 InpolDX: mov b,a ; B=DX
2007/8CE7 : ; podľa zmeru hore/dole sa pripraví offset
2008/8CE7 : 7C mov a,h ; porovnaj Y-ove suradnice
2009/8CE8 : 92 sub d
2010/8CE9 : 11 C0 FF lxi d,-64 ; offset uR - zdola hore
2011/8CEC : D2 F4 8C jnc InpolDY
2012/8CEF : 11 40 00 lxi d,64 ; offset uR - zhora dole
2013/8CF2 : 2F cma ; neguj
2014/8CF3 : 3C inr a
2015/8CF4 : 4F InpolDY: mov c,a ; C=DY
2016/8CF5 :
2017/8CF5 : EB xchg ; uloz si uR offset
2018/8CF6 : 22 3E C1 shld STACK
2019/8CF9 : EB xchg
2020/8CFA :
2021/8CFA : 50 mov d,b ; DX
2022/8CFB : 5F mov e,a ; DY
2023/8CFC : 1C inr e ; DY+1
2024/8CFD : D5 push d
2025/8CFE : C5 push b
2026/8CFF : CD 94 8C call POSPOINT ; HL=adresa VRAM aktualného bodu
2027/8D02 : 50 mov d,b ; D=bitova maska
2028/8D03 : C1 pop b
2029/8D04 : 58 mov e,b ; DX
2030/8D05 : 42 mov b,d ; bitova maska
2031/8D06 : 7B mov a,e ; DX-DY
2032/8D07 : B9 cmp c
2033/8D08 : D2 15 8D jnc InpolK
2034/8D0B :

```

```

2035/8D0B :                ; DY > DX
2036/8D0B :
2037/8D0B : B7             ora     a                ; CY=0
2038/8D0C : 1F             rar     ; DX/2
2039/8D0D : 91             sub     c                ; DX/2-DY
2040/8D0E : 5F             mov     e,a             ; ??????
2041/8D0F : 1D             dcr     e                ; DX/2-DY-1
2042/8D10 : 7B             mov     a,e             ; ??????
2043/8D11 : D1             pop     d
2044/8D12 : C3 42 8D       jmp     InpolP
2045/8D15 :
2046/8D15 :                ; DY >= DY
2047/8D15 :
2048/8D15 : 79             InpolK: mov     a,c
2049/8D16 : 1F             rar     ; DY/2
2050/8D17 : 93             sub     e                ; DY/2-DX
2051/8D18 : 2F             cma     ; NOT
2052/8D19 : 5F             mov     e,a
2053/8D1A : 1C             inr     e                ; NEG : DX-DY/2
2054/8D1B : 06 00         mvi     b,0
2055/8D1D : 78             InpolL: mov     a,b
2056/8D1E : 82             add     d
2057/8D1F : 47             mov     b,a
2058/8D20 : 7A             InpolM: mov     a,d
2059/8D21 : 0F             rrc
2060/8D22 : 57             mov     d,a
2061/8D23 : D2 2E 8D       jnc     InpolN
2062/8D26 : CD F4 C1       call    VYSTUP                ; vykresli bod
2063/8D29 : 2B             dcx     h
2064/8D2A : 16 20         mvi     d,20h
2065/8D2C : 06 00         mvi     b,0
2066/8D2E : 37             InpolN: stc
2067/8D2F : 7B             mov     a,e
2068/8D30 : 99             sbb     c
2069/8D31 : 5F             mov     e,a
2070/8D32 : D2 1D 8D       jnc     InpolL
2071/8D35 : CD F4 C1       InpolO: call    VYSTUP                ; vykresli bod
2072/8D38 : 42             mov     b,d
2073/8D39 : 7B             mov     a,e
2074/8D3A : EB             xchg
2075/8D3B : 2A 3E C1       lhld    STACK                ; uR offset
2076/8D3E : 19             dad     d                ; posun sa hore/dole
2077/8D3F : D1             pop     d                ; D=DY, E=DY+1
2078/8D40 : 1D             dcr     e                ; zniz pocitadlo bodov
2079/8D41 : C8             rz                ; hotovo, navrat
2080/8D42 :
2081/8D42 : D5             InpolP: push    d

```

```

2082/8D43 : 37          stc
2083/8D44 : 8A          adc     d
2084/8D45 : 5F          mov     e,a
2085/8D46 : 50          mov     d,b
2086/8D47 : D2 35 8D        jnc     Inpol0
2087/8D4A : C3 20 8D        jmp     InpolM
2088/8D4D :
2089/8D4D : ;-----
2090/8D4D : ; Obsluha klavesu DELT : Vymazanie znaku v mieste kurzora
2091/8D4D : ; klaves DELT (CNT + N)
2092/8D4D : CD 46 8B        DEL:      call    ENDLN          ; adresa posledneho znaku do HL
2093/8D50 : 4D          mov     c,l          ; pozicia posledneho znaku do C
2094/8D51 : CD 5E 81        call    CURPOS          ; adresa znaku v mieste kurzora
2095/8D54 : 54          mov     d,h          ; aj do DE
2096/8D55 : 5D          mov     e,l
2097/8D56 : 13          inc     d          ; posun na nasledujuci znak
2098/8D57 : 79          mov     a,c
2099/8D58 : BB          cmp     e          ; posledny znak?
2100/8D59 : DA E9 8B        jc      DeltS          ; skoc, ak ano
2101/8D5C : C3 E0 8B        jmp     DeltM          ; skoc presunut text vpravo od kurzora
2102/8D5F :
2103/8D5F : ;-----
2104/8D5F : ; Cakanie pred nacitanim bitu z MGF.
2105/8D5F : ; I: [SPEED]=dlzka jedneho pulzu
2106/8D5F : ; O: CY=uroven precitaneho bitu
2107/8D5F : ; M: AF
2108/8D5F : 3A 7C C0        WaitBitS:  lda     SPEED          ; konstanta dlzky pulzu
2109/8D62 : 3D          WaitBitA:  dcr     a          ; zniz dlzku cakania
2110/8D63 : C2 62 8D        jnz     WaitBitA        ; opakuje, kym sa nevynuluje
2111/8D66 : DB F6          in      SYS55C        ; precitaj stav portu C
2112/8D68 : 17          ral          ; bit X_MGF_IN do CY
2113/8D69 : C9          ret
2114/8D6A :
2115/8D6A : ;-----
2116/8D6A : 00 00          db      0,0
2117/8D6C :
2118/8D6C : ;-----
2119/8D6C : ; Zapis bloku dat na MGF.
2120/8D6C : ; I: HL=adresa zaciatku bloku dat, DE=dlzka bloku-1
2121/8D6C : ; O: -
2122/8D6C : ; M: vsetky
2123/8D6C : 13          TAPEOUT:  inc     d          ; dlzka +1
2124/8D6D : 06 00        mvi     b,0          ; vynuluj CRC
2125/8D6F : 7E          TapeOutL:  mov     a,m          ; vezmi byte
2126/8D70 : CD 7E 8D        call    WriteByte     ; a zapis na MGF
2127/8D73 : 7E          mov     a,m          ; opat vezmi byte
2128/8D74 : 80          add     b          ; modifikuj CRC

```

```

2129/8D75 : 47          mov     b,a           ; a uloz
2130/8D76 : 23          inx     h           ; posun sa na dalsi byte
2131/8D77 : 1B          dcx     d           ; zniz pocitadlo bytov
2132/8D78 : 7A          mov     a,d           ; opakuj zapis pre cwly blok
2133/8D79 : B3          ora     e
2134/8D7A : C2 6F 8D      jnz     TapeOutL
2135/8D7D : 78          mov     a,b           ; posli este vypocitane CRC
2136/8D7E :                ; pokračuj v WriteByte
2137/8D7E :
2138/8D7E :                ;-----
2139/8D7E :                ; Zapis bytu na MGF.
2140/8D7E :                ; Zapisu jednotlivych bitov bytu predchadza zapis start bitu Log.1.
2141/8D7E :                ; Bity sa zapisuju v poradi 0 az 7.
2142/8D7E :                ; I: A=zapisovany byte
2143/8D7E :                ; O: -
2144/8D7E :                ; M: AF
2145/8D7E : C5          WriteByte: push    b           ; odpamataj BC
2146/8D7F : 4F          mov     c,a           ; zapisovany byte do C
2147/8D80 : CD A2 8D      call   WriteBit1        ; zapis start bit
2148/8D83 : 06 08          mvi     b,8           ; 8 bitov bytu
2149/8D85 : 79          WriteByteL: mov     a,c           ; zapisovany byte
2150/8D86 : 1F          rar           ; odrotuj bit do CY
2151/8D87 : 4F          mov     c,a           ; uloz si modifikovanu hodnotu
2152/8D88 : DC A2 8D      cc      WriteBit1        ; pre CY=1 - vytvor Log.1
2153/8D8B : D4 94 8D      cnc     WriteBit0        ; pre CY=0 - vytvor Log.0
2154/8D8E : 05          dcr     b           ; opakuj 8x
2155/8D8F : C2 85 8D      jnz     WriteByteL
2156/8D92 : C1          pop     b           ; obnov BC
2157/8D93 : C9          ret
2158/8D94 :
2159/8D94 :                ;-----
2160/8D94 :                ; Zapis Log.0 na MGF.
2161/8D94 :                ; 1T 2T
2162/8D94 :                ;
2163/8D94 :                ; |_____|
2164/8D94 :                ; I: [SPEED]=dlzka jedneho pulzu (T)
2165/8D94 :                ; O: -
2166/8D94 :                ; M: AF
2167/8D94 : 3E 01          WriteBit0: mvi     a,X_MGF_OUT      ; nastav H na MGF vystup
2168/8D96 : D3 F6          out     SYS55C
2169/8D98 : CD 5F 8D      call   WaitBitS        ; vytvor zdrzanie 1T
2170/8D9B : AF          xra     a           ; nastav L na MGF vystup
2171/8D9C : D3 F6          out     SYS55C
2172/8D9E : CD 39 8C      call   WaitBitD        ; vytvor zdrzanie 2T
2173/8DA1 : C9          ret
2174/8DA2 :
2175/8DA2 :                ;-----

```

```

2176/8DA2 : ; Zapis Log.1 na MGF.
2177/8DA2 : ; 3T 2T
2178/8DA2 : ;
2179/8DA2 : ; |_____|_____|
2180/8DA2 : ; I: [SPEED]=dlzka jedneho pulzu (T)
2181/8DA2 : ; O: CY=1
2182/8DA2 : ; M: AF
2183/8DA2 : 3E 01 WriteBit1: mvi a,X_MGF_OUT ; nastav H na MGF vystup
2184/8DA4 : D3 F6 out SYS55C
2185/8DA6 : CD 5F 8D call WaitBitS ; vytvor zdrzanie 1T
2186/8DA9 : CD 39 8C call WaitBitD ; vytvor zdrzanie 2T
2187/8DAC : AF xra a ; nastav L na MGF vystup
2188/8DAD : D3 F6 out SYS55C
2189/8DAF : CD 39 8C call WaitBitD ; vytvor zdrzanie 2T
2190/8DB2 : 37 stc ; CY=1
2191/8DB3 : C9 ret
2192/8DB4 :
2193/8DB4 : ;-----
2194/8DB4 : ; Text spravy po ukonceni prikazu MGSV.
2195/8DB4 : (MACRO) T_MgfStop: dm "++ MG STOP ++"
2195/8DB4 : 2B 2B 20 4D 47 20 53 54 db "++ MG STOP ++"
8DBC : 4F 50 20 2B 2B
2195/8DC1 : 0D db CR
2196/8DC2 :
2197/8DC2 : ;-----
2198/8DC2 : ; Nacitanie bloku dat z MGF.
2199/8DC2 : ; I: HL=cielova adresa, DE=dlzka-1
2200/8DC2 : ; O: Z=1 - blok nacistany v poriadku
2201/8DC2 : ; Z=0 - blok bol nacistany s chybou, alebo bol stlaceny klaves STOP
2202/8DC2 : ; M: AF, BC, DE
2203/8DC2 : 0E FF LoadBlock: mvi c,0FFh ; priznak nahratia bloku do pamate
2204/8DC4 : ; pokracuj v LoadBlockC
2205/8DC4 :
2206/8DC4 : ;-----
2207/8DC4 : ; Nacitanie bloku dat z MGF.
2208/8DC4 : ; I: HL=cielova adresa, DE=dlzka-1
2209/8DC4 : ; C=0 - iba kontrola zaznamu, C<>0 - nacistanie do pamate
2210/8DC4 : ; O: Z=1 - blok nacistany v poriadku
2211/8DC4 : ; Z=0 - blok bol nacistany s chybou, alebo bol stlaceny klaves STOP
2212/8DC4 : ; M: AF, BC, DE
2213/8DC4 : E5 LoadBlockC: push h ; odpamataj ukaldaciu adresu
2214/8DC5 : 06 00 mvi b,0 ; vynuluj CRC
2215/8DC7 : CD 49 8F LoadBlockL: call ReadBit ; nacistaj "start bit"
2216/8DCA : CD FA 8E call LoadByte ; nacistaj byte
2217/8DCD : C2 77 85 jnz ADDR1T ; ukonci rutinu, ak bol stlaceny STOP
2218/8DD0 : 0C inr c ; robi sa iba kontrola zaznamu?
2219/8DD1 : 0D dcr c

```



2220/8DD2 : CA D6 8D	jz	LoadBlockS	; ak ano, preskoc ulozenie
2221/8DD5 : 77	mov	m,a	; inak, uloz nacistany byte
2222/8DD6 : 23	LoadBlockS:	inx	; posun cielovu adresu
2223/8DD7 : 80	add	b	; modifikuj CRC
2224/8DD8 : 47	mov	b,a	
2225/8DD9 : 7A	mov	a,d	; uz sa precital cely blok?
2226/8DDA : B3	ora	e	
2227/8ddb : 1B	dcx	d	; zniz pocitadlo citanych bytov
2228/8DDC : C2 C7 8D	jnz	LoadBlockL	; pokracuj v citani
2229/8DDF : C3 81 8F	jmp	LoadBlockF	; skoc nacistat CRC
2230/8DE2 :			
2231/8DE2 :			
2232/8DE2 :			;-----
2233/8DE2 :			; Vyhľadanie a nactanie suboru.
2234/8DE2 :			; Do dialogoveho riadku vypise nactanu hlavicku v tvare 00/T FILENAME
2235/8DE2 :			; I: [FINDTP]=pozadovany typ suboru, 0 pre akymkoľvek typ
2236/8DE2 :			; [FINDNR]=pozadovane cislo suboru, 0 pre akekoľvek cislo
2237/8DE2 :			; O: Z=1 - subor nactany v poriadku
2238/8DE2 :			; Z=0 - subor bol nactany s chybou, alebo bol stlaceny klaves STOP
2239/8DE2 : CD 76 8A	SHEAD:	call	MgLDMsg ; priprav spravu "FILE ERROR"
2240/8DE5 : CD BE 8E		call	ReadSynch ; nactaj synchronizaciu
2241/8DE8 : C0		rnz	; navrat, ak bol stlaceny klaves STOP
2242/8DE9 : CD E4 8E		call	LoadHead ; nactaj hlavicku
2243/8DEC : C0		rnz	; navrat, ak bol stlaceny klaves STOP
2244/8DED : 11 B2 C1		lxi	d,HEAD ; DE=adresa nactanej hlavicky
2245/8DF0 : 1A		ldax	d ; cislo suboru do A
2246/8DF1 : CD 73 8E		call	BINBCD ; preved na BCD
2247/8DF4 : 2A 30 C0		lhld	BUF ; HL=buffer DR
2248/8DF7 : CD 3B 81		call	PREV02 ; zapis cislo do buffra DR
2249/8DFA : 3E 00		mvi	a,0 ; vypisovat sa bude od zaciatku riadku
2250/8DFC : 32 36 C0		sta	DCUR
2251/8DFF : 23		inx	h ; posun ukazatel v buffri DR
2252/8E00 : 36 2F		mvi	m,'/' ; pridaj '/' do buffra DR
2253/8E02 : 23		inx	h ; posun ukazatele
2254/8E03 : 13		inx	d
2255/8E04 : 1A		ldax	d ; typ suboru do A
2256/8E05 : 77		mov	m,a ; zapis do buffra DR
2257/8E06 : 23		inx	h
2258/8E07 : 36 20		mvi	m,' ' ; pridaj oddelovaciu medzeru
2259/8E09 : 23		inx	h
2260/8E0A : 11 B8 C1		lxi	d,NAMFIL ; adresa mena suboru v hlavicke suboru
2261/8E0D : EB		xchg	; zamen ukazatele
2262/8E0E : CD C3 81		call	TEXTTH ; a presun meno suboru z hlav. do buffra
2263/8E11 : CD 55 88		call	WRBUFF ; vypis pripraveny text do DR
2264/8E14 : 2A B2 C1		lhld	NUMFIL ; L=cislo suboru, H=typ suboru
2265/8E17 : 3A B0 C1		lda	FINDNR ; pozadovane cislo
2266/8E1A : A7		ana	a ; akekoľvek?

