

```

; Prvni verze komunikacniho programu PC Loader pro Juniora
; Startovni adresa nahravani je pevna - 0200h (poc. RAM za Stackem)
; Program sam je od adresy 0000h (v EPROM od 0600h - lze relokovat pomoci ORG)
; Na displeji se rozsviti "L" a program prijima jednotlivé bity.
; Nahravani se ukonci po prijeti 3 bytu #stopznak (zde 76h)
; Program hlida prekroceni volneho prostoru RAM!
; Prekroci-li RAM 03FFh, je vypsan znak E a Junior se zacykli.
; Lze tedy nahrát max. 512 byte. Otestovano a funkci.
; Nahravani uspesne konci resetem Juniora

;ddra .equ $01A81 ;registry portu Juniora
;ddrb .equ $01A83
;porta .equ $01A80
;portb .equ $01A82

;maska1 .equ $001 ;PB0 1. komunikacni linka
;maska2 .equ $020 ;PB5 2. komunikacni linka

;poc_c .equ $0AA ;pomocne registry
;poc_b .equ $0AB
;pom_d .equ $0AC
;poc_e .equ $0AD
;startaddr .equ $0AE ;plus $0AF

;konec .equ $0FC1D ;RESET
;stopznak .equ $076

.ORG $0000 ;relokace pro EPROM bude $0600

0000 A0 00 ldy #$00 ;vynulovani pocitadla adres
0002 A9 00 LDA #$00 ;adresa pocatku nahravani 0200h je nejprve
0004 85 AE STA startaddr ;ulozena do Zero Page na adresy AE,AF
0006 A9 02 LDA #$02
0008 85 AF STA startaddr+1
000A
000A A9 FF LDA #$FF ;nastaveni DDRA, DDRB u PIA
000C 8D 81 1A STA ddra ;port A je vystup DDRA = FF
000F A9 DE LDA #$DE
0011 8D 83 1A STA ddrb ;port B je vystup, krome vstupu PB0 a PB5 (log. 0)
0014
0014 A9 47 LDA #$47 ;nastav znak L
0016 8D 80 1A STA porta ;a zapis na port jednotlivych segmentu
0019 A9 08 LDA #$08 ;nastav pozici 1. zleva
001B 8D 82 1A sta portb ;a aktivuj katodu
001E
001E A9 03 LDA #$03 ;Pocitadlo bytu #stopznak
0020 85 AD STA poc_e
0022
0022 pokr2
0022 A9 00 LDA #$00 ;Pomocna prom. pro ukladani bitu prijimaneho byte
0024 85 AA STA poc_c
0026 A9 08 LDA #$08 ;Pocitadlo prijatych bitu kazdeho byte
0028 85 AB sta poc_b
002A
002A cek1
002A AD 82 1A LDA portb ;cekani na jednicku. Cti port B
002D 85 AC STA pom_d ;uloz hodnotu do D registru (pomocne uchovani)
002F 29 20 AND #maska2 ;maskuj ostatni bity krome PB5
0031 F0 F7 BEQ cek1 ;je PB5 jednicka? Kdyz ne vrat se a cti znovu.
0033 A5 AA LDA poc_c ;Ano, hodnotu z pomocneho registru C
0035 2A ROL A ;odrotujeme vlevo (bez carry)
0036 A5 AA LDA poc_c ;6502 rotuje jen pres Carry, takže rotujeme 2x
0038 2A ROL A ;nejprve naplnime Carry a pak se rotuje "naostro"
0039 85 AA STA poc_c ;a uložíme zpět do reg. C
003B A5 AC LDA pom_d ;vezmeme uložený stav portu PB
003D 29 01 AND #maska1 ;vymaskujeme bity krome PB0
003F 05 AA ORA poc_c ;a podle PB0 z portu nastavíme D0 v C registru

```

```

0041 85 AA      STA poc_c      ;a ulocime C
0043
0043          cek2
0043 AD 82 1A   LDA portb      ;cekani na nulu. Cti port PB
0046 29 20     AND #maska2    ;maskuj ostatni bity krome PB5
0048 D0 F9     BNE cek2      ;je PB5 nula? Kdyz ne vrat se a cti znovu.
004A C6 AB     DEC poc_b      ;ANO, dekrementuj pocitadlo bitu z bytu
004C D0 DC     BNE cek1      ;neni-li jeste 0, skoc na prijem dalsiho bitu
004E A5 AA     LDA poc_c      ;je nulove = prijaty cely byte
0050 91 AE     STA (startaddr),Y ;uloz byte na adresu startaddr+Y
0052 A5 AA     LDA poc_c
0054 C9 76     CMP #stopznak  ;porovnej prijaty byte s hodnotou #stopznak
0056 D0 07     BNE dal2      ;neni-li to #stopznak, skoc na inkr. indexu
0058 C6 AD     DEC poc_e      ;je-li to #stopznak, dekrementuj pocitadlo
005A D0 07     BNE dal3      ;je pocitadlo uz nula? NE, skoc na dalsi byte
005C 4C 1D FC   jmp konec     ;kdyz ANO, koncime!
005F
005F          dal2
005F A9 03     LDA #$03      ;reset pocitadla #stopznaku
0061 85 AD     STA poc_e
0063
0063          dal3
0063 C8        INY                ;inkrementujeme Y
0064 D0 15     BNE dal4      ;pokud index Y pretekl
0066 E6 AF     inc (startaddr+1) ;inkrementujeme stranku!
0068 A5 AF     lda (startaddr+1)
006A C9 04     cmp #$04      ;prekroceni RAM ($03FF)?
006C D0 0D     bne dal4      ;NE, jedeme dal
006E
006E A9 06     LDA #$06      ;ANO, nastav znak E
0070 8D 80 1A  STA porta     ;a zapis na port jednotlivych segmentu
0073 A9 08     LDA #$08      ;nastav pozici l. zleva
0075 8D 82 1A  sta portb     ;aktivuj katodu
0078 4C 78 00  jmp *         ;a zacykli se
007B
007B          dal4
007B 4C 22 00  jmp pokr2     ;dalsi znak
007E
007E          .END

```

```
tasm: Number of errors = 0
```