

```

;
; Zdrojovy kod obsluzneho monitoru pocitace PMI-80
;*****
;*      (c)   www.nostalcomp.cz   2014      *
;*****
;
; Disassembled by:
;   DASMx object code disassembler
;   (c) Copyright 1996-1999   Conquest Consultants
;   Version 1.30 (Oct  6 1999)
;
; File: pmi80.rom,   Size: 1024 bytes,   Checksum: 82C9,   CRC-32: B93F4407
; Soubor PMI-80.ROM byl porovnan s obsahem ROM originalniho PMI a je identicky!
;
; Date:   Tue Apr 06 20:17:03 2010, revised 7.1. 2014!
;
; CPU:    Intel 8080 (MCS-80/85 family)
;
; Poznamka: RST 7 = FFh a je to nevyuzita (nenaprogramovana) pametova bunka
;
; Zaneseny tez upravu pro monitory PMI-80r, PMI-85 a PMI Z-80 (PMI-880)
;
; Binarni soubor prelozeny pomoci prekladace TASM 100% odpovida originalu!

```

```

; Definice portu a dulezitych mist v RAM:

```

```

PORT_A      .equ      0F8h
PORT_B      .equ      0F9h
PORT_C      .equ      0FAh
PORT_CW     .equ      0FBh
STACK       .equ      01FD9h
VIDEORAM    .equ      01FEFh
IN_ADR      .equ      01FF8h
IN_DATA     .equ      01FFAh
VIDEO_POINTER .equ     01FFCh
INT_VECTOR  .equ      01FE6h

```

```

.org 00000h

```

```

START:      ; 0000h - RESET
            mvi a,08Ah ; nastaveni sluzebniho 8255A.
            out 0FBh   ; CW 8Ah => rezim 0, PA out, PB inp, PC0-3 out, PC4-7 inp
            nop        ; lze vlozit DI (adresa 0004h = F3h) pro vsechny verze
            jmp L002E
;
;-----

```

```

ENTRY:      ; 0008h - ENTRY (BREAK STOP)

```

```

            shld $1FDF
            pop h
            shld $1FE2
            lxi h,0000h
            dad sp
            shld $1FE4
            lxi h,$1FDD
            sphl
            push b
            push d
            push psw
            pop h
            shld $1FDD
            lhld $1FEC
            lda $1FEE
            mov m,a
            lxi h,TEXT_BR_STOP
            jmp L0040

```

```

L002E:
  lxi h,STACK
  shld $1FE4
  jmp L003D          ;PMI Z-80 upravit skok na L_IM na adresu 003Bh:
                   ;na adresu 0035h staci dat misto 3Dh hodnotu 3Bh

  rst 7
;
;-----
INTERRUPT:
  jmp INT_VECTOR    ; 0038h - pevna adresa preruseni (typu RST7)

; nasleduji 2 volne byty, ktere lze vyuzit pro vlozeni instrukce IM1 pro Z-80:
  rst 7            ;003Bh = EDh (label L_IM: IM1 (režim přerušeni jako u 8080, 2 byte)
  rst 7            ;003Ch = 56h (IM1 je dvoubajtova!)
;
;-----
L003D:              ;pokracovani ENTRY
  lxi h,TEXT_PMI_80

L0040:
  lxi sp,STACK

L0043:
  shld VIDEO_POINTER
  call OUTKE
  lxi h,VIDEORAM
  shld VIDEO_POINTER

L004F:
  mvi a,01DH
  call CLEAR
  call OUTKE
  lxi h,TABPRIKAZY
  mvi b,006H

L005C:
  cmp m
  inx h
  jz L006D
  inx h
  inx h
  dcr b
  jnz L005C

L0067:
  lxi h,TEXT_ERROR
  jmp L0040

L006D:
  mov c,m
  inx h
  mov h,m
  mov l,c
  pchl
; PCHL - obsah HL do PC = skok na adresu, ktery byla v HL (indexovy skok).
; - adresy v HL jsou adresami exekutiv (rutin) jednotlivych prikazu.
;
; Konec hlavni programove smycky.
; Nasleduji jednotlivé vykonne exekutivy prikazu a podprogramy
;
;-----
PRIKAZ_MEM:        ;prikaz MEM
  mvi a,016H
  call CLEAR
  call MODAD
L007A:

