```
;-------------------------------------------------------------------------------;
; DC MOTOR CONTROLLER V1.0              (C)PowerPartner Co.,Ltd.
;
; filename:         rs232SP4i.asm  ********************>>>>>>>>>>>>>>>
;
;a-menu RS232
;b-test sub serial,getline
;c-timer0 enable + backgroud intr. ;/
;d-timer1 enable = counter+TIM1_CMPA intr. ;/
;d-cmd "L" ;/
;d-do_fwd/do_rev fixed    :1byte=ok , 2byte=ok  ;/
;e-do_fwd/do_rev <pulse>
;e-get_val=BL:AH:AL=3byte /run2byte=ok              ;/
;f-fslip/rslip ;fixed=38,38
;f-do_fwd/do_rev <pulse> ..adjslip                  ;/
;g-fslip/rslip :eeprom save/load                ;/
;
;h-rearrange/rename subroutine
;h-cmd KL,YZ listpos/tcnt1, zeropos/tcnt1
;h-change PC-i --> lbIXX:
;
;i-rxint: review cli/sei                    ;/
;i-do_showPOS bug.correct okay              ;/
;i-Timer0.intr =calc _posXHL                ;/
;i-do_fwd/do_rev updateposlast   ;/



;-------------------------------------------------------------------------------;
.nolist
.include "tn2313def.inc"
.include "avr1.inc"
.list

;-----------------------------------------------------
        ;hardware
.equ    SYSCLK          = 20000000
.equ    BPS             = 38400

.equ    motorout        = PORTD
.equ    motorddr        = DDRD

.equ    mfwd            = 2
.equ    mrev            = 3
;equ    mzero           = xx

;-----------------------------------------------------
        ;registers
.def    _PosX           = r12   ;Current position
.def    _PosH           = r11   ;
.def    _PosL           = r10   ;/
.def    _0              = r15   ;Permanent zero register
.def    _Flags  = r25   ; x|x|x|x| x|rev|fwd|echo

;-----------------------------------------------------------;
; EEPROM Area
;-----------------------------------------------------------;
.eseg
                ;EEbank#0 parameters
EEfslip: .db            39
EErslip: .db            37
EEfslow:        .dw             0x000
EErslow:        .dw             0x000,0x0000
                .db             0,0,0,0 ,0,0,0,0

.equ            N_PARM = 8      ; Number of words per bank.

;-----------------------------------------------------------;
; Data memory area
;-----------------------------------------------------------;
.dseg
.org    RAMTOP

                ;DC Motor Parameters
Parms:
Fslip:  .byte   1       ;fwd slip = 0-255
Rslip:  .byte   1       ;rev slip = 0-255
Fslow:  .byte   2       ;fwd slow = 0-1000
Rslow:  .byte   2       ;rev slow = 0-1000

Poslast: .byte  3       ;Pos.last:X:H:L

                ;Host command
```

```
RxBuf:  .byte   2+16    ; Serial receive buffer (Rp, Wp, Buff[16])
LineBuf:        .byte   20      ; Command line input buffer

; Command/Servo registers
;CtPos: .byte   3       ;Position       g/j     mode 3
;CtSub: .byte   2       ;Sub command s          mode 0/1/2
;PvInt:  .byte   2       ;Integration register
;PvPos: .byte   2       ;Velocity detection register
;OvTmr: .byte   2       ;Torque limit timer
;Mode:  .byte   1       ;Servo Mode     m

; Displacements referrd from RAMTOP
;.equ   iCtSub  = CtSub-RAMTOP
;.equ   iPvInt  = PvInt-RAMTOP
;.equ   iPvPos  = PvPos-RAMTOP
;.equ   iOvTmr  = OvTmr-RAMTOP
;.equ   iMode           = Mode-RAMTOP


;*****************************************************************
; Program code
;*****************************************************************
.cseg
        ; Interrupt Vectors (ATtiny2313)
        rjmp    reset           ;00 Reset