```

2267/8E1B : CA 22 8E          jz      SHeadN      ; skoc, ak ano
2268/8E1E : BD                cmp      1            ; zhoduje sa s cislom v hlavicke?
2269/8E1F : C2 8A 8F          jnz      SHeadB      ; nie, pipni a cakaj na dalsiu hlavicku
2270/8E22 : 3A B1 C1          SHeadN: lda      FINDTP      ; pozadovany typ
2271/8E25 : A7                ana      a            ; akykolvek?
2272/8E26 : 4F                mov      c,a          ; (uloz si typ do C pre LoadBlockC)
2273/8E27 : CA 2E 8E          jz      SHeadT      ; skoc, ak ano
2274/8E2A : BC                cmp      h            ; zhoduje sa s typom v hlavicke?
2275/8E2B : C2 8A 8F          jnz      SHeadB      ; nie, pipni a cakaj na dalsiu hlavicku
2276/8E2E : CD CB 8E          SHeadT: call     ReadSync    ; nacistaj synchro pred telom suboru
2277/8E31 : C0                rnz      ; navrat, ak bol stlaceny klaves STOP
2278/8E32 : CD ED 8E          call     LoadBody    ; nacistaj telo suboru da pamati
2279/8E35 : C0                rnz      ; navrat, ak bol stlaceny klaves STOP
2280/8E36 : 21 93 8F          lxi      h,T_MgfReadOK ; priprav spravu "READ OK"
2281/8E39 : 22 74 C0          shld     MESS
2282/8E3C : C9                ret
2283/8E3D :
2284/8E3D : ;-----
2285/8E3D : 00                db      0
2286/8E3E :
2287/8E3E : ;-----
2288/8E3E : ; Nacitanie zadaneho cisla suboru <00,62>.
2289/8E3E : ; Inicializuje vektor RAM na prosty navrat.
2290/8E3E : ; I: [CURCH]=adresa vstupu
2291/8E3E : ; O: CY=0, A=cislo suboru
2292/8E3E : ; CY=1 - chybne cislo suboru
2293/8E3E : ; M: vsetky
2294/8E3E : 21 91 82          DECPAR: lxi      h,T_ErrData    ; priprav text "ERROR IN DATA"
2295/8E41 : 22 74 C0          shld     MESS
2296/8E44 : 2A 72 C0          lhld     CURCH        ; adresa textu prikazu
2297/8E47 : CD F7 80          call     PAIRIN       ; precitaj sestnastkove cislo
2298/8E4A : D8                rc            ; navrat pri chybe
2299/8E4B : 23                inx      h            ; preskoc znak za cislom
2300/8E4C : 22 72 C0          shld     CURCH        ; uloz novu adresu
2301/8E4F : FE 63             cpi      63h          ; maximalne cislo je 62h
2302/8E51 : 3F                cmc
2303/8E52 : D8                rc            ; navrat s chybou, ak je cislo mimo
2304/8E53 : 4F                mov      c,a          ; uloz do C
2305/8E54 : 0F                rrc
2306/8E55 : 0F                rrc
2307/8E56 : 0F                rrc
2308/8E57 : 0F                rrc
2309/8E58 : E6 0F            ani      0Fh          ; odmaskuj
2310/8E5A : 87                add      a            ; x2
2311/8E5B : 47                mov      b,a
2312/8E5C : 87                add      a            ; x4
2313/8E5D : 87                add      a            ; x8

```

```

2314/8E5E : 80          add    b          ; x10
2315/8E5F : 47          mov    b,a        ; uloz do B
2316/8E60 : 79          mov    a,c        ; povodna hodnota
2317/8E61 : E6 0F      ani    0Fh        ; nizzsie 4 bity
2318/8E63 : 80          add    b          ; pripocitaj
2319/8E64 : E5          push   h
2320/8E65 : 21 00 C3    lxi    h,0C3h*256    ; JMP DECP1
2321/8E68 : 22 F0 C0    shld   RAM
2322/8E6B : 21 72 8E    lxi    h,DECP1
2323/8E6E : 22 F2 C0    shld   RAM+2
2324/8E71 : E1          pop     h
2325/8E72 : C9          DECP1: ret
2326/8E73 :
2327/8E73 : ;-----
2328/8E73 : ; Konverzia binarneho cisla na BCD.
2329/8E73 : ; I: A=binarne cislo 00h - 63h
2330/8E73 : ; O: A=BCD cislo 00 - 99.
2331/8E73 : ; M: HL, AF, B
2332/8E73 : 67          BINBCD: mov    h,a        ; prevadzana hodnota do H
2333/8E74 : 06 08      mvi    b,8        ; 8 bitove cislo
2334/8E76 : AF          xra    a        ; vynuluj vysledok
2335/8E77 : 29          BINBCDL: dad    h        ; odrotuj bit do CY
2336/8E78 : 8F          adc    a        ; zdvojnasoob vysledok a pripocitaj CY
2337/8E79 : 27          daa    a        ; preved na BCD
2338/8E7A : 05          dcr    b        ; opakuj 8x
2339/8E7B : C2 77 8E    jnz    BINBCDL
2340/8E7E : C9          ret
2341/8E7F :
2342/8E7F : ;-----
2343/8E7F : ; Obsluha prikazu MGSV : ulozenie suboru na MGF.
2344/8E7F : ; Format: MGSV nn ssss eeee filename
2345/8E7F : CD 3E 8E      MGSV:  call   DECPAR    ; nacistaj zadane cislo suboru
2346/8E82 : DA 49 80      jc     PRBWTXT    ; ak bolo chybne, vypis "ERROR IN DATA"
2347/8E85 : 21 46 80      lxi    h,SYST3    ; prirpav navratovu adresu do hlavnej
2348/8E88 : E5          push   h        ; slucky Monitora
2349/8E89 : 6F          mov    l,a        ; cislo suboru do L
2350/8E8A : 3A 36 C1    lda    MGCHAR    ; typ suboru
2351/8E8D : 67          mov    h,a        ; do H
2352/8E8E : 22 B2 C1    shld   NUMFIL    ; a uloz do hlavicky
2353/8E91 : CD 4B 81      call   RADR        ; precitaj zaciatoctu adresu
2354/8E94 : EB          xchg     ; do HL
2355/8E95 : 22 B4 C1    shld   ADRFIL    ; a uloz do hlavicky
2356/8E98 : E5          push   h        ; odpamataj
2357/8E99 : 21 72 C0      lxi    h,CURCH    ; adresa ukazatela na aktualny znak
2358/8E9C : 34          inr     m        ; preskoc oddelovac
2359/8E9D : CD 4B 81      call   RADR        ; precitaj koncovu adresu do DE
2360/8EA0 : E1          pop     h        ; obnov zaciatoctu adresu

```

```

2361/8EA1 : 7B          mov     a,e           ; odpocitaj: koniec-start
2362/8EA2 : 95          sub     l           ;
2363/8EA3 : 6F          mov     l,a           ;
2364/8EA4 : 7A          mov     a,d           ;
2365/8EA5 : 9C          sbb     h           ;
2366/8EA6 : 67          mov     h,a           ;
2367/8EA7 : DA 49 80     jc      PRBWTXT        ; ak je start > koniec, je to chyba
2368/8EAA : 22 B6 C1     shld   LENFIL         ; uloz dlzku-1 bloku
2369/8EAD : 2A 72 C0     lhld   CURCH         ; ukazatel na aktualny znak
2370/8EB0 : 23          inx     h           ; preskoc oddeľovac
2371/8EB1 : CD C0 81     call  TEXTHEAD       ; a presun nazov suboru do hlavicky
2372/8EB4 : CD 60 8F     call  PRGOUT         ; zapis subor na kazetu
2373/8EB7 : 21 B4 8D     lxi    h,T_MgfStop    ; priprav spravu "MG STOP"
2374/8EBA : 22 74 C0     shld   MESS          ;
2375/8EBD : C9          ret              ;
2376/8EBE :
2377/8EBE : ;-----
2378/8EBE : ; Nacitanie synchronizacie pred hlavickou suboru.
2379/8EBE : ; Ocakava sa aspon 64 Log.0 a nasledne prvý start bit Log.1. a byte 55h.
2380/8EBE : ; I: -
2381/8EBE : ; O: Z=1 - synchronizacia nacistana
2382/8EBE : ;     Z=0 - bol stlaceny klaves STOP
2383/8EBE : ; M: AF, L
2384/8EBE : CD CB 8E     ReadSyncH: call    ReadSync        ; nacistaj synchronizaciu
2385/8EC1 : C0          rnz              ; navrat, ak bol stlaceny klaves STOP
2386/8EC2 : CD FA 8E     call    LoadByte       ; precitaj byte
2387/8EC5 : FE 55       cpi     55h        ; musi to byt 55h
2388/8EC7 : C2 BE 8E     jnz     ReadSyncH       ; ak nie je, cakaj dalej
2389/8ECA : C9          ret              ;
2390/8ECB :
2391/8ECB : ;-----
2392/8ECB : ; Nacitanie synchronizacie.
2393/8ECB : ; Ocakava sa aspon 64 Log.0 a nasledne prvý start bit Log.1.
2394/8ECB : ; I: -
2395/8ECB : ; O: Z=1 - synchronizacia nacistana
2396/8ECB : ;     Z=0 - bol stlaceny klaves STOP
2397/8ECB : ; M: AF, L
2398/8ECB : DB F6       ReadSync:  in     SYS55C        ; precitaj stav portu C
2399/8ECD : 2F          cma              ; invertuj
2400/8ECE : E6 10       ani     X_STOP        ; a odmaskuj bit klavesu STOP
2401/8ED0 : C0          rnz              ; navrat, ak bol stlaceny
2402/8ED1 : 2E 40       mvi     l,64          ; aspon 64 bitov Log.0
2403/8ED3 : CD 49 8F     ReadSyncW0: call    ReadBit         ; precitaj bit
2404/8ED6 : DA CB 8E     jc      ReadSync        ; skoc, ak to bola Log.1
2405/8ED9 : 2D          dcr     l           ; zniz pocitadlo
2406/8EDA : C2 D3 8E     jnz     ReadSyncW0       ; a opakuj 64x
2407/8EDD : CD 49 8F     ReadSyncW1: call    ReadBit         ; precitaj bit

```

```

2408/8EE0 : D2 DD 8E          jnc      ReadSyncW1      ; a cakaj na prvý start bit Log.1
2409/8EE3 : C9                ret
2410/8EE4 :
2411/8EE4 : ;-----
2412/8EE4 : ; Nacitanie hlavicky suboru.
2413/8EE4 : ; I: -
2414/8EE4 : ; O: Z=1 - blok nacistany v poriadku, HL=HEAD
2415/8EE4 : ;     Z=0 - blok bol nacistany s chybou, alebo bol stlaceny klaves STOP
2416/8EE4 : ; M: AF, BC, DE
2417/8EE4 : 21 B2 C1      LoadHead:  lxi      h,HEAD          ; ukladacia adresa
2418/8EE7 : 11 0D 00          lxi      d,14-1          ; dlzka hlavicky
2419/8EEA : C3 C2 8D          jmp      LoadBlock      ; nacistaj blok hlavicky
2420/8EED :
2421/8EED : ;-----
2422/8EED : ; Nacitanie tela suboru.
2423/8EED : ; Rutina vola pred nahratim bloku vektor RAM, kde je napr. mozne dodatocne
2424/8EED : ; modifikovat ukladaci adresu v HL.
2425/8EED : ; I: [LENFIL]=dlzka suboru-1, [ADRFIL]=ukladacia adresa
2426/8EED : ; O: Z=1 - blok nacistany v poriadku, HL=HEAD
2427/8EED : ;     Z=0 - blok bol nacistany s chybou, alebo bol stlaceny klaves STOP
2428/8EED : ; M: AF, BC, DE
2429/8EED : 2A B6 C1      LoadBody:  lhld     LENFIL          ; dlzka suboru-1
2430/8EF0 : EB                xchg                     ; do DE
2431/8EF1 : 2A B4 C1          lhld     ADRFIL          ; cielova ukladacia adresa do HL
2432/8EF4 : CD F0 C0          call    RAM              ; zavolaj vektor RAM
2433/8EF7 : C3 C4 8D          jmp      LoadBlockC
2434/8EFA :
2435/8EFA : ;-----
2436/8EFA : ; Nacitanie jedneho bytu z MGF.
2437/8EFA : ; Jednotlive bity su v poradí od nulteho po siedmy.
2438/8EFA : ; Rutina je volana az po nacistani start bitu.
2439/8EFA : ; I: [SPEED]=urcenie dlzky pulzu
2440/8EFA : ; O: Z=1, A=nacistany byte
2441/8EFA : ;     Z=0, prerusene klavesom STOP
2442/8EFA : ; M: AF
2443/8EFA : C5                LoadByte:  push     b              ; odpamataj BC
2444/8EFB : 06 08            mvi      b,8              ; 8 bitov bytu
2445/8EFD : 0E 00            mvi      c,0              ; vynuluj vysledny byte
2446/8EFF : CD 49 8F      LoadByteL:  call    ReadBit          ; nacistaj bit
2447/8F02 : 79                mov      a,c              ; a vloz ho do vysledneho bytu
2448/8F03 : 1F                rar
2449/8F04 : 4F                mov      c,a
2450/8F05 : 05                dcr      b              ; opakuj 8x
2451/8F06 : C2 FF 8E          jnz      LoadByteL
2452/8F09 : DB F6            in       SYS55C          ; precitaj stav portu C
2453/8F0B : 2F                cma                     ; invertuj a odmaskuj
2454/8F0C : E6 10            ani      X_STOP          ; bit klavesu STOP - Z=0 : stlaceny

```

```

2455/8F0E : 79          mov     a,c           ; nacistany byte do A
2456/8F0F : C1          pop     b           ; obnov BC
2457/8F10 : C9          ret
2458/8F11 :
2459/8F11 : ;-----
2460/8F11 : ; Oblsuha prikazu BASIC - spustenie interpretera BASICu.
2461/8F11 : ; Format: BASIC
2462/8F11 : 21 00 98 BASIC:      lxi     h,Basic      ; adresa interpretera BASICu v ROM
2463/8F14 : 11 00 00          lxi     d,0           ; cielova adresa v RAM
2464/8F17 : 7E          BasicL:  mov     a,m           ; presun byte
2465/8F18 : 12          stax    d
2466/8F19 : 23          inx     h           ; posun ukazatele
2467/8F1A : 13          inx     d
2468/8F1B : 7C          mov     a,h           ; a pokracuj az do konca BASICu
2469/8F1C : FE BC          cpi     BasicEndHI
2470/8F1E : C2 17 8F          jnz     BasicL
2471/8F21 : C3 00 00          jmp     0           ; spusti interpreter BASICu
2472/8F24 :
2473/8F24 : ;-----
2474/8F24 : ; Rutina preskoci medzery na vstupe.
2475/8F24 : 2B          SSKIP:      dcx     h
2476/8F25 : 23          SSkipL:   inx     h
2477/8F26 : 7E          mov     a,m
2478/8F27 : FE 20          cpi     ' '
2479/8F29 : CA 25 8F          jz      SSkipL
2480/8F2C : C9          ret
2481/8F2D :
2482/8F2D : ;-----
2483/8F2D : ; Zapise Leader (uvodny ton pred hlavickou) na MGF.
2484/8F2D : ; 255 Log.0 bitov nasledovanych bytom 55h.
2485/8F2D : ; I: -
2486/8F2D : ; O: -
2487/8F2D : ; M: AF, B
2488/8F2D : 06 FF          WriteLeader: mvi     b,255      ; zapis 128 Log.0 na MGF
2489/8F2F : CD 41 8F          call    WriteBit0B
2490/8F32 : 3E 55          mvi     a,55h      ; zapis byte 55h na MGF
2491/8F34 : CD 7E 8D          call    WriteByte
2492/8F37 : C9          ret
2493/8F38 :
2494/8F38 : ;-----
2495/8F38 : ; Zapise Gap (medzeru medzi hlavickou a telom) na MGF.
2496/8F38 : ; 128 Log.0 bitov nasledovanych jednym log.1 bitom.
2497/8F38 : ; I: -
2498/8F38 : ; O: -
2499/8F38 : ; M: AF, B
2500/8F38 : 06 80          WriteGap:   mvi     b,128      ; zapis 128 Log.0 na MGF
2501/8F3A : CD 41 8F          call    WriteBit0B

```

```

2502/8F3D : CD A2 8D          call    WriteBit1      ; zapis Log.1 na MGF
2503/8F40 : C9                ret
2504/8F41 :
2505/8F41 : ;-----
2506/8F41 : ; Zapis B x Log.0 na MGF.
2507/8F41 : ; I: [SPEED]=dlzka jedneho pulzu (T), B=pocet
2508/8F41 : ; O: -
2509/8F41 : ; M: AF
2510/8F41 : CD 94 8D      WriteBit0B: call    WriteBit0      ; zapis Log.0 na MGF
2511/8F44 : 05                dcr      b              ; opakuje B krat
2512/8F45 : C2 41 8F          jnz     WriteBit0B
2513/8F48 : C9                ret
2514/8F49 :
2515/8F49 : ;-----
2516/8F49 : ; Nacitanie bitu z MGF.
2517/8F49 : ;
2518/8F49 : ;      |_____|_____|_____|_____|
2519/8F49 : ; I: [SPEED]=dlzka jedneho pulzu
2520/8F49 : ; O: CY=uroven precitaneho bitu
2521/8F49 : ; M: AF
2522/8F49 : DB F6      ReadBit:   in      SYS55C      ; precitaj stav portu C
2523/8F4B : 17                ral      ; bit X_MGF_IN do CY
2524/8F4C : DA 49 8F          jc      ReadBit      ; cakaj na zostupnu hranu
2525/8F4F : DB F6      ReadBitH:  in      SYS55C      ; precitaj stav portu C
2526/8F51 : 17                ral      ; bit X_MGF_IN do CY
2527/8F52 : D2 4F 8F          jnc     ReadBitH     ; cakaj na vzostupnu hranu
2528/8F55 : 3A 7C C0          lda      SPEED      ; A=konstanta dlzky pulzu
2529/8F58 : 17                ral      ; *2
2530/8F59 : C6 08          adi      8              ; +8
2531/8F5B : 00                nop
2532/8F5C : 00                nop
2533/8F5D : C3 62 8D          jmp     WaitBitA      ; precitaj bit z MGF
2534/8F60 :
2535/8F60 : ;-----
2536/8F60 : ; Zapis suboru na MGF.
2537/8F60 : ; I: [HEAD]=pripravena hlavicka suboru
2538/8F60 : ; O: -
2539/8F60 : ; M: vsetky
2540/8F60 : 06 FF      PRGOUT:   mvi      b,255      ; zapis 128 Log.0 na MGF
2541/8F62 : CD 41 8F          call    WriteBit0B
2542/8F65 : CD 2D 8F          call    WriteLeader    ; zapis Leader na MGF
2543/8F68 : 21 B2 C1          lxi     h,HEAD          ; adresa hlavicky
2544/8F6B : 11 0D 00          lxi     d,14-1         ; dlzka hlavicky
2545/8F6E : CD 6C 8D          call    TAPEOUT        ; zapis hlavicku na MGF
2546/8F71 : CD 38 8F          call    WriteGap       ; zapis medziblokovu medzeru
2547/8F74 : CD F0 C0          call    RAM            ; zavolaj vektor v RAM
2548/8F77 : 2A B6 C1          lhld   LENFIL         ; dlzka suboru

```