```



```

rz
jnc L0197
lhld IN_ADR
ani 00FH
dad h
dad h
dad h
dad h
add l
mov l,a
shld IN_ADR
jmp MODAD
;
;-----
OUTDA:                                ;OUTDA
    lxi b,$1FF6
    lhld IN_DATA
    jmp L00C5
;
;-----
MODDA:                                ;MODDA
    call OUTDA
    call OUTKE
    rz
    jnc L019D
    nop
    nop
    ani 00FH
    dad h
    dad h
    dad h
    dad h
    add l
    mov l,a
    shld IN_DATA
    jmp MODDA
;
;-----
OUTKE:                                ;OUTKE
    call DISP ;volej DISP
    jnc OUTKE ;bylo neco zmacknuto? Kdyz ne, volej DISP
    rrc      ;odrotuj vpravo (z DISPu jde hodnota odrotovana vlevo,
            ;takze RRC ji jen restauruje)
    mov c,a  ;ulozime se kod klavesy

L011E:
    call DISP ;volej DISP
    jc L011E ;byly klavesy uvolneny? Kdyz ne, volej DISP
    call DISP ;jeste jednou zavolej DISP (proc?)
    mov a,c   ;natahni hodnotu kodu klavesy z C
    cpi 090H ;a porovnej ji s kodem klavesy = (90h)
    ret      ;RETURN??? Nemelo by zde byt nejake dalsi vyhodnoceni?
            ;Nebo jde jen o nastaveni priznaku?
;
; *****
; pravdepodobne jiz nepouzivany usek s vyhodnocenim (nevedou sem zadne skoky)
; rozsah adres je 0012Bh - 0013Fh. K dispozici tedy 20 bajtu!
    .db 008H
    dad b
    dcr c
    dcx b
    ldax b
    inx d
    inr d
    mvi c,00CH
    rrc

```

```

dcr b
ldax d
dcr c
dcx b
ldax b
;cpo LD9DF
cpo $D9DF
in ODDH
rst 7
; konec asi jiz nepouzivane casti kodu
; *****
;
;-----
DISP:                ;DISP
; tento podprogram byl okomentovan jiz v puvodni dokumentaci v AR 11/1984
; ale v tomto AR je spatna tabulka TABKEY. Spravne je to zde.
push h
push b
push d
lxi d,0000h ;nul D,E
mov b,d     ;nul B
mov a,d     ;nul A
sta $1FFE  ;inic STATUS
LOOP1:
mvi a,07FH ;blok segmenty
out 0F8H   ;port segment
nop
mov a,e
cma
out 0FAH   ;nastav digit
nop
lhld VIDEO_POINTER ;nastav ukazatel vypisu
dad d     ;pripocitej digit
mov c,m   ;vloz zobrazovana data
lxi h,TPREV ;nastav tabulku prevodu znaku
dad b
mov a,m   ;segment data
out 0F8H  ;port segment
nop
lda $1FFE ;lda STATUS
ora a
jnz NOKEY ;KEY?
mvi c,009H ;ANO
lxi h,0019AH ;lxi h, TABKEY-9 nastav tabulku prevodu klaves
in 0FAH   ;vstup KEY kod
nop
ani 070H  ;maska
rlc
rlc
jnc PRVA  ;ANO, první rada
rlc      ;NE
jnc DRUHA ;ANO, druhá rada
rlc      ;NE
jc NOKEY  ;C!=1, zadna rada
dad b     ;pripocitej radu
DRUHA:
dad b     ;pripocitej radu
PRVA:
dad b     ;pripocitej radu
dad d     ;pripocitej KEY
mov a,m   ;vyber kod KEY
sta $1FFE ;vloz do STATUS
NOKEY:
inr e     ;dalsi digit
mvi a,00AH
cmp e
jnz LOOP1 ;posledni digit?