        rjmp    0               ;01 INT0
        rjmp    0               ;02 INT1
        rjmp    0               ;03 TC1 CAPT
        rjmp    TIM1_CMPA       ;04 TC1 COMPA
        rjmp    0               ;05 TC1 overflow
        rjmp    background      ;06 TC0 overflow
        rjmp    rxint           ;07 USART0 Rx ready
        rjmp    0               ;08 USART0 Tx UDRE
        rjmp    0               ;09 USART0 Tx empty
        rjmp    0               ;10 Analog comparator
        rjmp    0               ;11 PCINT
;       rjmp    0               ;12 TC1 COMPB
;       rjmp    0               ;13 TC0 COMPA
;       rjmp    0               ;14 TC0 COMPB
;       rjmp    0               ;15 USI START
;       rjmp    0               ;16 USI OVF
;       rjmp    0               ;17 EEPROM
;       rjmp    0               ;18 WDT

;-----------------------------------------------------
reset:
        ;--Stack.clear
        outi    SPL, low(RAMEND)        ;Stask ptr

        ;---SRAM.clear
        clr     _0                      ;Clear RAM
        ldiw    Y, RAMTOP               ;
        st      Y+, _0          ;
        cpi     YL, low(RAMTOP+128)     ;
        brne    PC-2                    ;/

        ;---PortD,B
        outi    PORTD, 0b01111111       ;Initialize PORTD
        outi    DDRD,  0b00000010       ;/
        outi    PORTB, 0b11100000       ;Initialize PORTB
        outi    DDRB,  0b00011111       ;/

        ;--Timer0,1
        outi    TCCR0B,0b00000101       ;Init Timer0, CS210= 101 =INTClk/1024
;       outi    TCCR0A,0b00000000       ;na/

        outi    TCCR1B,0b00000110       ;Init Timer1, CS210= 110
=ExtClk/Falling
;       outi    TCCR1A,0b00000000       ;na
;       outi    TCCR1C,0b00000000       ;na

;       outi    TIMSK,0b00000010        ;T0overflow.INTR
        outi    TIMSK,0b01000010        ;T1cmpA.INTR|...|T0overflow.INTR|

        ;--USART
        ldiw    A, SYSCLK/16/BPS-1      ;USART
        outw    UBRR, A                 ;
        outi    UCSRB, 0b10011000               ;RXCIE,RXen,TXen,8bit/
        outi    UCSRC, 0b00000110               ;Async,None,1stop,8bit/

        ;--init var....
        ldi     _Flags, 0b00000001      ;echoON,MotorStop
```

```
        ldi     AL,0                            ;eeprom bank0 parameters
        rcall   load_parms

        ;--StartUp
        ldiw    Z, (m_start<<1)  ;Start up message
        rcall   tx_romstr                ;/
        sei

;-----------------------------------------------------------;
; Command processing loop
;-----------------------------------------------------------;
main:
;       rcall   calcPosXHL
        ldiw    Z, (m_prompt<<1)         ;Display command prompt
        rcall   tx_romstr                ;/

        rcall   get_line ;Get a command line
        ld      BH,X+            ;BH = command char
        cpi     BH,'a'   ;CAPS
        brcs    PC+2             ;
        subi    BH,0x20;/
        cpi     BH,' '   ;Null line ?
        brlt    main

        cpi     BH,'F'   ;1.Forward?
        rjeq    do_fwd  ;
        cpi     BH,'R'   ;2.Reverse?
        rjeq    do_rev  ;
        cpi     BH,'G'   ;22.Goto?
        rjeq    do_goto;

        cpi     BH,'K'           ;3.List TCNT1?
        rjeq    do_showtcnt1     ;/
        cpi     BH,'L'           ;3.List posXHL?
        rjeq    do_showPOS               ;/

        cpi     BH,'Z'           ;33.Zero posXHL?