```

2549/8F7A : EB                xchg                ; do DE
2550/8F7B : 2A B4 C1          lhl             ADRFIL          ; zaciatočna adresa do HL
2551/8F7E : C3 6C 8D          jmp             TAPEOUT        ; zapis tela suboru na MGF
2552/8F81 :
2553/8F81 :                  ;-----
2554/8F81 :                  ; Ukončenie nactania bloku dat z MGF.
2555/8F81 :                  ; I: B=priebežne CRC
2556/8F81 :                  ; O: Z=1 - blok nactany v poriadku
2557/8F81 :                  ;     Z=0 - blok bol nactany s chybou, alebo bol stlaceny klaves STOP
2558/8F81 :                  ; M: AF, BC, DE
2559/8F81 : CD 49 8F          LoadBlockF:  call    ReadBit          ; nactaj "start bit"
2560/8F84 : CD FA 8E          call    LoadByte        ; nactaj CRC byte
2561/8F87 : A8                xra             b           ; porovnaj s priebežnym CRC
2562/8F88 : E1                pop             h           ; obnov ukladaciu adresu
2563/8F89 : C9                ret
2564/8F8A :
2565/8F8A :                  ;-----
2566/8F8A :                  ; Pokracovanie rutiny SHEAD.
2567/8F8A :                  ; Nasla sa hlavicka suboru, ktora nevyhovovala očakavanej.
2568/8F8A : CD B6 8B          SHeadB:    call    EOL           ; vypis hlavicku suboru z DR na obrazovku
2569/8F8D : CD A3 88          call    BEEP           ; pipni
2570/8F90 : C3 E2 8D          jmp             SHEAD        ; a dalej cakaj na spravnu hlavicku
2571/8F93 :
2572/8F93 :                  ;-----
2573/8F93 :                  ; Text spravy po ukončení prikazu MGLD/MGEND.
2574/8F93 : (MACRO)          T_MgfReadOK:  dm      "++ READING OK ++"
2574/8F93 : 2B 2B 20 52 45 41 44 49      db      "++ READING OK ++"
2574/8F9B : 4E 47 20 4F 4B 20 2B 2B      db
2574/8FA3 : 0D                db      CR
2575/8FA4 :
2576/8FA4 :                  ;-----
2577/8FA4 :                  ; Rutiny pre prenos obsahu ROM Modulu do pamate.
2578/8FA4 :                  ; Udaje prenosu sa zadavaju za volanie rutiny.
2579/8FA4 :                  ; CALL TRANSF
2580/8FA4 :                  ; DW      adresa zaciatku v ROM Module
2581/8FA4 :                  ; DW      počet prenasanych bytov -1
2582/8FA4 :                  ; DW      cielova adresa v RAM
2583/8FA4 : 3E 90          TRANSF:    mvi      a,ROMCTRL        ; inicializacia 8255 v ROM Module
2584/8FA6 : D3 FB          out      ROM55M
2585/8FA8 : E3                xthl             ; adresa parametrov do HL
2586/8FA9 : 7E                mov      a,m         ; nizsi byte adresy v ROM Module
2587/8FAA : D3 F9          out      ROM55B        ; nastav
2588/8FAC : 23                inx             h
2589/8FAD : 7E                mov      a,m         ; vyssi byte adresy v ROM Module
2590/8FAE : D3 FA          out      ROM55C        ; nastav
2591/8FB0 : 23                inx             h
2592/8FB1 : 4E                mov      c,m         ; počet prenasanych bytov do BC

```



```

2593/8FB2 : 23          inx      h
2594/8FB3 : 46          mov      b,m
2595/8FB4 : 04          inr      b          ; dlzka + 100h
2596/8FB5 : 23          inx      h
2597/8FB6 : 5E          mov      e,m          ; cielova adresa do DE
2598/8FB7 : 23          inx      h
2599/8FB8 : 56          mov      d,m
2600/8FB9 : 23          inx      h          ; HL=ukazuje za parametre volania
2601/8FBA : DB F8      TransfB: in      ROM55A          ; precitaj byte z ROM Modulu
2602/8FBC : 12          stax     d          ; uloz do pamate
2603/8FBD : 13          inx      d          ; posun ukazatel
2604/8FBE : 0B          dcx      b          ; zniz pocitadlo bytov
2605/8FBF : DB F9      in      ROM55B          ; zvys adresu v ROM Module
2606/8FC1 : 3C          inr      a
2607/8FC2 : D3 F9      out      ROM55B
2608/8FC4 : C2 CC 8F   jnz      TransfC          ; skoc, ak sa nizsi byte nevynuloval
2609/8FC7 : DB FA      in      ROM55C          ; inak, zvys aj vyssi byte
2610/8FC9 : 3C          inr      a
2611/8FCA : D3 FA      out      ROM55C
2612/8FCC : 78          TransfC: mov      a,b          ; preniesli sa uz vsetky byty?
2613/8FCD : B7          ora      a
2614/8FCE : C2 BA 8F   jnz      TransfB          ; nie, pokracuj
2615/8FD1 : 3E FF      mvi      a,0FFh          ; inak, deaktivuj ROM Modul
2616/8FD3 : D3 FA      out      ROM55C
2617/8FD5 : E3          xthl          ; navratova adresa na zasobnik
2618/8FD6 : C9          ret
2619/8FD7 :
2620/8FD7 :
;-----
2621/8FD7 :
; Obsluha prikazu JOB : prenesenie obsahu ROM Modulu do pamate.
2622/8FD7 :
; Format: JOB zzzzl1lllcccc
JOB:      mvi      a,ROMCTRL          ; inicializacia 8255 v ROM Module
          out      ROM55M
2624/8FD9 : D3 FB      call     RADR          ; precitaj adresu v ROM Module
2625/8FDB : CD 4B 81   mov      a,e          ; nastav ju tam
2626/8FDE : 7B          out      ROM55B          ; nizsi byte
2627/8FDF : D3 F9      mov      a,d
2628/8FE1 : 7A          out      ROM55C          ; vyssi byte
2629/8FE2 : D3 FA      call     RADR          ; precitaj dlzku
2630/8FE4 : CD 4B 81   mov      b,d          ; a presun do BC+100h
2631/8FE7 : 42          inr      b
2632/8FE8 : 04          mov      c,e
2633/8FE9 : 4B          push     b          ; odpamataj dlzku
2634/8FEA : C5          call     RADR          ; precitaj cielovu adresu
2635/8FEB : CD 4B 81   mov      h,d          ; a uloz do HL ako adresu spustenia
2636/8FEE : 62          mov      l,e
2637/8FEF : 6B          pop      b          ; obnov dlzku
2638/8FF0 : C1          call     TransfB          ; urob prenos
2639/8FF1 : CD BA 8F

```

```

2640/8FF4 :                ; Tu sa nikdy nevrati!
2641/8FF4 :                ; Instrukcie XTHL a RET na konci rutiny TransfB zabezpeca,
2642/8FF4 :                ; ze sa navrat urobi na adresu v HL.
2643/8FF4 :
2644/8FF4 :                ;-----
2645/8FF4 :                ; Toto je pozostatok povodnej rutiny SSKIP, z povoneho Monitora
2646/8FF4 : 2B             XSSKIP:      dcx      h
2647/8FF5 : 23             XSSkipL:     inx      h
2648/8FF6 : 7E             mov      a,m
2649/8FF7 : FE 20          cpi      ' '
2650/8FF9 : CA F5 8F       jz      XSSkipL
2651/8FFC : C9             ret
2652/8FFD :
2653/8FFD :                ;-----
2654/8FFD : 00             db      0
2655/8FFE : 55 52          db      "UR"                ; Urda... ;- )
2656/9000 :
2657/9000 :                ;-----
2658/9000 :
2703/9000 :
2705/9000 :                ;-----
2706/9000 :                ; vlozenie suboru obsluhy siete pre verziu `mato-lan.rom`
2707/9000 :                include "mato-lan.a8080"
(1) 1/9000 :                ;-----
(1) 2/9000 :                ; Tento zdrojovy kod a vypis vznikol disassemblovanim obsahov ROM pocitaca MATO.
(1) 3/9000 :                ; Pri komentovani kodu pomohli disassemblované vypisy od RaceSoft, Martin M1
(1) 4/9000 :                ; a Komentovany vypis Monitora PMD 85-2 od Vita Libovickeho a Jiriho Olmera.
(1) 5/9000 :                ; Vaccina mien navesti pisanych veľkymi pismenami bola prevzata z Komentovaneho
(1) 6/9000 :                ; vypisu Monitora PMD 85-2.
(1) 7/9000 :                ; Ostatne povodne nepomenovane navestia su v tvare Camel-Case.
(1) 8/9000 :
(1) 9/9000 :                ; RomBor 12/2025
(1) 10/9000 :                ;-----
(1) 11/9000 :
(1) 12/9000 :                ;-----
(1) 13/9000 :                ; * MATO LAN *
(1) 14/9000 :                ; Ovladac siete pre pocitac MATO.
(1) 15/9000 :                ; Obsahuje obsluhu prikazov SEND, READ, SET a DEMO.
(1) 16/9000 :                ;+++++
(1) 17/9000 :                ; V kode tohto sietoveho rozsirenia je chyba pri prijme programu v jazyku BASIC,
(1) 18/9000 :                ; kedy sa z ROM presunie BASIC aj s DEMO programom, takze sa vzdy nacistany
(1) 19/9000 :                ; program v BASICu prepise!
(1) 20/9000 :                ; Naviac sa podla prijateho programu nastavi systemova premenna konca programu,
(1) 21/9000 :                ; takze ani ten DEMO program nebude fungovat spravne.
(1) 22/9000 :                ;+++++
(1) 23/9000 :                ; Pridane prikazy SEND, READ a SET do tabulky `CmdTab` prepisali kod pre
(1) 24/9000 :                ; automaticke spustenie kodu nahraného suboru, ktoreho prve pismeno mena je '*'

```

```

(1) 25/9000 : ; (MgLdR). Takze nahrantie takehoto suboru, pravdepodobne skonci krachom.
(1) 26/9000 : ;-----
(1) 27/9000 :
(1) 28/9000 : ;-----
(1) 29/9000 : ; systemove premenne pre rutiny siete
(1) 30/9000 : =0C07DH LanFlag equ 0C07Dh ; priznak aktivovania siete 00h/0FFh
(1) 31/9000 : =0C1B0H LanIdS equ 0C1B0h ; ID stanice odosielatela
(1) 32/9000 : =0C1B1H LanIdR equ 0C1B1h ; ID stanice prijimatela
(1) 33/9000 : =0C1B2H LanDir equ 0C1B2h ; smer prenosu: R=prijem, T=odosielanie
(1) 34/9000 : =0C1B3H LanBlkTyp equ 0C1B3h ; typ bloku: T=Text, B=Basic, G=VideoRAM,
(1) 35/9000 : ; M=strojovy kod, D=data
(1) 36/9000 : ; Typ D sa spracuva rovnako ako M, len s tym
(1) 37/9000 : ; rozdielom, ze dany blok nespusti.
(1) 38/9000 : =0C1B4H LanBlkAdr equ 0C1B4h ; adresa zavciatku bloku
(1) 39/9000 : =0C1B6H LanBlkLen equ 0C1B6h ; dlzka bloku
(1) 40/9000 : =0C1B8H LanStoAdr equ 0C1B8h ; ukladacia adresa bloku
(1) 41/9000 : =0C1BCH LanSP equ 0C1BCh ; ulozenie SP pocas prijmu dat
(1) 42/9000 : =0C1BEH LanID equ 0C1BEh ; ID stanice siete
(1) 43/9000 : =0C1BFH LanErr equ 0C1BFh ; chybovy kod sietovej operacie
(1) 44/9000 : ; 0 - Transfer OK
(1) 45/9000 : ; '1' - Frame error
(1) 46/9000 : ; '2' - No ready for data
(1) 47/9000 : ; '3' - No ready for send
(1) 48/9000 : ; '4' - Check sum error
(1) 49/9000 : =0C1FEH LANTM1 equ 0C1FEh ; casovanie odosielania jedneho bitu
(1) 50/9000 : ; (pre start-bit sa pouzije tato hodnota,
(1) 51/9000 : ; pre ostatne bity dvojnásobok, pre stop-bit
(1) 52/9000 : ; stvornásobok)
(1) 53/9000 : =0C1FFH LANTM2 equ 0C1FFh ; casovanie bitov pri prijme
(1) 54/9000 :
(1) 55/9000 : =9000H C_HS_TO equ 9000h ; handshake timeout ~1s
(1) 56/9000 :
(1) 57/9000 : ;-----
(1) 58/9000 : ; Obsluha prikazu DEMO : presunutie BASICu a programu do RAM a spustenie
(1) 59/9000 : ; Format: DEMO
(1) 60/9000 : 21 00 98 DEMO: lxi h,Basic ; adresa interpretera BASICu v ROM
(1) 61/9003 : 11 00 00 lxi d,0 ; cielova adresa v RAM
(1) 62/9006 : CD 12 90 call MoveROM ; presun BASIC a DEMO program
(1) 63/9009 : 21 FD 27 lxi h,DemoPrgEnd ; nastav do systemovej premennej adresu
(1) 64/900C : 22 7A 5E shld BasPrgEnd ; konca programu
(1) 65/900F : C3 00 00 jmp 0 ; a spusti BASIC
(1) 66/9012 :
(1) 67/9012 : ;-----
(1) 68/9012 : ; Presunutie obsahu ROM do RAM
(1) 69/9012 : 7E MoveROM: mov a,m ; presun byte
(1) 70/9013 : 12 stax d
(1) 71/9014 : 23 inx h ; posun ukazatele

```

```

(1) 72/9015 : 13                inx      d
(1) 73/9016 : 7C                mov     a,h          ; a pokračuj az do konca ROM
(1) 74/9017 : FE C0             cpi     DemoEndHI
(1) 75/9019 : C2 12 90          jnz     MoveROM
(1) 76/901C : C9                ret
(1) 77/901D :
(1) 78/901D :
(1) 79/901D : ;-----
(1) 80/901D : ;+++++
(1) 81/901D : ;-----
(1) 82/901D : ; Obľuha prikazu SEND - odoslanie dat danemu adresatovi.
(1) 83/901D : ; Format: SEND AD/X ... TEXT          - X=G,B,T
(1) 84/901D : ;       SEND AD;OD-DO/X              - X=G,T,D,M
(1) 85/901D : ;       SEND AD;OD-DO;TO/X          - X=G,M,D
(1) 86/901D : ; AD - adresa/cislo adresata
(1) 87/901D : ; OD - pociatocna adresa
(1) 88/901D : ; DO - koncova adresa (1 byte za poslednym bytom bloku)
(1) 89/901D : ; TO - adresa ulozenia u adresata
(1) 89/901D : 21 46 83          SEND:    lxi     h,T_ErrAddr    ; priprav hlasienie "ERROR IN ADDRESS"
(1) 90/9020 : 22 74 C0          shld    MESS
(1) 91/9023 : 3E 52             mvi     a,'R'          ; nastav smer prenosu
(1) 92/9025 : 32 B2 C1          sta     LanDir         ; z pohľadu adresata - citanie
(1) 93/9028 : CD 9A 95          call   IsLanOn        ; je LAN aktivovana?
(1) 94/902B : CA 49 80          jz      PRBWTXT        ; ak nie, vypis "LAN OFF"
(1) 95/902E : CD 84 95          call   SkipSpcX       ; preskoc medzery na vstupe
(1) 96/9031 : CD 60 91          call   GetIdR         ; precitaj cislo stanice adresata
(1) 97/9034 : DA 49 80          jc      PRBWTXT        ; chyba? - vypis "ERROR IN ADDRESS"
(1) 98/9037 : 2A 72 C0          lhld   CURCH          ; ukazatel do vstupneho riadku
(1) 99/903A : 7E               mov     a,m            ; znak zo vsupu
(1) 100/903B : FE 2F           cpi     '/'            ; nasleduje '/'?
(1) 101/903D : C2 49 90          jnz     SendBlkRng     ; nie, skoc spracovat rozsah bloku
(1) 102/9040 : CD 0F 91          call   PrepBlkGBT     ; ano, bloky G, B alebo T
(1) 103/9043 : C2 6C 92          jnz     PrtNoCmd       ; chyba "NO COMMAND" pri nespr. zadani
(1) 104/9046 : C3 7C 90          jmp     GoSendBlk     ; skoc odoslat blok dat
(1) 105/9049 :
(1) 106/9049 : ;-----
(1) 107/9049 : ; Odoslanie bloku dat v zadanom rozsahu.
(1) 108/9049 : ; Pozadovany typ bloku je D, M, G.
(1) 109/9049 : ; D - blok sa iba prenesie
(1) 110/9049 : ; M - blok sa prenesie a spusti
(1) 111/9049 : ; G - blok sa prenesie do VRAM
(1) 112/9049 : CD 92 95          SendBlkRng: call   IncPtr      ; preskoc oddelovac
(1) 113/904C : CD 6E 91          call   GetBlkAdr      ; precitaj pociatocnu adresu bloku
(1) 114/904F : DA 49 80          jc      PRBWTXT        ; chyba? - vypis "ERROR IN ADDRESS"
(1) 115/9052 : CD 92 95          call   IncPtr         ; preskoc oddelovac
(1) 116/9055 : CD 76 91          call   GetBlkLen      ; precitaj koncovu adresu
(1) 117/9058 : DA 49 80          jc      PRBWTXT        ; chyba? - vypis "ERROR IN ADDRESS"
(1) 118/905B : 2A 72 C0          lhld   CURCH          ; HL=ukazatel vstupu

```

```

(1) 119/905E : 7E                mov    a,m          ; znak zo vstupu
(1) 120/905F : FE 2F              cpi      '/'          ; je to '/'?
(1) 121/9061 : C2 6D 90           jnz     SendBlkRngS   ; nie, pokracuje este cielova adresa
(1) 122/9064 : CD E5 90           call    PrepBlkStoAdr ; priprav predv. adr. ukal. G, D, T, M
(1) 123/9067 : C2 6C 92           jnz     PrtNoCmd       ; chyba "NO COMMAND" pri nespr. zadani
(1) 124/906A : C3 7C 90           jmp     GoSendBlk      ; skoc odoslat blok dat
(1) 125/906D :
(1) 126/906D : CD 92 95           SendBlkRngS: call    IncPtr      ; preskoc oddelovac
(1) 127/9070 : CD 7E 91           call    GetStoAdr      ; precitaj cielovu ukladaci adresu
(1) 128/9073 : DA 49 80           jc      PRBWTXT        ; chyba? - vypis "ERROR IN ADDRESS"
(1) 129/9076 : CD C5 90           call    ExpBlkGMD      ; ocakava sa "/" - G, M, D
(1) 130/9079 : C2 6C 92           jnz     PrtNoCmd       ; chyba "NO COMMAND" pri nespr. zadani
(1) 131/907C : CD 82 90           GoSendBlk: call    ProcSendBlk ; odosli pozadovany blok
(1) 132/907F : C3 49 80           jmp     PRBWTXT        ; vypis hlasienie o vysledku prenosu
(1) 133/9082 :
(1) 134/9082 :
(1) 135/9082 :
(1) 136/9082 :
(1) 137/9082 : 2A B6 C1           ProcSendBlk: lhld     LanBlkLen ; aktualne koncova adresa bloku
(1) 138/9085 : EB                xchg                    ; do DE
(1) 139/9086 : 2A B4 C1           lhld     LanBlkAdr      ; zaciatozna adresa bloku
(1) 140/9089 : 7B                mov      a,e            ; odpocitaj a vypocitaj dlzku bloku
(1) 141/908A : 95                sub      l              ;
(1) 142/908B : 6F                mov      l,a            ;
(1) 143/908C : 7A                mov      a,d            ;
(1) 144/908D : 9C                sbb      h              ;
(1) 145/908E : 67                mov      h,a            ;
(1) 146/908F : 22 B6 C1           shld     LanBlkLen      ; uloz dlzku bloku
(1) 147/9092 : D8                rc                     ; START>END - vypis "ERROR IN ADDRESS"
(1) 148/9093 : CD AC 90           call     SendHead       ; odosli hlavicku bloku
(1) 149/9096 : CD 3A 95           call     PrepErrMsg     ; priprav chybove hlasienie
(1) 150/9099 : 3A BF C1           lda      LanErr         ; bola chyba?
(1) 151/909C : A7                ana      a              ;
(1) 152/909D : C0                rnz                    ; ano, navrat
(1) 153/909E : 2A B6 C1           lhld     LanBlkLen      ; dlzka bloku do DE
(1) 154/90A1 : EB                xchg                    ;
(1) 155/90A2 : 2A B4 C1           lhld     LanBlkAdr      ; zaciatozna adresa
(1) 156/90A5 : CD A5 93           call     SendBlock      ; odosli telo bloku
(1) 157/90A8 : CD 3A 95           call     PrepErrMsg     ; priprav chybove hlasienie
(1) 158/90AB : C9                ret
(1) 159/90AC :
(1) 160/90AC :
(1) 161/90AC :
(1) 162/90AC :
(1) 163/90AC :
(1) 164/90AC :
(1) 165/90AC : 21 88 94           SendHead: lxi      h,T_LanWaitPls ; vypis "LAN ACTIVED - WAIT PLEASE"

```