```

```

lda $1FFE      ;lda STATUS   ANO, posledni
rlc           ;nastav carry
pop d
pop b
pop h
ret
;
L0197:
  lxi h,TEXT_ERR_ADRES
  jmp L0040
;
L019D:
  lxi h,TEXT_ERR_DATA
  jmp L0040
;
;-----
; tabulka klaves.
; 80h-8Fh = klavesy 0-F, 9xh = ridici klavesy, FFh = neosazeno

TABKEY:
  .db 080H, 084H, 088H, 091H, 08DH, 08CH, 089H, 085H, 081H ;3. radek matice tl.
  .db 082H, 086H, 08AH, 09AH, 08FH, 08EH, 08BH, 087H, 083H ;2. radek matice tl.
  .db 0FFH, 094H, 093H, 0FFH, 097H, 092H, 0FFH, 0FFH, 090H ;1. radek matice tl.

; tabulka pro prevod kodu znaku na sedmissegmentove vyjadreni
; pozor, je to invertovane!
TPREV:
  .db 040H      ;znak 0
  .db 079H      ;znak 1, atd. dle tabulky znaku:
  .db 024H, 030H, 019H, 012H, 002H, 078H, 000H, 018H, 008H, 003H, 046H, 021H
  .db 006H, 00EH, 007H, 023H, 02FH, 00CH, 047H, 063H, 048H, 071H, 037H, 07FH
  .db 009H, 02BH, 00BH, 02CH, 05DH, 03FH, 042H, 061H
  .db 07BH      ;znak c. 22 (caraka), posledni v tabulce znaku
  .db 011H      ;v tabulce znaku v manualu neni

rst 7          ; volny prostor (lze vyuzit pro upravy)
rst 7
rst 7
rst 7
rst 7

; .org 001E7h

TEXT_PMI_80: ;zacina na adrese 01E7h
  .db 01EH, 013H, 016H, 001H, 019H, 01FH, 008H, 000H, 01EH

;TEXT_PMI_80r:
; .db 01EH, 013H, 016H, 001H, 01FH, 008H, 000H, 012H 01EH
; text pro repliku PMI-80r od 01E7h

;TEXT_PMI_85:
; .db 01EH, 013H, 016H, 001H, 019H, 01FH, 008H, 005H, 01EH
; na 01EEh dat hodnotu 05h

;TEXT_PMI Z-80
; .db 013H, 016H, 001H, 019H, 019H, 002H, 01FH, 008H, 000H
; alternativni text pro Z-80 (od 01E7h)

;TEXT_PMI_880:
; .db 01EH, 013H, 016H, 001H, 019H, 008H, 008H, 000H, 01EH
; alternativni text pro U880D
; na 01EC dat hodnotu 08h

TEXT_ERR_ADRES:
  .db 00EH, 012H, 012H, 018H, 00AH, 00DH, 012H, 00EH, 005H

TEXT_ERR_DATA:

```

```

.db 00EH, 012H, 012H, 018H, 019H, 00DH, 00AH, 010H, 00AH

TEXT_ERROR:
.db 01EH, 019H, 00EH, 012H, 012H, 011H, 012H, 019H, 01EH

TABPRIKAZY:      ;kod prikazu (klavesy) + adresa obsluzneho programu (low, high)
.db 092H, 072H, 000H      ; MEM
.db 091H, 029H, 002H      ; EX
.db 097H, 05AH, 002H      ; BR
.db 09AH, 07EH, 002H      ; R
.db 094H, 04CH, 003H      ; SAVE
.db 093H, 08CH, 003H      ; LOAD
.db 0FFH, 0FFH, 0FFH      ; Zde mely byt asi Rst 7 (nevyuzite misto)

TEXT_BR_STOP:
.db 01EH, 00BH, 012H, 01FH, 005H, 010H, 011H, 013H, 01EH
;
;-----
PRIKAZ_EX:      ; prikaz EX
mvi a,020H
call CLEAR
lhld $1FE2
shld IN_ADR
call MODAD
lhld IN_ADR
shld $1FE2
mvi a,006H
out 0F8H
nop
mvi a,00FH
out 0FAH
nop
lxi h,STACK
sphl
pop d
pop b
pop psw
lhld $1FE4
sphl
lhld $1FE2
push h
lhld $1FDF
ret
;
;-----
PRIKAZ_BR:      ; prikaz BR
mvi a,00BH
call CLEAR
lhld $1FEC
shld IN_ADR
call MODAD
lhld IN_ADR
shld $1FEC
mov a,m
sta $1FEE
mvi m,0CFH
lhld $1FE2
dcx h
shld $1FE2
jmp PRIKAZ_EX
;
;-----
PRIKAZ_R:      ; prikaz R
mvi a,012H
call CLEAR
call OUTKE
jnc L0067