        rjeq    do_zeroPOS               ;/

        cpi     BH, 'S'          ;4.ShowFR slip set?
        rjeq    do_showSlipSet ;

        cpi     BH, 'T'          ;5.FWD slip set?
        rjeq    do_fwdslipset    ;
        cpi     BH, 'U'          ;5.REV slip set?
        rjeq    do_revslipset    ;
        cpi     BH, '?'          ;99.Help?
        breq    do_help          ;

cmd_err:
        ldiw    Z,(m_error<<1)   ;Syntax error
        rcall   tx_romstr
        rjmp    main     ;PC+3

do_help:
        ldiw    Z,(m_help<<1)    ;Help
        rcall   tx_romstr
        rjmp    main
        ;/sss

;--------------------------------------
do_goto:
        rcall   get_val          ;val24= BL:AH:AL
        rjeq    cmd_err
        rjmp    main

;--------------------------------------
do_rev:
        rcall   get_val          ;val24= BL:AH:AL
        rjeq    cmd_err
        lds     BL,Rslip                 ;AH:AL=revset - Rslip
        sub     AL,BL                    ;
        sbci    AH,0                     ;
        out     OCR1AH,AH                ;rev.pulse.set
        out     OCR1AL,AL                ;

        rcall   updatePoslast    ;updateposlast
        outi    tcnt1h,0
        outi    tcnt1l,0

        cbi     motorout,mfwd    ;MotorRev
        sbi     motorout,mrev
        ldi     _flags,0b00000101        ;flag.rev
```

```
;       rjmp    do_showTCNT1 ;/
        rjmp    do_showPOS          ;/or/

;-------------------------------------
do_fwd:
        rcall   get_val         ;val24= BL:AH:AL
        rjeq    cmd_err
        lds     BL,Fslip        ;AH:AL=fwdset - Fslip
        sub     AL,BL                   ;
        sbci    AH,0                    ;
        out     OCR1AH,AH               ;fwd.pulse.set
        out     OCR1AL,AL

        rcall   updatePoslast   ;update poslast
        outi    tcnt1h,0
        outi    tcnt1l,0

        sbi     motorout,mfwd   ;MotorFwd
        cbi     motorout,mrev
        ldi     _flags,0b00000011

;       rjmp    do_showTCNT1 ;/
        rjmp    do_showpos          ;/or/


;-------------------------------------
updateposLast:
        sts     poslast,_posL
        sts     poslast+1,_posH
        sts     poslast+2,_posX
        ret

;---------------------------------------
do_showSlipSet:
        rcall   showSlipSet
        rjmp    main

do_fwdslipset:
        rcall   get_val         ;get new Fslip?
        rjeq    cmd_err
        sts     Fslip,AL
        ldi     BL,low(EEfslip)  ;AL=data, BL=addr
        sbic    EECR, EEPE              ;loop_wait_writedone
        rjmp    PC-1                    ;/
        out     EEAR, BL                ;out_addr1
        out     EEDR, AL                ;out_data1
        sbi     EECR, EEMPE             ;out_write1
        sbi     EECR, EEPE              ;/
        rjmp    main

do_revslipset:
        rcall   get_val         ;get new Fslip?
        rjeq    cmd_err

        sts     Rslip,AL
        ldi     BL,low(EErslip)  ;AL=data, BL=addr
        sbic    EECR, EEPE              ;loop_wait_writedone
        rjmp    PC-1                    ;/
        out     EEAR, BL                ;out_addr1
        out     EEDR, AL                ;out_data1
        sbi     EECR, EEMPE             ;out_write1
        sbi     EECR, EEPE              ;/
        rjmp    main

;-----------------------------------------------;
do_zeroTCNT1:
        ;Set zero position counter
        outi    tcnt1h,0        ;out hibyte first
        outi    tcnt1l,0        ;then out lobyte
        rjmp    do_showTCNT1 ;/show loc

do_showTCNT1:
        ;Show location counter
        ;IN:  TCNT1H:L
        ldi     AL, 0x0a        ;[LF]
        rcall   txmit
dtc_p:
        ldi     AL, 0x0d        ;[CR]
        rcall   txmit           ;
```

```
;       cli                     ;**************>>>>>>>>???????????
        in      AL,tcnt1l       ;lobyte.first ;/
        in      ah,tcnt1h
        mov     BL, _0   ;
;       sei                     ;**************>>>>>>>>???????????