```

(1) 166/90AF : 22 74 C0          shld    MESS
(1) 167/90B2 : CD 89 8A          call    PRTTEXT
(1) 168/90B5 : 3A BE C1          lda     LanID           ; nastav do hlavicky cislo odosielatela
(1) 169/90B8 : 32 B0 C1          sta     LanIdS
(1) 170/90BB : 21 B0 C1          lxi     h,LanIdS        ; adresa hlavicky
(1) 171/90BE : 11 0A 00          lxi     d,10            ; hlavicka ma 10 bytov
(1) 172/90C1 : CD A5 93          call    SendBlock       ; odosli hlavicku
(1) 173/90C4 : C9               ret
(1) 174/90C5 :
(1) 175/90C5 :                  ;-----
(1) 176/90C5 :                  ; Ocakava sa lomitko a typ bloku G, M alebo D.
(1) 177/90C5 :                  ; I: [CURCH]=ukazatel do vstupneho riadku
(1) 178/90C5 :                  ; O: Z=1, [LanBlkTyp]=typ bloku
(1) 179/90C5 :                  ;     Z=0 - chybny typ bloku
(1) 180/90C5 :                  ; M: HL, AF
(1) 181/90C5 : 2A 72 C0          ExpBlkGMD:  lhld    CURCH           ; ukazatel do vstupneho riadku
(1) 182/90C8 : 7E               mov     a,m             ; znak na vstupe
(1) 183/90C9 : FE 2F            cpi     '/'            ; je to '/'?
(1) 184/90CB : CA D5 90          jz      ExpBlkGMD0      ; ano, skoc dalej
(1) 185/90CE : 21 92 83          lxi     h,T_NoCmd       ; chyba - priprav hlasenie "NO COMMAND"
(1) 186/90D1 : 22 74 C0          shld    MESS
(1) 187/90D4 : C9               ret
(1) 188/90D5 :
(1) 189/90D5 : CD 92 95          ExpBlkGMD0: call    IncPtr           ; preskoc oddelovac
(1) 190/90D8 : 7E               mov     a,m             ; znak na vstupe
(1) 191/90D9 : 32 B3 C1          sta     LanBlkTyp       ; uloz typ bloku
(1) 192/90DC : FE 47            cpi     'G'             ; povolene su G, M a D
(1) 193/90DE : C8               rz
(1) 194/90DF : FE 4D            cpi     'M'
(1) 195/90E1 : C8               rz
(1) 196/90E2 : FE 44            cpi     'D'
(1) 197/90E4 : C9               ret
(1) 198/90E5 :
(1) 199/90E5 :                  ;-----
(1) 200/90E5 :                  ; Pripravi predvolenu adresu ukaldania.
(1) 201/90E5 :                  ; Ocakava sa typ bloku G, D, T alebo M.
(1) 202/90E5 :                  ; I: [CURCH]=ukazatel do vstupneho riadku
(1) 203/90E5 :                  ; O: Z=1, [LanBlkTyp]=typ bloku, [LanStoAdr]=ukladacia adresa
(1) 204/90E5 :                  ;     Z=0 - chybny typ bloku
(1) 205/90E5 :                  ; M: HL, AF
(1) 206/90E5 : CD 92 95          PrepBlkStoAdr: call   IncPtr           ; preskoc oddelovac
(1) 207/90E8 : 7E               mov     a,m             ; znak na vstupe
(1) 208/90E9 : 32 B3 C1          sta     LanBlkTyp       ; uloz typ bloku
(1) 209/90EC : FE 47            cpi     'G'
(1) 210/90EE : CA 01 91          jz      PrepBlkStoAdrG
(1) 211/90F1 : FE 44            cpi     'D'
(1) 212/90F3 : CA 08 91          jz      PrepBlkStoAdrX

```

```

(1) 213/90F6 : FE 54          cpi      'T'
(1) 214/90F8 : CA 08 91      jz       PrepBlkStoAdrX
(1) 215/90FB : FE 4D          cpi      'M'
(1) 216/90FD : CA 08 91      jz       PrepBlkStoAdrX
(1) 217/9100 : C9            ret
(1) 218/9101 :
(1) 219/9101 : 21 00 C0      PrepBlkStoAdrG: lxi      h,VRAM          ; pre blok G je to adresa VRAM
(1) 220/9104 : 22 B8 C1      shld     LanStoAdr
(1) 221/9107 : C9            ret
(1) 222/9108 :
(1) 223/9108 : 2A B4 C1      PrepBlkStoAdrX: lhld     LanBlkAdr       ; pre ostatne typy je to zadana
(1) 224/910B : 22 B8 C1      shld     LanStoAdr       ; pociatocna adresa
(1) 225/910E : C9            ret
(1) 226/910F :
(1) 227/910F :
(1) 228/910F :
(1) 229/910F :
(1) 230/910F :
(1) 231/910F :
(1) 232/910F :
(1) 233/910F : CD 92 95      PrepBlkGBT:   call     IncPtr          ; preskoc oddelovc
(1) 234/9112 : 7E            mov      a,m              ; prevezmi typ bloku
(1) 235/9113 : 32 B3 C1      PrepBlkGB:   sta      LanBlkTyp       ; zapis do hlavicky
(1) 236/9116 : FE 47          cpi      'G'              ; je to typ Grafika (VRAM)?
(1) 237/9118 : C2 1E 91      jnz      PrepBlkBT       ; nie, skoc dalej
(1) 238/911B : C3 2C 91      jmp      PrepBlkG        ; ano, priprav data bloku Grafika
(1) 239/911E :
(1) 240/911E : FE 42          PrepBlkBT:   cpi      'B'              ; je to typ Basic?
(1) 241/9120 : C2 26 91      jnz      PrepBlkT        ; nie, skoc dalej
(1) 242/9123 : C3 3C 91      jmp      PrepBlkB        ; ano, priprav data bloku Basic
(1) 243/9126 :
(1) 244/9126 : FE 54          PrepBlkT:    cpi      'T'              ; je to typ Text?
(1) 245/9128 : CA 4C 91      jz       PrepBlkT0       ; ano, priprav data bloku Text
(1) 246/912B : C9            ret                          ; vrat sa s chybou - Z=0
(1) 247/912C :
(1) 248/912C :
(1) 249/912C :
(1) 250/912C :
(1) 251/912C :
(1) 252/912C :
(1) 253/912C : 21 00 C0      PrepBlkG:    lxi      h,VRAM          ; adresa bloku je VRAM
(1) 254/912F : 22 B4 C1      shld     LanBlkAdr
(1) 255/9132 : 22 B8 C1      shld     LanStoAdr
(1) 256/9135 : 21 00 FD      lxi      h,VRAM+244*64    ; odosle sa 244 uR
(1) 257/9138 : 22 B6 C1      shld     LanBlkLen
(1) 258/913B : C9            ret
(1) 259/913C :

```

```

(1) 260/913C : ;-----
(1) 261/913C : ; Pripravi do hlavicky parameter prenasanych bloku B.
(1) 262/913C : ; I: -
(1) 263/913C : ; O: [LanBlkAdr]=BasPrgAdr, [LanStoAdr]=BasPrgAdr, [LanBlkLen]=[BasPrgEnd]
(1) 264/913C : ; M: HL
(1) 265/913C : 21 00 24 PrepBlkB: lxi h,BasPrgAdr ; adresa bloku je zaciatok
(1) 266/913F : 22 B4 C1 shld LanBlkAdr ; programu v BASICu
(1) 267/9142 : 22 B8 C1 shld LanStoAdr
(1) 268/9145 : 2A 7A 5E lhld BasPrgEnd ; koncova adresa programu v BASICu
(1) 269/9148 : 22 B6 C1 shld LanBlkLen
(1) 270/914B : C9 ret
(1) 271/914C :
(1) 272/914C : ;-----
(1) 273/914C : ; Pripravi do hlavicky parameter prenasanych bloku T.
(1) 274/914C : ; I: -
(1) 275/914C : ; O: [LanBlkAdr]=[CURCH], [LanStoAdr]=[CURCH], [LanBlkLen]=dlzka textu
(1) 276/914C : ; M: HL, AF
(1) 277/914C : CD 92 95 PrepBlkT0: call IncPtr ; preskoc oddeľovac
(1) 278/914F : 22 B4 C1 shld LanBlkAdr ; adresa bloku je aktualny ukazatel
(1) 279/9152 : 22 B8 C1 shld LanStoAdr ; vstupu
(1) 280/9155 : 7E PrepBlkT0L: mov a,m ; najdi koniec textu v DR
(1) 281/9156 : 23 inx h
(1) 282/9157 : FE 0D cpi CR
(1) 283/9159 : C2 55 91 jnz PrepBlkT0L
(1) 284/915C : 22 B6 C1 shld LanBlkLen ; a uloz ako koncovu adresu
(1) 285/915F : C9 ret
(1) 286/9160 :
(1) 287/9160 : ;-----
(1) 288/9160 : ; Precitanie cisla stanice prijemcu zo vstupu ako 2xHEX.
(1) 289/9160 : ; I: [CURCH]=ukazatel do vstupneho riadku
(1) 290/9160 : ; O: CY=1 pri chybe
(1) 291/9160 : ; CY=0 - OK
(1) 292/9160 : ; [CURCH]=ukazatel do vstupneho riadku za precitanym cislom
(1) 293/9160 : ; [LanIdR]=cislo stanice prijemcu
(1) 294/9160 : ; M: AF, B, HL
(1) 295/9160 : 2A 72 C0 GetIdR: lhld CURCH ; HL=ukazatel na vstup
(1) 296/9163 : CD F7 80 call PAIRIN ; spracuj 2xHEX
(1) 297/9166 : 22 72 C0 shld CURCH ; uloz novy ukazatel
(1) 298/9169 : D8 rc ; chyba, navrat s CY
(1) 299/916A : 32 B1 C1 sta LanIdR ; uloz cislo stanice prijemcu
(1) 300/916D : C9 ret
(1) 301/916E :
(1) 302/916E : ;-----
(1) 303/916E : ; Precitanie zaciatočnej adresy bloku dat.
(1) 304/916E : ; I: [CURCH]=ukazatel do vstupneho riadku
(1) 305/916E : ; O: CY=1 pri chybe
(1) 306/916E : ; CY=0 - OK

```



```

(1) 307/916E : ; [CURCH]=ukazatel do vstupneho riadku za precitany cislom
(1) 308/916E : ; [LanBlkAdr]=precitana hodnota
(1) 309/916E : ; M: AF, B, DE, HL
(1) 310/916E : CD 86 91 GetBlkAdr: call GetAdr ; precitaj 4xHEX zo vstupu ako adresu
(1) 311/9171 : D8 rc ; pri chybe, navrat
(1) 312/9172 : 22 B4 C1 shld LanBlkAdr ; uloz pociatocnu adresu
(1) 313/9175 : C9 ret
(1) 314/9176 :
(1) 315/9176 : ;-----
(1) 316/9176 : ; Precitanie koncovej adresy bloku dat.
(1) 317/9176 : ; I: [CURCH]=ukazatel do vstupneho riadku
(1) 318/9176 : ; O: CY=1 pri chybe
(1) 319/9176 : ; CY=0 - OK
(1) 320/9176 : ; [CURCH]=ukazatel do vstupneho riadku za precitany cislom
(1) 321/9176 : ; [LanBlkLen]=precitana hodnota
(1) 322/9176 : ; M: AF, B, DE, HL
(1) 323/9176 : CD 86 91 GetBlkLen: call GetAdr ; precitaj 4xHEX zo vstupu ako adresu
(1) 324/9179 : D8 rc ; pri chybe, navrat
(1) 325/917A : 22 B6 C1 shld LanBlkLen ; uloz koncovu adresu
(1) 326/917D : C9 ret
(1) 327/917E :
(1) 328/917E : ;-----
(1) 329/917E : ; Precitanie cielovej adresy bloku dat.
(1) 330/917E : ; I: [CURCH]=ukazatel do vstupneho riadku
(1) 331/917E : ; O: CY=1 pri chybe
(1) 332/917E : ; CY=0 - OK
(1) 333/917E : ; [CURCH]=ukazatel do vstupneho riadku za precitany cislom
(1) 334/917E : ; [LanStoAdr]=precitana hodnota
(1) 335/917E : ; M: AF, B, DE, HL
(1) 336/917E : CD 86 91 GetStoAdr: call GetAdr ; precitaj 4xHEX zo vstupu ako adresu
(1) 337/9181 : D8 rc ; pri chybe, navrat
(1) 338/9182 : 22 B8 C1 shld LanStoAdr ; uloz ukladaciu adresu
(1) 339/9185 : C9 ret
(1) 340/9186 :
(1) 341/9186 : ;-----
(1) 342/9186 : ; Precitanie 4xHEX zo vstupu ako adresu.
(1) 343/9186 : ; I: [CURCH]=ukazatel do vstupneho riadku
(1) 344/9186 : ; O: CY=1 pri chybe
(1) 345/9186 : ; CY=0 - OK
(1) 346/9186 : ; [CURCH]=ukazatel do vstupneho riadku za precitany cislom
(1) 347/9186 : ; HL=precitana hodnota
(1) 348/9186 : ; M: AF, B, DE, HL
(1) 349/9186 : 2A 72 C0 GetAdr: lhld CURCH ; HL=ukazatel na vstup
(1) 350/9189 : CD 09 81 call ADRIN ; spracuj 4xHEX
(1) 351/918C : D8 rc ; chyba, navrat s CY
(1) 352/918D : 22 72 C0 shld CURCH ; uloz novy ukazatel
(1) 353/9190 : EB xchg ; precitana hodnota do HL

```

```

(1) 354/9191 : C9                                ret
(1) 355/9192 :
(1) 356/9192 : ;-----
(1) 357/9192 : ;+++++
(1) 358/9192 : ;-----
(1) 359/9192 : ; Obluha prikazu READ - prijem dat od daneho odosielatela.
(1) 360/9192 : ; Format: READ AD/X                      - X=G,B
(1) 361/9192 : ;       READ AD;OD-DO/X              - X=G,T,D,M
(1) 362/9192 : ;       READ AD;OD-DO;TO/X           - X=G,M,D
(1) 363/9192 : ; AD - adresa/cislo odosielatela
(1) 364/9192 : ; OD - pociatocna adresa
(1) 365/9192 : ; DO - koncova adresa (1 byte za poslednym bytom bloku)
(1) 366/9192 : ; TO - adresa ulozenia u tejto stanice
(1) 367/9192 : 21 46 83      READ:      lxi      h,T_ErrAddr      ; priprav hlasenie "ERROR IN ADDRESS"
(1) 368/9195 : 22 74 C0      shld      MESS
(1) 369/9198 : 3E 54        mvi      a,'T'          ; nastav smer prenosu
(1) 370/919A : 32 B2 C1      sta      LanDir        ; z pohladu adresata - odosielanie
(1) 371/919D : CD 9A 95      call     IsLanOn       ; je LAN aktivovana?
(1) 372/91A0 : CA 49 80      jz       PRBWTXT       ; chyba? - vypis "LAN OFF"
(1) 373/91A3 : CD 84 95      call     SkipSpcX      ; preskoc medzery na vstupe
(1) 374/91A6 : CD 60 91      call     GetIdR       ; precitaj cislo stanice adresata
(1) 375/91A9 : DA 49 80      jc       PRBWTXT       ; chyba? - vypis "ERROR IN ADDRESS"
(1) 376/91AC : 2A 72 C0      lhld     CURCH        ; ukazatel do vstupneho riadku
(1) 377/91AF : 7E          mov      a,m           ; znak zo vsupu
(1) 378/91B0 : FE 2F        cpi      '/'          ; nasleduje '/'?
(1) 379/91B2 : C2 C7 91      jnz      ReadBlkRng     ; nie, skoc spracovat rozsah bloku
(1) 380/91B5 : CD 92 95      call     IncPtr        ; preskoc oddelovac
(1) 381/91B8 : 7E          mov      a,m           ; znak zo vstupu
(1) 382/91B9 : FE 54        cpi      'T'          ; je to T? - pre READ je nepovoleny
(1) 383/91BB : CA 6C 92      jz       PrtNoCmd      ; chyba "NO COMMAND" pri nespr. zadani
(1) 384/91BE : CD 13 91      call     PrepBlkGB     ; su povolene bloky G alebo B
(1) 385/91C1 : C2 6C 92      jnz      PrtNoCmd      ; chyba "NO COMMAND" pri nespr. zadani
(1) 386/91C4 : C3 FA 91      jmp      GoReadBlk     ; skoc prijat blok dat
(1) 387/91C7 :
(1) 388/91C7 : ;-----
(1) 389/91C7 : ; Prijatie bloku dat v zadanom rozsahu.
(1) 390/91C7 : ; Pozadovany typ bloku je D, M, G.
(1) 391/91C7 : ; D - blok sa iba prenesie
(1) 392/91C7 : ; M - blok sa prenesie a spusti
(1) 393/91C7 : ; G - blok sa prenesie do VRAM
(1) 394/91C7 : CD 92 95      ReadBlkRng: call     IncPtr        ; preskoc oddelovac
(1) 395/91CA : CD 6E 91      call     GetBlkAdr     ; precitaj pociatocnu adresu bloku
(1) 396/91CD : DA 6C 92      jc       PrtNoCmd      ; chyba "NO COMMAND" pri nespr. zadani
(1) 397/91D0 :              ; tu malo byt - jc PRBWTXT
(1) 398/91D0 : CD 92 95      call     IncPtr        ; preskoc oddelovac
(1) 399/91D3 : CD 76 91      call     GetBlkLen     ; precitaj koncovu adresu
(1) 400/91D6 : DA 49 80      jc       PRBWTXT       ; chyba? - vypis "ERROR IN ADDRESS"

```