```

```
    ani 00FH
    lxi b,00006H
L028E:
    lxi h,0012AH
    dcx b
    dad b
    inr c
    dcr c
    jz  L004F
    cmp m
    jnz L028E
```

```
L029C:
    lxi h,0012FH
    call L02CD
    mov e,l
    lxi h,00134H
    call L02CD
    mov h,e
    shld $1FF6
    push b
    call L02CA
    push h
    mov c,m
    inx h
    mov h,m
    mov l,c
    shld IN_ADR
    call MODAD
    pop d
    mov a,l
    stax d
    inx d
    mov a,h
    stax d
    pop b
    dcr c
    jnz L029C
    jmp L004F
```

```
;/
L02CA:
    lxi h,00139H
L02CD:
    mvi b,000H
    dad b
    mov l,m
    mvi h,01FH
    ret
```

```
;/
-----
TOUT:                                ;TOUT      02D4h
    mvi b,009H
```

```
L02D6:
    mvi a,0C7H
    call L02EE
    mov a,c
    rar
    mov c,a
    mvi a,08FH
    rar
    call L02EE
    mvi a,047H
    call L02EE
    dcr b
    jnz L02D6
    ret
```

```
;/
L02EE:
```



```

    mvi d,020H
L02F0:
    out 0F8H
    mvi e,004H
L02F4:
    dcr e
    jnz L02F4
    xri 040H
    dcr d
    jnz L02F0
    ret
;
rst 7
;
-----
TIN:                                ;TIN    0300h
    mvi b,008H
    mvi d,000H
L0304:
    call L0342
    jc L0304
    call L0342
    jc L0304
L0310:
    call L0342
    jnc L0310
    call L0342
    jnc L0310
L031C:
    dcr d
    call L0342
    jc L031C
    call L0342
    jc L031C
L0329:
    inr d
    call L0342
    jnc L0329
    call L0342
    jnc L0329
    mov a,d
    ral
    mov a,c
    rar
    mov c,a
    mvi d,000H
    dcr b
    jnz L031C
    ret
;
L0342:
    mvi e,002H
L0344:
    dcr e
    jnz L0344
    in 0FAH
    ral
    ret
;
-----
PRIKAZ_SAVE:                        ; prikaz SAVE  034Ch
    mvi a,005H
    call CLEAR
    call MODAD
    call MODDA
    lxi h,TEXT_MG_RUN
    shld VIDEO_POINTER

```

```

call OUTKE
mvi a,023H
out 0F8H
mvi a,00FH
out 0FAH
mvi d,0F0H
mvi a,0C7H
call L02F0
lda IN_DATA
mov c,a
call TOUT
mvi a,010H
call CLEAR
lhld IN_ADR
L037E:
mov c,m
call TOUT
inr l
jnz L037E
L0386:
lxi h,TEXT_MG_STOP
jmp L0043
;
;-----
PRIKAZ_LOAD:                ; prikaz LOAD 038Ch
mvi a,014H
call CLEAR
call MODAD
call MODDA
lxi h,TEXT_MG_RUN
L039A:
shld VIDEO_POINTER
call OUTKE
L03A0:
lhld IN_ADR
mvi a,007H
out 0F8H
mvi a,00FH
out 0FAH
L03AB:
mvi d,0A0H
L03AD:
call L0342
jc L03AB
dcr d
jnz L03AD
call TIN
lda IN_DATA
cmp c
jnz L03CC
L03C1:
call TIN
mov m,c
inr l
jnz L03C1
jmp L0386
;
L03CC:
jc L03E7
mvi a,00FH
call CLEAR
mov a,c
lxi b,$1FF6
call L00C6
lxi h,VIDEORAM
shld VIDEO_POINTER
call OUTKE

```

```

    jmp L03A0
;
L03E7:
    lxi h,TEXT_MG_SPAT
    jmp L039A

TEXT_MG_SPAT:
    .db 01EH, 016H, 020H, 019H, 005H, 013H, 00AH, 010H, 01EH

    rst 7      ; volny prostor na konci PROM (lze vyuzit pro upravy)
    rst 7
    rst 7
    rst 7
    rst 7
    rst 7
    rst 7
    rst 7
    rst 7      ;03FFh konec 1KB PROM

; Poznamka: port PB sluzebni 8255A lze nastavit
; pouze na jednoduchy vstup/vystup v rezimu 0
;
; vstup:     mvi a,08Ah
;            out 0FBh
;
; vystup:    mvi a,088h
;            out 0FBh
;
; jina nastaveni tohoto obvodu nejsou mozna!
; Pridavny 8255A lze nastavit dle libosti.
;
;***** www.nostalcomp.cz *****
.END

```