        mov     T0H, al  ;
        rcall   tx_valdec       ;
        ldi     AL, ' '    ;
        rcall   txmit           ;/
dtc_w:
        rcall   receive  ;pc-4::Break if any key was pressed
        rjne    main            ;/

        in      al,tcnt1l ;
        cp      T0H, al  ;Continue if not changed
        breq    dtc_w           ;/
        rjmp    dtc_p           ;/ok/

;-----------------------------------------------
do_zeroPOS:
        ;Set zero position counter
        outi    tcnt1h,0        ;out hibyte first
        outi    tcnt1l,0        ;then out lobyte
        sts     poslast,_0
        sts     poslast+1,_0
        sts     poslast+2,_0
        mov     _posL,_0
        mov     _posH,_0
        mov     _posX,_0
        rjmp    do_showPOS              ;/show loc

do_showPOS:
        ;Show location counter
        ;IN:  _posXHL
        ldi     AL, 0x0a        ;[LF]
        rcall   txmit
dps_p:
        ldi     AL, 0x0d        ;[CR]
        rcall   txmit           ;
```

```
        cli                     ;**************>>>>>>>>?????????
        mov     al,_posl
        mov     ah,_posh
        mov     bl, _posX       ;
        sei                     ;**************>>>>>>>>?????????

        mov     T0H, al  ;
        rcall   tx_valdec       ;  tx-decimal
        ldi     AL, ' '    ;
        rcall   txmit           ;/

dps_w:
        rcall   receive  ;pc-4::Break if any key was pressed
        rjne    main            ;/

        cp      T0H, _posL      ;Continue if not changed
        breq    dps_w           ;PC-4/
        rjmp    dps_p           ;/ok/

;************************************************************
;200--subXXX
;************************************************************
;-----------------------------------------

MotorStop:
        cbi     motorout,mfwd
        cbi     motorout,mrev
        ldi     _flags,0b00000001
        ret

MotorBrake:
        sbi     motorout,mfwd
        sbi     motorout,mrev
        ldi     _flags,0b00000001
        ret
        ;/


;-----------------------------------------------------
showSlipset:
        ;Show Fslip SRAM value
```

```
        ldi     AL,0                    ;read eerom:bank0
        rcall   load_parms

        ldi     Al,10
        rcall   txmit
        ldi     Al,13
        rcall   txmit

        ldi     Al,'T'
        rcall   txmit
        ldi     Al,'='
        rcall   txmit
        mov     BL,_0
        mov     AH,_0
        lds     Al,low(fslip)   ;fslip.val
        rcall   tx_valdec

        ldi     Al,13                   ;[CR][LF]
        rcall   txmit
        ldi     Al,10
        rcall   txmit

        ldi     Al,'U'
        rcall   txmit
        ldi     Al,'='
        rcall   txmit
        mov     BL,_0
        mov     AH,_0
        lds     Al,low(rslip)   ;rslip.val
        rcall   tx_valdec               ;/ok/
        ret

;-eeprom-----------------------------------------
load_parms:
        ;in:  AL=bank#
        ;out: BH=EEROM.addr
        ;       Y =SRAM.addr
        ;       AH=no.byte
        rcall   get_eeadr
load1:
        out     EEAR, BH
        inc     BH
        sbi     EECR, EERE
        in      AL, EEDR
        st      Y+, AL
        dec     AH
        brne    load1   ;PC-6           ;/smc3/
        ret

get_eeadr:
        ;IN:  AL=bank#
        ;OUT: BH=EEROM.addr
        ;    Y =SRAM.addr
        ;    AH=no.byte
        ldi     AH, N_PARM*2    ;word*2=byte