```

(1) 401/91D9 : 2A 72 C0          lhld   CURCH           ; HL=ukazatel vstupu
(1) 402/91DC : 7E                mov    a,m             ; znak zo vstupu
(1) 403/91DD : FE 2F             cpi     '/'           ; je to '/'?
(1) 404/91DF : C2 EB 91          jnz     ReadBlkRngS    ; nie, pokracuje este cielova adresa
(1) 405/91E2 : CD E5 90          call   PrepBlkStoAdr   ; priprav predv. adr. ukal. G, D, T, M
(1) 406/91E5 : C2 6C 92          jnz     PrtNoCmd       ; chyba "NO COMMAND" pri nespr. zadani
(1) 407/91E8 : C3 FA 91          jmp     GoReadBlk      ; skoc prijat blok dat
(1) 408/91EB :
(1) 409/91EB : CD 92 95          ReadBlkRngS: call   IncPtr        ; preskoc oddelovac
(1) 410/91EE : CD 7E 91          call   GetStoAdr       ; precitaj cielovu ukladaci adresu
(1) 411/91F1 : C2 49 80          jnz     PRBWTXT        ; chyba? - vypis "NO COMMAND"
(1) 412/91F4 : CD C5 90          call   ExpBlkGMD       ; ocakava sa "/" - G, M, D
(1) 413/91F7 : C2 49 80          jnz     PRBWTXT        ; chyba? - vypis "NO COMMAND"
(1) 414/91FA :
(1) 415/91FA : CD AC 90          GoReadBlk: call   SendHead      ; odosli najprv hlavicku pozadov. bloku
(1) 416/91FD : CD 3A 95          call   PrepErrMsg     ; priprav chybove hlaseenie
(1) 417/9200 : 3A BF C1          lda     LanErr         ; bola chyba?
(1) 418/9203 : A7                ana     a
(1) 419/9204 : C2 49 80          jnz     PRBWTXT        ; ak ano, vypis chybove hlaseenie
(1) 420/9207 : CD A3 88          call   BEEP           ; nebola chyba, pipni
(1) 421/920A : CD 7E 92          call   RecvHead       ; prijmi hlavicku od odosielatela
(1) 422/920D : CD 3A 95          call   PrepErrMsg     ; priprav chybove hlaseenie
(1) 423/9210 : 3A BF C1          lda     LanErr         ; bola chyba?
(1) 424/9213 : A7                ana     a
(1) 425/9214 : C2 49 80          jnz     PRBWTXT        ; ak ano, vypis chybove hlaseenie
(1) 426/9217 : CD B8 92          call   RecvReqBlock   ; inak, skoc prijat dany blok dat
(1) 427/921A : CD 3A 95          call   PrepErrMsg     ; priprav chybove hlaseenie
(1) 428/921D : C3 49 80          jmp     PRBWTXT        ; vypis chybove hlaseenie
(1) 429/9220 :
(1) 430/9220 :
(1) 431/9220 :
(1) 432/9220 :
(1) 433/9220 :
(1) 434/9220 :
(1) 435/9220 :
(1) 436/9220 : C5
(1) 437/9221 : 47
(1) 438/9222 : 0B
(1) 439/9223 : 79
(1) 440/9224 : B0
(1) 441/9225 : C2 22 92
(1) 442/9228 : C1
(1) 443/9229 :
(1) 444/9229 : 3A 7D C0          lda     LanFlag       ; je LAN aktivovana?
(1) 445/922C : A7                ana     a
(1) 446/922D : C8                rz
(1) 447/922E : DB F6          in      SYS55C        ; zisti stav linky

;-----
;*****
;-----
; Rutina testuje, ci neprichadzaju nejake data.
; Je volana z rutiny INKLAV.
; I: A=zdrzanie
ProcLanReq:  push    b           ; povodna rutina zdrzania
              mov     b,a
ProcLanReqD: dcx     b
              mov     a,c
              ora     b
              jnz     ProcLanReqD
              pop     b

```

```

(1) 448/9230 : 17          ral          ; X_MGF_IN do CY
(1) 449/9231 : D0          rnc          ; navrat, ziadny prenos na linke
(1) 450/9232 :
(1) 451/9232 : E1          pop          h          ; zahod navratovu adresu
(1) 452/9233 : 21 00 00    lxi          h,0        ; odpamataj aktualne SP
(1) 453/9236 : 39          dad          sp
(1) 454/9237 : 22 BC C1    shld         LanSP
(1) 455/923A : 31 00 7F    lxi          sp,7F00h      ; nastav nove SP
(1) 456/923D :
(1) 457/923D : CD 75 92    call         RecvHeadM      ; prijmi hlavicku
(1) 458/9240 : CD 3A 95    call         PrepErrMsg      ; priprav chybove hlasienie
(1) 459/9243 : 3A BF C1    lda          LanErr      ; bola chyba?
(1) 460/9246 : A7          ana          a
(1) 461/9247 : CA 50 92    jz          ProcLanReqP      ; nie, skoc dalej
(1) 462/924A :
(1) 463/924A : CD 9F 93    call         Delay290        ; male zdrzanie
(1) 464/924D : C3 60 92    jmp          ProcLanReqM      ; skoc vypisat hlasienie
(1) 465/9250 :
(1) 466/9250 : CD 9B 92    ProcLanReqP: call         ProcReqTrans      ; vykonaj prenos dat
(1) 467/9253 : CD 3A 95    call         PrepErrMsg      ; priprav chybove hlasienie
(1) 468/9256 : 3A BF C1    lda          LanErr      ; bola chyba?
(1) 469/9259 : A7          ana          a
(1) 470/925A : CA 60 92    jz          ProcLanReqM
(1) 471/925D :
(1) 472/925D : CD 9F 93    ProcLanReqM: call         Delay290        ; male zdrzanie
(1) 473/9260 : CD 89 8A    call         PRTTEXT        ; vypis hlasienie
(1) 474/9263 :
(1) 475/9263 : 2A BC C1    lhld         LanSP          ; obnov adresu zasobnika
(1) 476/9266 : F9          sphl
(1) 477/9267 : E1          pop          h          ; "obnov" HL a BC
(1) 478/9268 : C1          pop          b
(1) 479/9269 : C3 A1 84    jmp          INKLAV          ; a skoc nazad do klavesnicovej rutiny
(1) 480/926C :
(1) 481/926C :
(1) 482/926C :
(1) 483/926C : 21 92 83    ;-----
; Vypisanie spravy "NO COMMAND" do DR a navrat do Monitora.
PrtNoCmd:      lxi          h,T_NoCmd
(1) 484/926F :              ; pokračuj v PrtMessage
(1) 485/926F :
(1) 486/926F :
(1) 487/926F :
(1) 488/926F :
(1) 489/926F :
(1) 490/926F :
(1) 491/926F : 22 74 C0    ;-----
; Vypisanie spravy/hlasienia do DR a navrat do Monitora.
; I: HL=adresa textu
; O: -
; M: vsetky
PrtMessage:    shld         MESS          ; nastav adresu hlasienia
(1) 492/9272 : C3 49 80    jmp          PRBWTXT        ; vypis dane hlasienie
(1) 493/9275 :
(1) 494/9275 :
;-----

```

```

(1) 495/9275 :
(1) 496/9275 : 21 88 94
(1) 497/9278 : 22 74 C0
(1) 498/927B : CD 89 8A
(1) 499/927E : AF
(1) 500/927F : 32 BF C1
(1) 501/9282 : 21 B0 C1
(1) 502/9285 : 11 0A 00
(1) 503/9288 : CD C3 94
(1) 504/928B : 3A BF C1
(1) 505/928E : A7
(1) 506/928F : C0
(1) 507/9290 : CD D8 94
(1) 508/9293 : B8
(1) 509/9294 : C8
(1) 510/9295 : 3E 34
(1) 511/9297 : 32 BF C1
(1) 512/929A : C9
(1) 513/929B :
(1) 514/929B :
(1) 515/929B :
(1) 516/929B : 3A B2 C1
(1) 517/929E : FE 52
(1) 518/92A0 : CA B8 92
(1) 519/92A3 : 3A B0 C1
(1) 520/92A6 : 32 B1 C1
(1) 521/92A9 : 3A BE C1
(1) 522/92AC : 32 B0 C1
(1) 523/92AF :
(1) 524/92AF :
(1) 525/92AF : 3E 54
(1) 526/92B1 : 32 B2 C1
(1) 527/92B4 : CD 82 90
(1) 528/92B7 : C9
(1) 529/92B8 :
(1) 530/92B8 :
(1) 531/92B8 :
(1) 532/92B8 : 3A B3 C1
(1) 533/92BB : FE 54
(1) 534/92BD : CA C8 92
(1) 535/92C0 : FE 47
(1) 536/92C2 : CA F7 92
(1) 537/92C5 : C3 4E 93
(1) 538/92C8 :
(1) 539/92C8 :
(1) 540/92C8 :
(1) 541/92C8 : 3E 0D

; Prijatie hlavicky bloku.
RecvHeadM:    lxi    h,T_LanWaitPls ; vypis "LAN ACTIVED - WAIT PLEASE"
              shld   MESS
              call   PRTTEXT
RecvHead:     xra    a                ; vynuluj chybu
              sta    LanErr
              lxi    h,LanIdS        ; adresa hlavicky
              lxi    d,10            ; hlavicka ma 10 bytov
              call   RecvBlock       ; prijmi hlavicku
              lda    LanErr          ; bol prijem v poriadku?
              ana    a
              rnz    ; navrat, ak nie
              call   RecvByte        ; precitaj este CRC
              cmp    b              ; porovnaj s vypocitany
              rz     ; OK, navrat
              mvi    a,'4'          ; inak, priprav chybu `Check sum error`
              sta    LanErr
              ret

;-----
; Vykonanie prenosu dat podla udajov v hlavicke.
ProcReqTrans: lda    LanDir          ; smer prenosu z nasho pohladu
              cpi    'R'            ; prijem?
              jz     RecvReqBlock    ; ano, skoc prijat dany blok dat
              lda    LanIdS          ; pred odoslanim vyziadaneho bloku
              sta    LanIdR          ; "zamen" cisla odosilately/prijimateľa
              lda    LanID
              sta    LanIdS
; Na tomto mieste boli v dochovanej ROM byty 28h a 50h, co je zjavny nezmysel.
; Logicky tam patri nastavenie smeru prenosu.
              mvi    a,'T'          ; nastav smer prenosu
              sta    LanDir          ; (asi by to tu uz nemuselo byt)
              call   ProcSendBlk     ; odosli pozadovany blok
              ret

;-----
; Prijatie bloku dat podla typu.
RecvReqBlock: lda    LanBlkTyp       ; typ bloku
              cpi    'T'            ; ide o Text?
              jz     RecvText        ; ano, skoc ho prijat
              cpi    'G'            ; ide o Grafiku (VRAM)?
              jz     RecvVram        ; ano, skoc prijat data VRAM
              jmp    RecvOther       ; skoc pre prijem M, D alebo B

;-----
; Prijem textu.
RecvText:     mvi    a,CR            ; odriadkuj

```

(1)	542/92CA : CD 00 85	call	PRTOUT	
(1)	543/92CD : 2A B6 C1	lhld	LanBlkLen	; dlzka odosielaného textu
(1)	544/92D0 : 06 00	mvi	b,0	; vynuluj CRC
(1)	545/92D2 : CD D8 94	RecvTextL:	call	RecvByte ; prijmi byte
(1)	546/92D5 : 5F	mov	e,a	; uloz do E
(1)	547/92D6 : 80	add	b	; modifikuj CRC
(1)	548/92D7 : 47	mov	b,a	; a uloz nazad do B
(1)	549/92D8 : 3A BF C1	lda	LanErr	; bola chyba?
(1)	550/92DB : A7	ana	a	
(1)	551/92DC : C0	rnz		; navrat, ak ano
(1)	552/92DD : 7B	mov	a,e	; prijaty znak do A
(1)	553/92DE : CD 00 85	call	PRTOUT	; vypis ho
(1)	554/92E1 : 2B	dcx	h	; zniz pocitadlo dlzky textu
(1)	555/92E2 : 7D	mov	a,l	; a kym nie je nulove,
(1)	556/92E3 : B4	ora	h	
(1)	557/92E4 : C2 D2 92	jnz	RecvTextL	; pokracuj v prijme
(1)	558/92E7 : 3E 0D	mvi	a,CR	; odriadkuj
(1)	559/92E9 : CD 00 85	call	PRTOUT	
(1)	560/92EC : CD D8 94	call	RecvByte	; prijmi este CRC
(1)	561/92EF : B8	cmp	b	; a porovnaj s vypocitanym
(1)	562/92F0 : C8	rz		; navrat, ak su zhodne
(1)	563/92F1 : 3E 34	mvi	a,'4'	; inak, nastav chybu `Check sum error`
(1)	564/92F3 : 32 BF C1	sta	LanErr	
(1)	565/92F6 : C9	ret		
(1)	566/92F7 :			
(1)	567/92F7 :			
(1)	568/92F7 :			
(1)	569/92F7 :			
(1)	570/92F7 :			
(1)	571/92F7 : CD A7 85	RecvVram:	call	ERASE ; zmaz obrazovku
(1)	572/92FA : 2A B6 C1	lhld	LanBlkLen	; dlzka prenasanych dat
(1)	573/92FD : EB	xchg		; do DE
(1)	574/92FE : 2A B8 C1	lhld	LanStoAdr	; cielova adresa do HL
(1)	575/9301 : 06 00	mvi	b,0	; vynuluj CRC
(1)	576/9303 : 0E 30	RecvVramA:	mvi	c,48 ; 48 bytov na uR
(1)	577/9305 : E5	push	h	; odpamataj cielovu adresu
(1)	578/9306 : CD D8 94	RecvVramB:	call	RecvByte ; prijmi byte
(1)	579/9309 : 77	mov	m,a	; uloz
(1)	580/930A : 80	add	b	; modifikuj CRC
(1)	581/930B : 47	mov	b,a	; a vrat do B
(1)	582/930C : 3A BF C1	lda	LanErr	; bola chyba?
(1)	583/930F : A7	ana	a	
(1)	584/9310 : CA 15 93	jz	RecvVramD	; nie, skoc dalej
(1)	585/9313 : E1	RecvVramC:	pop	h ; ak ano, ukonci prijem a vypis chybu
(1)	586/9314 : C9	ret		
(1)	587/9315 :			
(1)	588/9315 : 23	RecvVramD:	inx	h ; posun ukazatel

```

(1) 589/9316 : 1B          dcx    d          ; zniz celkovy pocet prijimanych bytov
(1) 590/9317 : 7B          mov    a,e
(1) 591/9318 : B2          ora     d
(1) 592/9319 : CA 42 93    jz      RecvVramF      ; ak je nulovy, skoc este prijat CRC
(1) 593/931C : 0D          dcr     c          ; zniz pocet bytov v uR
(1) 594/931D : C2 06 93    jnz     RecvVramB      ; a opakuj pre cely uR
(1) 595/9320 :
(1) 596/9320 : 0E 10          mvi     c,16          ; 16 bytov sa cita "na prazdno"
(1) 597/9322 : CD D8 94    RecvVramE: call   RecvByte      ; prijmi byte
(1) 598/9325 : 80          add     b          ; iba modifikuj CRC
(1) 599/9326 : 47          mov     b,a        ; a vrat do B
(1) 600/9327 : 3A BF C1    lda     LanErr      ; bola chyba?
(1) 601/932A : A7          ana     a
(1) 602/932B : C2 13 93    jnz     RecvVramC      ; ak ano, ukonci prijem a vypis chybu
(1) 603/932E : 1B          dcx     d          ; zniz celkovy pocet prijimanych bytov
(1) 604/932F : 7B          mov     a,e
(1) 605/9330 : B2          ora     d
(1) 606/9331 : CA 42 93    jz      RecvVramF      ; ak je nulovy, skoc este prijat CRC
(1) 607/9334 : 0D          dcr     c          ; zniz pocitadlo prijimanych bytov
(1) 608/9335 : C2 22 93    jnz     RecvVramE      ; a opkauj pre vsetkych 16 bytov
(1) 609/9338 :            ; Nasledujuci kod s prechodom na dalsi uR by tu nemusel byt, keby sa pri prijme
(1) 610/9338 :            ; "zahadzovanych" bytov inkrementovala aj adresa v HL. Samozrejme, aj `push h`
(1) 611/9338 :            ; za navestim RecvVramA by uz nemalo byt.
(1) 612/9338 : E1          pop     h          ; obnov "riadkovu" adresu
(1) 613/9339 : D5          push    d          ; odpamataj pocitadlo
(1) 614/933A : 11 40 00    lxi     d,64          ; posun sa na dalsi "uR"
(1) 615/933D : 19          dad     d
(1) 616/933E : D1          pop     d          ; obnov pocitadlo
(1) 617/933F : C3 03 93    jmp     RecvVramA      ; a pokracuj v prijme
(1) 618/9342 :
(1) 619/9342 : E1          RecvVramF: pop     h          ; zahod adresu na zasobniku
(1) 620/9343 : CD D8 94    call   RecvByte      ; prijmi este CRC
(1) 621/9346 : B8          cmp     b          ; a porovnaj s vypocitanym
(1) 622/9347 : C8          rz      ; navrat, ak su zhodne
(1) 623/9348 : 3E 34      mvi     a,'4'         ; inak, nastav chybu `Check sum error`
(1) 624/934A : 32 BF C1    sta     LanErr
(1) 625/934D : C9          ret
(1) 626/934E :
(1) 627/934E :
(1) 628/934E :            ;-----
(1) 629/934E :            ; Prijem bloku dat typu M, D alebo B.
(1) 629/934E : 2A B6 C1    RecvOther: lhld   LanBlkLen      ; dlzka dat bloku
(1) 630/9351 : EB          xchg
(1) 631/9352 : 2A B8 C1    lhld   LanStoAdr      ; ukladacia adresa do HL
(1) 632/9355 : CD C3 94    call   RecvBlock      ; prijmi blok
(1) 633/9358 : 3A BF C1    lda     LanErr      ; bola chyba?
(1) 634/935B : A7          ana     a
(1) 635/935C : C0          rnz

```

```

(1) 636/935D : CD D8 94      call   RecvByte      ; prijmi este CRC
(1) 637/9360 : B8           cmp      b              ; a porovnaj s vypocitanym
(1) 638/9361 : CA 6A 93      jz       RecvOtherM    ; skoc, ak su zhodne
(1) 639/9364 : 3E 34         mvi      a,'4'         ; inak, nastav chybu `Check sum error`
(1) 640/9366 : 32 BF C1      sta      LanErr
(1) 641/9369 : C9           ret
(1) 642/936A :
(1) 643/936A : 3A B3 C1      RecvOtherM:  lda      LanBlkTyp    ; typ bloku
(1) 644/936D : FE 4D         cpi      'M'          ; je to blok M?
(1) 645/936F : C2 79 93      jnz      RecvOtherB    ; nie, skoc dalej
(1) 646/9372 : 31 00 80      lxi      sp,Stack      ; blok typu M sa aj spusti, nastav SP
(1) 647/9375 : 2A B8 C1      lhld     LanStoAdr     ; ukaladacia adresa je aj spustacia
(1) 648/9378 : E9           pchl
(1) 649/9379 :
(1) 650/9379 : FE 42      RecvOtherB:  cpi      'B'          ; je to blok B?
(1) 651/937B : C0           rnz
(1) 652/937C : 21 A9 94      lxi      h,T_LanRunBasic ; vypis "RUN BASIC PROGRAM"
(1) 653/937F : 22 74 C0      shld     MESS
(1) 654/9382 : CD 89 8A      call     PRTTEXT
(1) 655/9385 :
(1) 656/9385 :
(1) 657/9385 :
(1) 658/9385 :
(1) 659/9385 : 21 00 98      ; Tu je zasadna chyba, kedy sa z ROM presunie BASIC aj s DEMO programom, takze
(1) 660/9388 : 11 00 00      ; sa vzdy nacistany program v BASICu prepise!
(1) 661/938B : CD 12 90      ; Naviac sa podla prijateho programu nastavi systemova premenna konca programu,
(1) 662/938E : 2A B6 C1      ; takze ani ten DEMO program nebude fungovat spravne.
(1) 663/9391 : EB           lxi      h,Basic       ; adresa interpretera BASICu v ROM
(1) 664/9392 : 2A B8 C1      lxi      d,0           ; cielova adresa v RAM
(1) 665/9395 : 19           call     MoveROM       ; presun BASIC a aj DEMO program!
(1) 666/9396 : 22 7A 5E      lhld     LanBlkLen     ; dlzka programu v BASICu
(1) 667/9399 : CD 9F 93      xchg
(1) 668/939C : C3 00 00      ; do DE
(1) 669/939F :
(1) 670/939F :
(1) 671/939F :
(1) 672/939F : 11 00 80      lhld     LanStoAdr     ; ukladacia adresa do HL
(1) 673/93A2 : C3 B5 88      dad      d            ; pripocitaj
(1) 674/93A5 :
(1) 675/93A5 :
(1) 676/93A5 :
(1) 677/93A5 :
(1) 678/93A5 :
(1) 679/93A5 :
(1) 680/93A5 : 06 00      shld     BasPrgEnd     ; a uloz ako koncovy adresu programu
(1) 681/93A7 : 7E           call     Delay290      ; male zdrzanie
(1) 682/93A8 : 80           jmp      0             ; a spusti BASIC

;-----
; Zdrzanie asi 290 ms.
Delay290:  lxi      d,32768
           jmp      DelayDE

;-----
; Odoslanie bloku.
; I: HL=pociatocna adresa, DE=dlzka bloku
; O: [LanErr]=chybovy kod
; M: vsetky
SendBlock: mvi      b,0           ; vynuluj CRC
SendBlockL: mov     a,m          ; posielany byte do A
           add     b            ; modifikuj CRC

```



```

(1) 683/93A9 : 47      mov     b,a           ; a vrat do B
(1) 684/93AA : 4E      mov     c,m           ; posielany byte do C
(1) 685/93AB : CD BF 93 call    SendByte       ; odosli
(1) 686/93AE : 3A BF C1 lda     LanErr       ; bolo odoslanie v poriadku?
(1) 687/93B1 : A7      ana     a
(1) 688/93B2 : C0      rnz             ; nie, navrat
(1) 689/93B3 : 23      inx     h           ; posun na dalsiu adresu
(1) 690/93B4 : 1B      dcx     d           ; zniz pocitadlo bytov
(1) 691/93B5 : 7A      mov     a,d
(1) 692/93B6 : B3      ora     e
(1) 693/93B7 : C2 A7 93 jnz     SendBlockL     ; a opakuj pre cely blok
(1) 694/93BA : 48      mov     c,b           ; odosli este CRC
(1) 695/93BB : CD BF 93 call    SendByte
(1) 696/93BE : C9      ret
(1) 697/93BF :
(1) 698/93BF :
(1) 699/93BF : ;-----
(1) 700/93BF : ; Odoslanie bytu.
(1) 701/93BF : ;
(1) 702/93BF : ;         init  S    0  1  2  3  4  5  6  7    P
(1) 703/93BF : ;   OUT
(1) 704/93BF : ; odosielatel  ____|____|____|____|____|____|____|____|____|____|____|____|
(1) 705/93BF : ;
(1) 706/93BF : ;         DCD
(1) 707/93BF : ;
(1) 708/93BF : ;   IN
(1) 709/93BF : ; prijemca   ____|_____|_____|_____|_____|_____|_____|_____|_____|____|____|
(1) 710/93BF : ;
(1) 711/93BF : ; I: C=odosielany byte
(1) 712/93C0 : ; O: [LanErr]=stav siete
(1) 713/93C3 : ; M: AF
(1) 714/93C3 : D5      xra     a           ; vynuluj chybu
(1) 715/93C4 : C5      sta     LanErr
(1) 716/93C5 :
(1) 717/93C5 : 3E 01      push    d           ; odpamataj DE, BC
(1) 718/93C7 : D3 F6      push    b
(1) 719/93C9 : 11 00 90 mvi     a,X_MGF_OUT    ; nastav Log.1, ako vyzvu prijemcu
(1) 720/93CC : DB F6      out     SYS55C
(1) 721/93CE : 17      lxi     d,C_HS_TO    ; timeout ~1s
(1) 722/93CF : DA E0 93 SendByteA: in     SYS55C    ; precitaj stav portu C
(1) 723/93D2 : 1B      ral             ; X_MGF_IN do CY
(1) 724/93D3 : 7A      jc     SendByteB    ; skoc, ak prijemca potvrdil vyzvu
(1) 725/93D4 : B3      dcx     d           ; zniz pocitadlo timeoutu
(1) 726/93D5 : C2 CC 93 mov     a,d           ; a kym nie je nulove,
(1) 727/93D8 : 3E 32      ora     e
(1) 728/93DA : 32 BF C1 jnz     SendByteA     ; cakaj dalej na potvrdenie
(1) 729/93DD : C3 18 94 mvi     a,'2'         ; inak, nastav chybu `No ready for data`
                        sta     LanErr
                        jmp     SendByteE    ; a ukonci rutinu

```

```

(1) 730/93E0 :
(1) 731/93E0 :
(1) 732/93E0 : AF
(1) 733/93E1 : D3 F6
(1) 734/93E3 : CD 6F 95
(1) 735/93E6 :
(1) 736/93E6 : 06 08
(1) 737/93E8 : 79
(1) 738/93E9 : E6 01
(1) 739/93EB : D3 F6
(1) 740/93ED : 79
(1) 741/93EE : 1F
(1) 742/93EF : 4F
(1) 743/93F0 : CD 75 95
(1) 744/93F3 : 05
(1) 745/93F4 : C2 E8 93
(1) 746/93F7 :
(1) 747/93F7 : 3E 01
(1) 748/93F9 : D3 F6
(1) 749/93FB : CD 75 95
(1) 750/93FE : CD 75 95
(1) 751/9401 :
(1) 752/9401 : AF
(1) 753/9402 : D3 F6
(1) 754/9404 : 11 00 90
(1) 755/9407 : DB F6
(1) 756/9409 : 17
(1) 757/940A : D2 18 94
(1) 758/940D : 1B
(1) 759/940E : 7A
(1) 760/940F : B3
(1) 761/9410 : C2 07 94
(1) 762/9413 : 3E 31
(1) 763/9415 : 32 BF C1
(1) 764/9418 :
(1) 765/9418 : AF
(1) 766/9419 : D3 F6
(1) 767/941B :
(1) 768/941B : C1
(1) 769/941C : D1
(1) 770/941D : C9
(1) 771/941E :
(1) 772/941E :
(1) 773/941E :
(1) 774/941E : (MACRO)
(1) 774/941E : 2B 2B 20 54 52 41 4E 53
9426 : 46 45 52 20 4F 4B 20 2B

; start-bit
SendByteB: xra a ; nastav Log.0 - start-bit
out SYS55C
call LanBitDly1 ; zdrzanie pre start bit ~154,3 us
; datove bity
SendByteC: mvi b,8 ; 8 datovych bitov
mov a,c ; odosielany byte do A
ani 1 ; odmaskuj X_MGF_OUT bit
out SYS55C ; a nastav na vystupe podla daneho bitu
mov a,c ; posielany byte
rar ; odrotuj doprava
mov c,a
call LanBitDly2 ; zdrzanie pre datovy bit ~305,6 us
dcr b ; opakuj 8x pre kazdy datovy bit
jnz SendByteC
; stop-bit
mvi a,X_MGF_OUT ; nastav Log.1, ako stop-bit
out SYS55C
call LanBitDly2 ; ma dvojnásobnu dĺžku oproti datovému
call LanBitDly2 ; bitu 2x ~305,6 us

xra a ; nastav Log.0 - koniec prenosu bytu
out SYS55C
SendByteD: lxi d,C_HS_TO ; timeout ~1s
in SYS55C ; precitaj stav portu C
ral ; X_MGF_IN do CY
jnc SendByteE ; skoc, ak prijemca potvrdil prij. bytu
dcx d ; zniz pocitadlo timeoutu
mov a,d ; a kym nie je nulove,
ora e
jnz SendByteD ; cakaj dalej na potvrdenie
mvi a,'1' ; inak, nastav chybu `Frame error`
sta LanErr

SendByteE: xra a ; nastav Log.0 - koniec prenosu bytu
out SYS55C
pop b ; obnov BC, DE
pop d
ret

;-----
; Texty vypisovane pri pouzivani sietovych prikazov READ, SEND a SET.
T_LanXferOK: dm "++ TRANSFER OK ++"
db "++ TRANSFER OK ++"

```

```

942E : 2B
(1) 774/942F : 0D                                db    CR
(1) 775/9430 : (MACRO)                          T_LanFrmErr: dm    "++ FRAME ERROR ++"
(1) 775/9430 : 2B 2B 20 46 52 41 4D 45          db    "++ FRAME ERROR ++"
          9438 : 20 45 52 52 4F 52 20 2B
          9440 : 2B
(1) 775/9441 : 0D                                db    CR
(1) 776/9442 : (MACRO)                          T_LanNRdyData: dm    "++ NO READY FOR DATA ++"
(1) 776/9442 : 2B 2B 20 4E 4F 20 52 45          db    "++ NO READY FOR DATA ++"
          944A : 41 44 59 20 46 4F 52 20
          9452 : 44 41 54 41 20 2B 2B
(1) 776/9459 : 0D                                db    CR
(1) 777/945A : (MACRO)                          T_LanNRdySend: dm    "++ NO READY FOR SEND ++"
(1) 777/945A : 2B 2B 20 4E 4F 20 52 45          db    "++ NO READY FOR SEND ++"
          9462 : 41 44 59 20 46 4F 52 20
          946A : 53 45 4E 44 20 2B 2B
(1) 777/9471 : 0D                                db    CR
(1) 778/9472 : (MACRO)                          T_LanCrcErr:  dm    "++ CHECK SUM ERROR ++"
(1) 778/9472 : 2B 2B 20 43 48 45 43 4B          db    "++ CHECK SUM ERROR ++"
          947A : 20 53 55 4D 20 45 52 52
          9482 : 4F 52 20 2B 2B
(1) 778/9487 : 0D                                db    CR
(1) 779/9488 : (MACRO)                          T_LanWaitPls:  dm    "++ LAN ACTIVED - WAIT PLEASE! ++"
(1) 779/9488 : 2B 2B 20 4C 41 4E 20 41          db    "++ LAN ACTIVED - WAIT PLEASE! ++"
          9490 : 43 54 49 56 45 44 20 2D
          9498 : 20 57 41 49 54 20 50 4C
          94A0 : 45 41 53 45 21 20 2B 2B
(1) 779/94A8 : 0D                                db    CR
(1) 780/94A9 : (MACRO)                          T_LanRunBasic: dm    "++ RUN BASIC PROGRAM ! ++"
(1) 780/94A9 : 2B 2B 20 52 55 4E 20 42          db    "++ RUN BASIC PROGRAM ! ++"
          94B1 : 41 53 49 43 20 50 52 4F
          94B9 : 47 52 41 4D 20 21 20 2B
          94C1 : 2B
(1) 780/94C2 : 0D                                db    CR
(1) 781/94C3 :
(1) 782/94C3 :
(1) 783/94C3 :
(1) 784/94C3 :
(1) 785/94C3 :
(1) 786/94C3 :
(1) 787/94C3 : 06 00          RecvBlock:    mvi    b,0          ; vynuluj CRC
(1) 788/94C5 : CD D8 94      RecvBlockL:    call   RecvByte       ; prijmi byte
(1) 789/94C8 : 77              mov    m,a          ; uloz byte
(1) 790/94C9 : 80              add    b          ; modifikuj CRC
(1) 791/94CA : 47              mov    b,a          ; a vrat do B
(1) 792/94CB : 3A BF C1        lda    LanErr       ; bol prijem v poriadku?
(1) 793/94CE : A7              ana    a

```

```

;-----
; Prijatie bloku.
; I: HL=pociatocna adresa, DE=dlzka bloku
; O: B=vypocitane CRC, [LanErr]=chybovy kod
; M: vsetky
RecvBlock:    mvi    b,0          ; vynuluj CRC
RecvBlockL:    call   RecvByte       ; prijmi byte
              mov    m,a          ; uloz byte
              add    b          ; modifikuj CRC
              mov    b,a          ; a vrat do B
              lda    LanErr       ; bol prijem v poriadku?
              ana    a

```

```

(1) 794/94CF : C0          rnz          ; nie, navrat
(1) 795/94D0 : 23          inx      h          ; posun na dalsiu adresu
(1) 796/94D1 : 1B          dcx      d          ; zniz pocitadlo bytov
(1) 797/94D2 : 7A          mov      a,d
(1) 798/94D3 : B3          ora      e
(1) 799/94D4 : C2 C5 94    jnz      RecvBlockL      ; a opakuj pre cely blok
(1) 800/94D7 : C9          ret
(1) 801/94D8 :
(1) 802/94D8 : ;-----
(1) 803/94D8 : ; Prijatie bytu.
(1) 804/94D8 :
(1) 805/94D8 : ;                      DCD                      ACK
(1) 806/94D8 : ; OUT
(1) 807/94D8 : ; prijemca _____|_____|_____
(1) 808/94D8 :
(1) 809/94D8 : ;                      init S  0  1  2  3  4  5  6  7  P
(1) 810/94D8 : ; IN
(1) 811/94D8 : ; odosielatel ____|____|____|____|____|____|____|____|____|____|____
(1) 812/94D8 :
(1) 813/94D8 : ; I: -
(1) 814/94D8 : ; O: A=prijaty byte, [LanErr]=stav siete
(1) 815/94D8 : ; M: AF
(1) 816/94D8 : AF          RecvByte:  xra      a          ; vynuluj chybu
(1) 817/94D9 : 32 BF C1    sta      LanErr
(1) 818/94DC :
(1) 819/94DC : D5          push     d          ; odpamataj DE, BC
(1) 820/94DD : C5          push     b
(1) 821/94DE :
(1) 822/94DE : ; cakanie na vyzvu od odosielatela
(1) 823/94DE : 11 00 90    lxi      d,C_HS_TO      ; timeout ~1s
(1) 824/94E1 : DB F6          RecvByteA: in      SYS55C          ; precitaj stav portu C
(1) 825/94E3 : 17          ral          ; X_MGF_IN do CY
(1) 826/94E4 : DA F5 94    jc      RecvByteB      ; skoc, ak odosielatel poslal vyzvu
(1) 827/94E7 : 1B          dcx      d          ; zniz pocitadlo timeoutu
(1) 828/94E8 : 7A          mov      a,d          ; a kym nie je nulove,
(1) 829/94E9 : B3          ora      e
(1) 830/94EA : C2 E1 94    jnz      RecvByteA      ; cakaj dalej na vyzvu
(1) 831/94ED : 3E 33          mvi      a,'3'          ; inak, nastav chybu `No ready for send`
(1) 832/94EF : 32 BF C1    sta      LanErr
(1) 833/94F2 : C3 33 95    jmp      RecvByteH      ; a ukonci rutinu
(1) 834/94F5 :
(1) 835/94F5 : 3E 01          RecvByteB: mvi      a,X_MGF_OUT      ; nastav Log.1, ako potvrdenie vyzvy
(1) 836/94F7 : D3 F6          out      SYS55C
(1) 837/94F9 : ; cakanie na start-bit
(1) 838/94F9 : 11 00 90    lxi      d,C_HS_TO      ; timeout ~1s
(1) 839/94FC : DB F6          RecvByteC: in      SYS55C          ; precitaj stav portu C
(1) 840/94FE : 17          ral          ; X_MGF_IN do CY

```