        clr     BH
gee1:
        subi    AL, 1
        brcs    gee2    ;PC+3
        add     BH, AH
        rjmp    gee1    ;PC-3

gee2:
        ldiw    Y, Parms                ;/smc3/
        ret


;*********************************************************
;INTRxxx
;*********************************************************
;background0:
;       ;--get _PosX:H:L = TCNT1H:L
;       in      _PosL,TCNT1L    ;in lobyte first
;       in      _PosH,TCNT1H    ;then in hibyte
;       mov     _PosX,_0                ;/
;       reti

background:
        ;--push.1234
        push    Al              ;ps1
        in      al,sreg ;ps2
        push    Al              ;ps22
```

```
        push    ah              ;ps3
        push    bl              ;ps4
        push    bh              ;ps5
        push    cl              ;ps6

        ;--calcposXHL:
        ;IN:    _flags, tcnt
        ;OUT: _PosX:H:L
        in      bl,TCNT1L               ; in lo.byte then in hi.byte
        in      bh,TCNT1H               ;b=tnct1   >>>

        lds     al,poslast              ;ca=poslast >>>
        lds     ah,poslast+1
        lds     cl,poslast+2

        cpi     _flags,0b00000011       ;+fwd?
        breq    bfwd
        cpi     _flags,0b00000101       ;-rev?
        breq    brev
        cpi     _flags,0b00000001       ;stop?   ;**********?????????
        breq    bfwd                    ;stop= +fwd       ;**********?????????
        ;breq   bsave                   ;stop= 0none
        rjmp    bpop                    ;else bpop

bfwd:   ;--calc fwd
        add     al,bl                   ;ca=poslast+tcnt1 >>>
        adc     ah,bh
        adc     cl,_0
        rjmp    bsave

brev:   ;--calc rev
        sub     al,bl                   ;ca=poslast-tcnt1 >>>>
        sbc     ah,bh
        sbc     cl,_0

bsave:;--save
        mov     _posL,al                ;pos=ca   >>>
        mov     _posH,ah
        mov     _posX,cl

bpop:   ;--pop.1234
        pop     cl              ;ps6
        pop     bh              ;ps5
        pop     bl              ;ps4
        pop     ah              ;ps3
        pop     al              ;ps2
        out     sreg,al  ;ps2
        pop     Al              ;ps1
        reti

;-----------------------------
TIM1_CMPA:
        sbi     motorout,mfwd           ;motorbrake
        sbi     motorout,mrev
        reti
        ;/

;--------------------
rxint:
        ;Read RxDATA -->RxBuf
        ;push.1234
        push    AL              ;push1
        in      AL, SREG
        push    BL              ;push2
        in      BL, UDR
        cbi     UCSRB, RXCIE
;       sei                     ;-------********>>>>>>>>>>>>ORG
        pushw   A               ;push3
        pushw   Y               ;push4

        ldiw    Y, RxBuf
        ldd     AL, Y+0
        ldd     AH, Y+1
        inc     AH
        andi    AH, 15

        cp      AH, AL
        breq    rxexit   ;pc+6
        std     Y+1, AH
        dec     AH
        andi    AH, 15
        add     YL, AH
```

```
        std     Y+2, BL

rxexit:
        ;pop.4321
        popw    Y               ;pop4
        popw    A               ;pop3
        pop     BL              ;pop2
        out     SREG, AL
        pop     AL              ;pop1
;       cli                     ;-------**********>>>>>>>>>>>ORG
        sbi     UCSRB, RXCIE
        reti                    ;/smc3/


;-----------------------------------------------------
;RXsub..