```

(1) 841/94FF : D2 0B 95      jnc      RecvByteD      ; skoc, ak prisiel start-bit
(1) 842/9502 : 1B            dcx      d      ; zniz pocitadlo timeoutu
(1) 843/9503 : 7A            mov      a,d      ; a kym nie je nulove,
(1) 844/9504 : B3            ora      e
(1) 845/9505 : C2 FC 94      jnz      RecvByteC      ; cakaj dalej na start-bit
(1) 846/9508 : C3 2E 95      jmp      RecvByteG      ; inak, nastav chybu `Frame error`
(1) 847/950B :
(1) 848/950B :                ; prijem datovych bitov
(1) 849/950B : 0E 08      RecvByteD: mvi      c,8      ; 8 datovych bitov
(1) 850/950D : 06 00      mvi      b,0      ; do B sa vysklada prijimany byte
(1) 851/950F : CD 7C 95      RecvByteE: call    LanRecvDly ; zdrzanie pri prijme bitu ~297,9 us
(1) 852/9512 : DB F6      in      SYS55C      ; precitaj stav portu C
(1) 853/9514 : 17          ral          ; X_MGF_IN do CY
(1) 854/9515 : 78          mov      a,b      ; vyskladavany byte
(1) 855/9516 : 1F          rar          ; vsun bit do vysledneho bytu
(1) 856/9517 : 47          mov      b,a      ; a opat uloz do B
(1) 857/9518 : 0D          dcr      c      ; opakuj 8x pre kazdy datovy bit
(1) 858/9519 : C2 0F 95      jnz      RecvByteE
(1) 859/951C : CD 7C 95      call    LanRecvDly      ; zdrzanie po prijati posledneho bitu
(1) 860/951F :                ; ~297,9 us
(1) 861/951F :
(1) 862/951F :                ; stop-bity sa "ignoruju" a caka sa na ukoncenie prenosu
(1) 863/951F : 11 00 90      RecvByteF: lxi      d,C_HS_T0      ; timeout ~1s
(1) 864/9522 : DB F6      in      SYS55C      ; precitaj stav portu C
(1) 865/9524 : 17          ral          ; X_MGF_IN do CY
(1) 866/9525 : D2 33 95      jnc      RecvByteH      ; skoc, ak odosielatel ukonci prenos bytu
(1) 867/9528 : 1B          dcx      d      ; zniz pocitadlo timeoutu
(1) 868/9529 : 7A          mov      a,d      ; a kym nie je nulove,
(1) 869/952A : B3          ora      e
(1) 870/952B : C2 22 95      jnz      RecvByteF      ; cakaj dalej na ukoncenie prenosu
(1) 871/952E :                ; inak, nastav chybu `Frame error`
(1) 872/952E : 3E 31      RecvByteG: mvi      a,'1'      ; Frame error
(1) 873/9530 : 32 BF C1      sta      LanErr
(1) 874/9533 :
(1) 875/9533 : AF          RecvByteH: xra      a      ; Log.0 - potvrd prijem bytu
(1) 876/9534 : D3 F6      out      SYS55C
(1) 877/9536 :
(1) 878/9536 : 78          mov      a,b      ; A=prijaty byte
(1) 879/9537 :
(1) 880/9537 : C1          pop      b      ; obnov BC, DE
(1) 881/9538 : D1          pop      d
(1) 882/9539 : C9          ret
(1) 883/953A :
(1) 884/953A :                ; -----
(1) 885/953A :                ; Pripravi adresu hlasenia podla kodu chyby.
(1) 886/953A :                ; I: [LanErr]=kod chyby
(1) 887/953A :                ; O: [MESS]=adresa hlasenia

```

```

(1) 888/953A : ; M: AF, HL
(1) 889/953A : 3A BF C1 PrepErrMsg: lda LanErr ; kod chyby
(1) 890/953D : A7 ana a ; podla kodu priprav adresu hlasenia
(1) 891/953E : CA 57 95 jz PrepErrMsg0
(1) 892/9541 : FE 31 cpi '1'
(1) 893/9543 : CA 5D 95 jz PrepErrMsg1
(1) 894/9546 : FE 32 cpi '2'
(1) 895/9548 : CA 63 95 jz PrepErrMsg2
(1) 896/954B : FE 33 cpi '3'
(1) 897/954D : CA 69 95 jz PrepErrMsg3
(1) 898/9550 :
(1) 899/9550 : 21 72 94 lxi h,T_LanCrcErr
(1) 900/9553 : 22 74 C0 PrepErrMsgM: shld MESS
(1) 901/9556 : C9 ret
(1) 902/9557 :
(1) 903/9557 : 21 1E 94 PrepErrMsg0: lxi h,T_LanXferOK
(1) 904/955A : C3 53 95 jmp PrepErrMsgM
(1) 905/955D :
(1) 906/955D : 21 30 94 PrepErrMsg1: lxi h,T_LanFrmErr
(1) 907/9560 : C3 53 95 jmp PrepErrMsgM
(1) 908/9563 :
(1) 909/9563 : 21 42 94 PrepErrMsg2: lxi h,T_LanNRdyData
(1) 910/9566 : C3 53 95 jmp PrepErrMsgM
(1) 911/9569 :
(1) 912/9569 : 21 5A 94 PrepErrMsg3: lxi h,T_LanNRdySend
(1) 913/956C : C3 53 95 jmp PrepErrMsgM
(1) 914/956F :
(1) 915/956F : ;-----
(1) 916/956F : ; Rutiny zdrzania pri odosielani a prijimani bitu.
(1) 917/956F : 3A FE C1 LanBitDly1: lda LANTM1 ; zdrzanie pre start-bit : ~154,3 us
(1) 918/9572 : C3 7F 95 jmp LanBitDly
(1) 919/9575 :
(1) 920/9575 : 3A FE C1 LanBitDly2: lda LANTM1 ; zdrzanie pre datove bity : ~305,6 us
(1) 921/9578 : 17 ral ; je dvojnásobne
(1) 922/9579 : C3 7F 95 jmp LanBitDly
(1) 923/957C :
(1) 924/957C : 3A FF C1 LanRecvDly: lda LANTM2 ; zdrzanie pri prijme bitu : ~297,9 us
(1) 925/957F : 3D LanBitDly: dcr a
(1) 926/9580 : C2 7F 95 jnz LanBitDly
(1) 927/9583 : C9 ret
(1) 928/9584 :
(1) 929/9584 : ;-----
(1) 930/9584 : ; Radoby rutina na preskocenie medzier pri spracovani vstupu z riadku.
(1) 931/9584 : ; K posunu adresy nikdy nedojde, pretoze sa adresa vzdy nanovo nacita zo syst.
(1) 932/9584 : ; premennej CURCH.
(1) 933/9584 : ; I: [CURCH]=ukazatel do vstupneho riadku
(1) 934/9584 : ; O: [CURCH]=ukazatel do vstupneho riadku

```

(1) 935/9584 : 2A 72 C0	SkipSpcX:	lhld	CURCH	; aktualna adresa buffra vstupneho textu
(1) 936/9587 : 7E		mov	a,m	; vezmi byte
(1) 937/9588 : FE 20		cpi	' '	; je to medzera
(1) 938/958A : 22 72 C0		shld	CURCH	; uloz "novu" adresu
(1) 939/958D : C0		rnz		; navrat, ak to nie je medzera
(1) 940/958E : 23		inx	h	; posun ukazatel
(1) 941/958F : C3 84 95		jmp	SkipSpcX	; tu mal byt skok na `mov a,m`
(1) 942/9592 :				
(1) 943/9592 :				
(1) 944/9592 :				; Inkrementacia ukazatela na zadany text.
(1) 945/9592 :				; I: [CURCH]=ukazatel do vstupneho riadku
(1) 946/9592 :				; O: [CURCH]=ukazatel do vstupneho riadku
(1) 947/9592 : 2A 72 C0	IncPtr:	lhld	CURCH	; aktualna adresa
(1) 948/9595 : 23		inx	h	; +1
(1) 949/9596 : 22 72 C0		shld	CURCH	; uloz novu
(1) 950/9599 : C9		ret		
(1) 951/959A :				
(1) 952/959A :				
(1) 953/959A :				; Rutina overi, ci je LAN aktivovana.
(1) 954/959A :				; I: [LanFlag]=stav aktivity siete
(1) 955/959A :				; O: Z=0 - LAN aktivovana
(1) 956/959A :				; Z=1 - LAN neaktivovana, pripravene hlasenie "LAN OFF"
(1) 957/959A :				; M: AF, HL
(1) 958/959A : 3A 7D C0	IsLanOn:	lda	LanFlag	; je LAN aktivovana?
(1) 959/959D : A7		ana	a	
(1) 960/959E : C0		rnz		; ano, navrat s Z=0
(1) 961/959F : 21 D6 95		lxi	h,T_LanOff	; priprav hlasenie "LAN OFF"
(1) 962/95A2 : 22 74 C0		shld	MESS	
(1) 963/95A5 : C9		ret		
(1) 964/95A6 :				
(1) 965/95A6 :				
(1) 966/95A6 :				;*****
(1) 967/95A6 :				
(1) 968/95A6 :				
(1) 969/95A6 :				; Obsluha prikazu SET - de/aktivacia LAN a zobrazenie pomocnika
(1) 970/95A6 :				; Format: SET AD - aktivacia siete s adresou uzivatela AD
(1) 971/95A6 :				; SET - deaktivacia LAN
(1) 972/95A6 : CD 84 95				; SET / - zobrazenie pomocnika
(1) 973/95A9 : FE 2F	SET:	call	SkipSpcX	; preskoc medzery na vstupe
(1) 974/95AB : CA E4 95		cpi	'/'	; je to '/'
(1) 975/95AE : CD 60 91		jz	PrtHelp	
(1) 976/95B1 : DA C2 95		call	GetIdR	; precitaj cislo pre tuto stanicu
(1) 977/95B4 : 32 BE C1		jc	SetPrtLanOff	; skoc, chybne cislo
(1) 978/95B7 : 3E FF		sta	LanID	; uloz cislo pre tuto stanicu
(1) 979/95B9 : 21 C9 95	SetPrtLanOn:	mvi	a,0FFh	; priznak - siet je aktivovana
(1) 980/95BC : 32 7D C0		lxi	h,T_LanOn	; text "LAN ON"
(1) 981/95BF : C3 6F 92	SetPrtLan:	sta	LanFlag	; uloz priznak stavu siete
		jmp	PrtMessage	; vypis spravu

```

(1) 982/95C2 :
(1) 983/95C2 : 21 D6 95          SetPrtLanOff: lxi    h,T_LanOff    ; text "LAN OFF"
(1) 984/95C5 : AF                xra    a                ; priznak - sie nie je aktivovana
(1) 985/95C6 : C3 BC 95          jmp    SetPrtLan        ; uloz priznak a vypis spravu
(1) 986/95C9 :
(1) 987/95C9 :
(1) 988/95C9 :                  ;-----
(1) 988/95C9 :                  ; Texty stavu siete.
(1) 989/95C9 : (MACRO)          T_LanOn:    dm    "++ LAN=ON ++"
(1) 989/95C9 : 2B 2B 20 4C 41 4E 3D 4F      db    "++ LAN=ON ++"
          95D1 : 4E 20 2B 2B
(1) 989/95D5 : 0D                db    CR
(1) 990/95D6 : (MACRO)          T_LanOff:   dm    "++ LAN=OFF ++"
(1) 990/95D6 : 2B 2B 20 4C 41 4E 3D 4F      db    "++ LAN=OFF ++"
          95DE : 46 46 20 2B 2B
(1) 990/95E3 : 0D                db    CR
(1) 991/95E4 :
(1) 992/95E4 :                  ;-----
(1) 993/95E4 :                  ; Zobrazenie pomocnika.
(1) 994/95E4 : CD A7 85          PrtHelp:    call   ERASE            ; zmaz obrazovku
(1) 995/95E7 : 21 00 96          lxi    h,T_LanHelp    ; adresa textu
(1) 996/95EA : 7E                PrtHelpL:   mov    a,m            ; vezmi znak
(1) 997/95EB : A7                ana    a                ; je to koniec textu?
(1) 998/95EC : CA F6 95          jz     PrtHelpE
(1) 999/95EF : CD 00 85          call   PRTOUT          ; vypis znak
(1) 1000/95F2 : 23               inx    h                ; posun ukazatel
(1) 1001/95F3 : C3 EA 95          jmp    PrtHelpL
(1) 1002/95F6 :
(1) 1003/95F6 : 3A 7D C0          PrtHelpE:   lda    LanFlag        ; je LAN aktivovana?
(1) 1004/95F9 : A7                ana    a
(1) 1005/95FA : CA C2 95          jz     SetPrtLanOff    ; nie, skoc vypisat
(1) 1006/95FD : C3 B7 95          jmp    SetPrtLanOn
(1) 1007/9600 :
(1) 1008/9600 :                  ;-----
(1) 1009/9600 :                  ; Text pomocnika po zadani prikazu SET/
(1) 1010/9600 : (MACRO)          T_LanHelp:   dm    "***** LAN HELP *****"
(1) 1010/9600 : 2A 2A 2A 2A 2A 2A 2A 2A      db    "***** LAN HELP *****"
          9608 : 2A 2A 2A 2A 2A 2A 2A 2A
          9610 : 2A 20 20 4C 41 4E 20 48
          9618 : 45 4C 50 20 20 20 2A 2A
          9620 : 2A 2A 2A 2A 2A 2A 2A 2A
          9628 : 2A 2A 2A 2A 2A 2A
(1) 1010/962E : 0D                db    CR
(1) 1011/962F : 0D 0D            db    CR,CR
(1) 1012/9631 : (MACRO)          dm    "SET AD                AD=2xHEX AKTIVUJE LAN stn."
(1) 1012/9631 : 53 45 54 20 41 44 20 20      db    "SET AD                AD=2xHEX AKTIVUJE LAN stn."
          9639 : 20 20 20 20 20 20 20 20
          9641 : 20 20 20 20 41 44 3D 32

```



```

9649 : 78 48 45 58 20 41 4B 54
9651 : 49 56 55 4A 45 20 4C 41
9659 : 4E 20 73 74 6E 2E
(1) 1012/965F : 0D
(1) 1013/9660 : (MACRO)
(1) 1013/9660 : 20 20 20 20 20 20 20 20
9668 : 20 20 20 20 20 20 20 20
9670 : 20 20 20 20 41 44 3D 78
9678 : 78 20 44 45 41 4B 54 49
9680 : 56 55 4A 45 20 4C 41 4E
9688 : 20 73 74 6E 2E
(1) 1013/968D : 0D
(1) 1014/968E : (MACRO)
(1) 1014/968E : 53 45 54 2F 20 20 20 20
9696 : 20 20 20 20 20 20 20 20
969E : 20 20 20 20 48 45 4C 50
(1) 1014/96A6 : 0D
(1) 1015/96A7 : 0D
(1) 1016/96A8 : (MACRO)
(1) 1016/96A8 : 56 59 53 4C 41 4E 49 45
96B0 : 20 42 4C 4F 4B 55 20 41
96B8 : 44 52 45 53 41 54 4F 56
96C0 : 49 20 28 41 44 29 3A
(1) 1016/96C7 : 0D
(1) 1017/96C8 : 0D
(1) 1018/96C9 : (MACRO)
(1) 1018/96C9 : 53 45 4E 44 20 41 44 2F
96D1 : 58 2E 2E 2E 74 65 78 74
96D9 : 20 20 20 20 58 20 3D 20
96E1 : 47 2C 42 2C 54
(1) 1018/96E6 : 0D
(1) 1019/96E7 : (MACRO)
(1) 1019/96E7 : 53 45 4E 44 20 41 44 3B
96EF : 4F 44 2D 44 4F 2F 58 20
96F7 : 20 20 20 20 58 20 3D 20
96FF : 47 2C 54 2C 44 2C 4D
(1) 1019/9706 : 0D
(1) 1020/9707 : (MACRO)
(1) 1020/9707 : 53 45 4E 44 20 41 44 3B
970F : 4F 44 2D 44 4F 3B 54 4F
9717 : 2F 58 20 20 58 20 3D 20
971F : 47 2C 4D 2C 44
(1) 1020/9724 : 0D
(1) 1021/9725 : 0D 0D
(1) 1022/9727 : (MACRO)
(1) 1022/9727 : 56 59 5A 49 41 44 41 4E
972F : 49 45 20 42 4C 4F 4B 55

```

```

db CR
dm " AD=xx DEAKTIVUJE LAN stn."
db " AD=xx DEAKTIVUJE LAN stn."

db CR
dm "SET/ HELP"
db "SET/ HELP"

db CR
db CR
dm "VYSLANIE BLOKU ADRESATOVI (AD):"
db "VYSLANIE BLOKU ADRESATOVI (AD):"

db CR
db CR
dm "SEND AD/X...text X = G,B,T"
db "SEND AD/X...text X = G,B,T"

db CR
dm "SEND AD;OD-DO/X X = G,T,D,M"
db "SEND AD;OD-DO/X X = G,T,D,M"

db CR
dm "SEND AD;OD-DO;TO/X X = G,M,D"
db "SEND AD;OD-DO;TO/X X = G,M,D"

db CR
db CR,CR
dm "VYZIADANIE BLOKU OD ADRESATA (AD):"
db "VYZIADANIE BLOKU OD ADRESATA (AD):"

```

```

9737 : 20 4F 44 20 41 44 52 45
973F : 53 41 54 41 20 28 41 44
9747 : 29 3A
(1) 1022/9749 : 0D
(1) 1023/974A : 0D
(1) 1024/974B : (MACRO)
(1) 1024/974B : 52 45 41 44 20 41 44 2F
9753 : 58 20 20 20 20 20 20 20
975B : 20 20 20 20 58 20 3D 20
9763 : 47 2C 42
(1) 1024/9766 : 0D
(1) 1025/9767 : (MACRO)
(1) 1025/9767 : 52 45 41 44 20 41 44 3B
976F : 4F 44 2D 44 4F 2F 58 20
9777 : 20 20 20 20 58 20 3D 20
977F : 47 2C 44 2C 54
(1) 1025/9784 : 0D
(1) 1026/9785 : (MACRO)
(1) 1026/9785 : 52 45 41 44 20 41 44 3B
978D : 4F 44 2D 44 4F 3B 54 4F
9795 : 2F 58 20 20 58 20 3D 20
979D : 47 2C 4D 2C 44
(1) 1026/97A2 : 0D
(1) 1027/97A3 : 0D
(1) 1028/97A4 : (MACRO)
(1) 1028/97A4 : 54 20 3D 20 54 45 58 54
(1) 1028/97AC : 0D
(1) 1029/97AD : (MACRO)
(1) 1029/97AD : 42 20 3D 20 42 41 53 49
97B5 : 43 20 50 52 4F 47 52 41
97BD : 4D
(1) 1029/97BE : 0D
(1) 1030/97BF : (MACRO)
(1) 1030/97BF : 47 20 3D 20 47 52 41 46
97C7 : 49 4B 41
(1) 1030/97CA : 0D
(1) 1031/97CB : (MACRO)
(1) 1031/97CB : 4D 20 3D 20 50 52 4F 47
97D3 : 52 41 4D 20 56 20 53 54
97DB : 52 4F 4A 2E 20 4B 4F 44
97E3 : 45
(1) 1031/97E4 : 0D
(1) 1032/97E5 : (MACRO)
(1) 1032/97E5 : 44 20 3D 20 44 41 54 41
(1) 1032/97ED : 0D
(1) 1033/97EE : 00 00
(1) 1034/97F0 :

```

```

db CR
db CR
dm "READ AD/X          X = G,B"
db "READ AD/X          X = G,B"

db CR
dm "READ AD;OD-DO/X      X = G,D,T"
db "READ AD;OD-DO/X      X = G,D,T"

db CR
dm "READ AD;OD-DO;TO/X   X = G,M,D"
db "READ AD;OD-DO;TO/X   X = G,M,D"

db CR
db CR
dm "T = TEXT"
db "T = TEXT"
db CR
dm "B = BASIC PROGRAM"
db "B = BASIC PROGRAM"

db CR
dm "G = GRAFIKA"
db "G = GRAFIKA"

db CR
dm "M = PROGRAM V STROJ. KODE"
db "M = PROGRAM V STROJ. KODE"

db CR
dm "D = DATA"
db "D = DATA"
db CR
db 0,0

```

```

(1) 1035/97F0 : ;-----
(1) 1036/97F0 :
(1) 1037/97F0 : FF FF FF FF FF FF FF FF db (Basic-$) dup (0FFh)
      97F8 : FF FF FF FF FF FF FF FF
(1) 1038/9800 :
(1) 1039/9800 : ;-----
(1) 1040/9800 :
(1) 1041/9800 :
      2709/9800 :
      2710/9800 : ;-----
      2711/9800 :
      2715/9800 : bininclude "basic.bin"
      2717/BC00 : bininclude "demo.bin"
      2718/C000 :
      2719/C000 : ;-----
      2720/C000 :
      2721/C000 : end

```

▲ AS V1.42 Beta [Bld 269] - Source File mato-monitor.a8080 - Page 2 - 12/18/2025 20:28:58  
 Monitor pocitaca MATO

Symbol Table (\* = unused):

-----			
ADRAS :	83CE C	ADRASM :	83E4 C
ADRASU :	83EE C	ADRFIL :	0C1B4 -
ADRIN :	8109 C	ADRR1T :	8577 C
ADRRET :	8575 C		
*ARCHITECTURE :		"i386-unknown-win32" -	
ASCII :	0C134 -	A_Font_1 :	8600 C
A_Font_2 :	88BE C	BASIC :	8F11 C
*BCUR :	0C17A -	BEEDT :	0C130 -
BEEP :	88A3 C	BEEPPOS :	0C03B -
BELL :	88A6 C	BINBCD :	8E73 C
BINBCDL :	8E77 C	BS :	8 -
BUF :	0C030 -	BasPrgAdr :	2400 -
BasPrgEnd :	5E7A -	Basic :	9800 -
BasicEndHI :	0BC -	BasicL :	8F17 C
BasicLen :	2400 -	Begin :	8000 -
BitMasks :	80D7 C	*CASESENSITIVE :	1 -
CD :	889C C	CDBUF :	0C13A -
CHAR :	8194 C	CLR :	8113 C
CLS :	1C -	COLOR :	0C03A -
*CONSTPI :	3.141592653589793 -	CR :	0D -
CURCH :	0C072 -	CURPOS :	815E C
CURPOSHL :	8161 C	CURROL :	0C0F8 -
CURSOR :	0C03E -	C_HS_TO :	9000 -

CmdTab :	82EC C	*DATE :	"12/18/2025" -
DCUR :	0C036 -	DCUROLD :	0C175 -
DECP1 :	8E72 C	DECPAR :	8E3E C
DEL :	8D4D C	*DELKA :	0C177 -
DEMO :	9000 C	DIAL :	0C032 -
DLZP :	0C0FF -	DLZR :	0C0FE -
DLZRAD :	0C034 -	DOKEY :	8C43 C
DUMP :	81E6 C	Delay290 :	939F C
*DelayA :	8C1B C	DelayB :	8C1D C
DelayDE :	88B5 C	DeltM :	8BE0 C
DeltS :	8BE9 C	DemoEndHI :	0C0 -
DemoLen :	400 -	DemoPrgEnd :	27FD -
DoKeyS :	8C5C C	DumpC :	8210 C
DumpH :	81F8 C	DumpL :	81EA C
DumpS :	8206 C	DumpW :	821B C
ENDL :	8B90 C	ENDLN :	8B46 C
ENLNW :	0C0FB -	ENTER :	8BEE C
ENTER1 :	8B5B C	ENTER2 :	8B62 C
EOL :	8BB6 C	ERASE :	85A7 C
ERROR :	8B32 C	EndLnE :	8B53 C
EndLnS :	8B49 C	EndlA :	8B9B C
EndlB :	8BB0 C	EoLL :	8BC6 C
EraLines :	85B3 C	EraLinesL :	85C2 C
ExpBlkGMD :	90C5 C	ExpBlkGMD0 :	90D5 C
*FALSE :	0 -	FINDNR :	0C1B0 -
FINDTP :	0C1B1 -	FndCmdK :	806C C
FndCmdL :	806F C	FndCmdM :	807E C
GetAdr :	9186 C	GetBlkAdr :	916E C
GetBlkLen :	9176 C	GetIdR :	9160 C
GetStoAdr :	917E C	GoReadBlk :	91FA C
GoSendBlk :	907C C	*HAS64 :	0 -
HEAD :	0C1B2 -	HEX1 :	80DF C
HEX1D :	80F1 C	HOME :	85F6 C
IIROLL :	0C0F4 -	INIARA :	8C6B C
INITAB :	8231 C	INKLAV :	84A1 C
*INPOL :	8CD0 C	INS :	880D C
ITABDIR :	0C07E -	InKlavB :	8C35 C
InKlavN :	8C27 C	InKlavR :	84AA C
InKlavS :	8C25 C	IncPtr :	9592 C
InpolDX :	8CE6 C	InpolDY :	8CF4 C
InpolK :	8D15 C	InpolL :	8D1D C
InpolM :	8D20 C	InpolN :	8D2E C
InpolO :	8D35 C	InpolP :	8D42 C
InsL :	881A C	IsLanOn :	959A C
JOB :	8FD7 C	JUMP :	81D6 C
JmpIndHL :	8C66 C	KBEEP :	10 -
KBGNL :	0C -	KCD :	1E -

KCLL :	1B -	KCLS :	1F -
KDELL :	1 -	KDELT :	1D -
KDIR :	0C132 -	KDOWN :	13 -
KENDL :	3 -	KEOL :	0D -
KEY :	8827 C	KEYEND :	0C17C -
KEYMAP :	83F2 C	KEYOUT :	8800 C
KEYTOP :	0C17E -	KINST :	1C -
KINVR :	0A -	KK0 :	0D0 -
KK1 :	0D1 -	KK10 :	0DA -
KK11 :	0DB -	KK2 :	0D2 -
KK3 :	0D3 -	KK4 :	0D4 -
KK5 :	0D5 -	KK6 :	0D6 -
KK7 :	0D7 -	KK8 :	0D8 -
KK9 :	0D9 -	KLEFT :	8 -
KMON :	12 -	KPTL :	0F -
KRCL :	17 -	KRIGHT :	18 -
KSDOWN :	14 -	KSG0 :	80 -
KSG1 :	81 -	KSG10 :	8A -
KSG11 :	8B -	KSG12 :	8C -
KSG13 :	8D -	KSG14 :	8E -
KSG15 :	8F -	KSG16 :	90 -
KSG17 :	91 -	KSG18 :	92 -
KSG19 :	93 -	KSG2 :	82 -
KSG20 :	94 -	KSG21 :	95 -
KSG3 :	83 -	KSG4 :	84 -
KSG5 :	85 -	KSG6 :	86 -
KSG7 :	87 -	KSG8 :	88 -
KSG9 :	89 -	KSLEFT :	1A -
KSRIGHT :	19 -	KSUP :	9 -
KUP :	11 -	KWRK :	0B -
KbdTab :	82B2 C	*KeyLast :	0C0BC -
KeyRepeat :	0C0BD -	LANTM1 :	0C1FE -
LANTM2 :	0C1FF -	LEFT :	81B3 C
LENFIL :	0C1B6 -	LF :	0A -
*LISTON :	3 -	LLEFT :	81A9 C
LSTR :	0C13C -	LanBitDly :	957F C
LanBitDly1 :	956F C	LanBitDly2 :	9575 C
LanBlkAdr :	0C1B4 -	LanBlkLen :	0C1B6 -
LanBlkTyp :	0C1B3 -	LanDir :	0C1B2 -
LanErr :	0C1BF -	LanFlag :	0C07D -
LanID :	0C1BE -	LanIdR :	0C1B1 -
LanIdS :	0C1B0 -	LanRecvDly :	957C C
LanSP :	0C1BC -	LanStoAdr :	0C1B8 -
LoadBlock :	8DC2 C	LoadBlockC :	8DC4 C
LoadBlockF :	8F81 C	LoadBlockL :	8DC7 C
LoadBlockS :	8DD6 C	LoadBody :	8EED C
LoadByte :	8EFA C	LoadByteL :	8EFF C

LoadHead :	8EE4 C	*MACEXP :	7 -
MEM :	8587 C	MESS :	0C074 -
MGCHAR :	0C136 -	MGEND :	8879 C
MGLD :	8873 C	MGSV :	8E7F C
MODBEP :	8167 C	*MODIFY :	0C1FA -
MODNEG :	859F C	*MOMCPU :	8080 -
*MOMCPUNAME :	"8080" -	MON :	8006 C
MON1 :	803B C	MONIL1 :	8017 C
MONIL2 :	8014 C	MONLL :	800D C
MONX :	0C138 -	MVER :	1 -
M_Beep :	80D2 C	M_Key :	80F4 C
M_Velcome :	8189 C	MemL :	858F C
MgLdA :	8A6E C	MgLdMsg :	8A76 C
MgLdR :	833F C	MgLdT :	887A C
*Monit :	8C40 C	MoveROM :	9012 C
NAMFIL :	0C1B8 -	*NESTMAX :	100 -
NUMFIL :	0C1B2 -	OCHR :	0C1F1 -
OCHRA :	8B1C C	OCHRA :	8B1F C
ODLOZ :	0C078 -	*OLDKEY :	0C178 -
OnlyRet :	81B2 C	PAIRIN :	80F7 C
PLOTCOL :	0C1F8 -	*POINT :	8C7D C
POROLL :	0C0F6 -	POSPPOINT :	8C94 C
PRBTXT :	8B31 C	PRBWTXT :	8049 C
PRE00 :	886B C	PREV21 :	8137 C
PREV01 :	8125 C	PREV02 :	813B C
PRGOUT :	8F60 C	PRHTXT :	8B39 C
PRMTXT :	8B36 C	PRPTXT :	889B C
PRT02 :	8542 C	PRTCR :	84FE C
PRTOUT :	8500 C	PRTTEXT :	8A89 C
PTL :	83C7 C	PointZ :	8C84 C
PrepBlkB :	913C C	PrepBlkB :	911E C
PrepBlkG :	912C C	PrepBlkGB :	9113 C
PrepBlkGBT :	910F C	PrepBlkStoAdr :	90E5 C
PrepBlkStoAdrG :	9101 C	PrepBlkStoAdrX :	9108 C
PrepBlkT :	9126 C	PrepBlkT0 :	914C C
PrepBlkT0L :	9155 C	PrepErrMsg :	953A C
PrepErrMsg0 :	9557 C	PrepErrMsg1 :	955D C
PrepErrMsg2 :	9563 C	PrepErrMsg3 :	9569 C
PrepErrMsgM :	9553 C	ProcLanReq :	9220 C
ProcLanReqD :	9222 C	ProcLanReqM :	9260 C
ProcLanReqP :	9250 C	ProcReqTrans :	929B C
ProcSendBlk :	9082 C	PrtHelp :	95E4 C
PrtHelpE :	95F6 C	PrtHelpL :	95EA C
PrtMessage :	926F C	PrtNoCmd :	926C C
PrtOutA :	8523 C	RADR :	814B C
RADSIR :	0C0FC -	RAM :	0C0F0 -
RCL :	8833 C	READ :	9192 C

*RELAXED :	0 -	RETCALL :	0C076 -
RETSP :	0C07A -	RIGHT :	819F C
ROLL :	8088 C	ROM55A :	0F8 -
ROM55B :	0F9 -	ROM55C :	0FA -
ROM55M :	0FB -	ROMCTRL :	90 -
*ROMIN :	8C00 C	*RPOINT :	85E6 C
RPOS :	0C038 -	RRIGHT :	81B9 C
RSIRRAD :	0C0F5 -	Rc1C :	884D C
Rc1L :	8840 C	Rc1S :	8838 C
ReadBit :	8F49 C	ReadBitH :	8F4F C
ReadBlkRng :	91C7 C	ReadBlkRngS :	91EB C
ReadSync :	8ECB C	ReadSynch :	8EBE C
ReadSyncW0 :	8ED3 C	ReadSyncW1 :	8EDD C
RecvBlock :	94C3 C	RecvBlockL :	94C5 C
RecvByte :	94D8 C	RecvByteA :	94E1 C
RecvByteB :	94F5 C	RecvByteC :	94FC C
RecvByteD :	950B C	RecvByteE :	950F C
RecvByteF :	9522 C	RecvByteG :	952E C
RecvByteH :	9533 C	RecvHead :	927E C
RecvHeadM :	9275 C	RecvOther :	934E C
RecvOtherB :	9379 C	RecvOtherM :	936A C
RecvReqBlock :	92B8 C	RecvText :	92C8 C
RecvTextL :	92D2 C	RecvVram :	92F7 C
RecvVramA :	9303 C	RecvVramB :	9306 C
RecvVramC :	9313 C	RecvVramD :	9315 C
RecvVramE :	9322 C	RecvVramF :	9342 C
RollK :	809D C	RollL :	809F C
SBEEP :	0C0BF -	SCAN :	8C03 C
SEND :	901D C	SET :	95A6 C
SETWRK :	8AC2 C	SHCLR :	8116 C
SHEAD :	8DE2 C	SHeadB :	8F8A C
SHeadN :	8E22 C	SHeadT :	8E2E C
SPACE :	84FB C	SPEED :	0C07C -
SPTL :	0C137 -	SPVAL :	0C070 -
SSKIP :	8F24 C	SSkipL :	8F25 C
STACK :	0C13E -	STOP :	8C74 C
SUB :	8A8F C	SUB03 :	83A3 C
SUBTXT :	83A8 C	SWRK :	0C135 -
SYS55A :	0F4 -	SYS55B :	0F5 -
SYS55C :	0F6 -	SYS55M :	0F7 -
SYSCTRL :	8A -	SYST :	8054 C
SYST1 :	8579 C	SYST3 :	8046 C
SYSVAR :	0C030 -	ScanC :	84D7 C
ScanE1 :	84F5 C	ScanE2 :	84F4 C
ScanE3 :	84F3 C	ScanM :	84C3 C
ScanR :	8C08 C	ScanV :	84EF C
SendBlkRng :	9049 C	SendBlkRngS :	906D C

SendBlock :	93A5 C	SendBlockL :	93A7 C
SendByte :	93BF C	SendByteA :	93CC C
SendByteB :	93E0 C	SendByteC :	93E8 C
SendByteD :	9407 C	SendByteE :	9418 C
SendHead :	90AC C	SetPrtLan :	95BC C
SetPrtLanOff :	95C2 C	SetPrtLanOn :	95B7 C
SetWrkF :	8B14 C	SetWrkG :	8B0C C
SetWrkH :	8AF7 C	SetWrkI :	8AC8 C
SetWrkJ :	8AF6 C	SetWrkK :	8ADE C
SetWrkM :	8AEC C	SetWrkO :	8B16 C
SetWrkS :	8AD2 C	ShClrL :	811D C
SkipSpcX :	9584 C	Stack :	8000 -
*Start :	8000 C	SubA :	8AB0 C
SubL :	8A98 C	SubS :	8AB1 C
TAPEOUT :	8D6C C	*TAS00 :	0C03C -
*TASCII :	0C0B0 -	TEXTC :	81CE C
TEXTHEAD :	81C0 C	TEXTSP :	81C5 C
TEXTTH :	81C3 C	*TIME :	"20:28:58" -
TRANSF :	8FA4 C	*TRUE :	1 -
*TYPFIL :	0C1B3 -	T_ErrAddr :	8346 C
T_ErrData :	8291 C	T_Exec :	827D C
T_FileErr :	835C C	T_LanCrcErr :	9472 C
T_LanFrmErr :	9430 C	T_LanHelp :	9600 C
T_LanNRdyData :	9442 C	T_LanNRdySend :	945A C
T_LanOff :	95D6 C	T_LanOn :	95C9 C
T_LanRunBasic :	94A9 C	T_LanWaitPls :	9488 C
T_LanXferOK :	941E C	T_MemOF :	837C C
T_MgfReadOK :	8F93 C	T_MgfStop :	8DB4 C
T_NoCmd :	8392 C	T_OdReady :	836D C
TapeOutL :	8D6F C	Ton :	8170 C
TonB :	8178 C	TonD :	8171 C
TonE :	8185 C	TransfB :	8FBA C
TransfC :	8FCC C	UNDEF :	8A6E C
*VERSION :	142F -	VRAM :	0C000 -
VYSKA :	0C0FA -	VYSTUP :	0C1F4 -
WRBUFF :	8855 C	WRCHAR :	854A C
*WRCHAROLD :	8584 C	WRK :	83C1 C
WRLINE :	8858 C	WaitBitA :	8D62 C
WaitBitD :	8C39 C	WaitBitS :	8D5F C
WrCharL :	8567 C	WrLineL :	8860 C
WriteBit0 :	8D94 C	WriteBit0B :	8F41 C
WriteBit1 :	8DA2 C	WriteByte :	8D7E C
WriteByteL :	8D85 C	WriteGap :	8F38 C
WriteLeader :	8F2D C	X1 :	0C170 -
X2 :	0C173 -	XCCUR :	8A80 C
*XCHCUR :	8A7D C	XRCL :	87 -
*XSSKIP :	8FF4 C	XSSkipL :	8FF5 C



X_AM :	2 -		*X_CNT :	40 -	
*X_MGF_IN :	80 -		X_MGF_OUT :	1 -	
*X_POL :	8 -		X_RFU :	4 -	
X_SHF :	20 -		X_STOP :	10 -	
Y1 :	0C172 -		*Y2 :	0C174 -	
*Z80SYNTAX :	0 -				

525 symbols

38 unused symbols

▲ AS V1.42 Beta [Bld 269] - Source File mato-monitor.a8080 - Page 3 - 12/18/2025 20:28:58  
Monitor pocitaca MATO

Defined Macros:

-----

dm		kbd
moncmd		

3 macros

▲ AS V1.42 Beta [Bld 269] - Source File mato-monitor.a8080 - Page 4 - 12/18/2025 20:28:58  
Monitor pocitaca MATO

Code Pages:

-----

STANDARD (0 changed characters)

1 code page

0.03 seconds assembly time

3765 lines source file

3936 lines incl. macro expansions

2 passes

0 errors

0 warnings