;-----------------------------------------------------
;-rx-----------------------------------------------
receive:
        ; Receive a char into AL. (ZR=no data)
        push    AH
        pushw   Y
        ldiw    Y, RxBuf
        cli                     ;***************>>>
        ldd     AH, Y+0
        ldd     AL, Y+1
        cp      AH, AL
        breq    rcv1    ;PC+8
        add     YL, AH
        ldd     AL, Y+2
        sub     YL, AH
        inc     AH
        andi    AH, 15
        std     Y+0, AH
        clz
rcv1:
        sei                     ;***************>>>
        popw    Y
        pop     AH
        ret
```

```
;-rx-----------------------------------------
get_line:
        ;Input a command line into LineBuf.
        ;IN:    rxint->linebuf
        ;OUT:
        ;DISP: txecho
        ldiw    X,LineBuf
        ldi     BH,0

gl1:  rcall     receive
        breq    gl1             ;PC-1
        st      X,AL

        cpi     AL,0x0d ;[CR]?
        brne    gl4             ;pc+4
        ldiw    X,LineBuf
        rjmp    txecho

gl4:  cpi AL,0x08 ;[BS]?
        brne    gl7             ;PC+7
        cpi     BH,0
        breq    gl1
        rcall   txecho
        sbiw    XL,1
        dec     BH
        rjmp    gl1

gl7: cpi  AL,' '    ;[SPACEbar]?
        brcs    gl1
        cpi     BH,20-1
        breq    gl1
        rcall   txecho

        adiw    XL,1
        inc     BH
        rjmp    gl1             ;/sp4/

;-rx-----------------------------------------
get_val:
        ;---------------------------------------
        ; Get value of decimal string
```

```
        ; IN:       X = ASCII string pointer
        ; OUT:  X                       = updated
        ;       BL:AH:AL = 24bit value
        ;                       C  = if    C=1: error
        ;       Z  = elsif Z=1: end of line, value=0
        ;          = else:      BL:AH:AL = 24bit value
        ;
        ;  Positive:   "300"
        ;  Negative:   "-125000"
        ;----------------------------------------
        clt                             ;clr Tbitreg
        clr     AL
        clr     AH
        clr     BL

        ld      BH,X+
        cpi     BH,' '
        brcs    gd_n
        breq    PC-3

        cpi     BH,'-'
        brne    PC+3
        set                             ;set Tbitreg

gd_l:
        ld      BH,X+

        cpi     BH,' '+1
        brcs    gd_e
        subi    BH,'0'
        brcs    gd_x
        cpi     BH,10
        brcc    gd_x

        ldi     CL, 25
        ldi     CH, 10
        sub     r0, r0
        lsr     r0
        ror     BL
        ror     AH
        ror     AL
        brcc    PC+2
        add     r0, CH
        dec     CL
        brne    PC-7
        add     AL, BH
        adc     AH, _0
        adc     BL, _0
        rjmp    gd_l

gd_x:
        sec
        sez
        ret

gd_e:
        sbiw    XL,1
        brtc    PC+7
        com     AL
        com     AH
        com     BL
        subi    AL,-1
        sbci    AH,-1
        sbci    BL,-1
        clc                             ;clear carry.bit
        ret

gd_n:
        sbiw    XL,1
        clc                             ;clear carry.bit
        sez                             ;set  zero.bit
        ret
        ;/smc3/



;-tx-----------------------------------;
tx_valdec:
        ;Display a value in decimal string
        ;
        ;Call: BL:AH:AL  = 24bit signed value to be displayed
        ;Ret:  BL:AH:AL = broken
```

```
        ldi     CH,' '
        sbrs    BL, 7
        rjmp    PC+8            ;/
        com     AL
        com     AH
        com     BL
        adc     AL,_0
        adc     AH,_0
        adc     BL,_0
        ldi     CH,'-'

dp_8:   ;
        clr     T0L             ;digit counter
        inc     T0L             ;---- decimal string generating loop
        clr     BH              ;var1 /= 10;
dp_18:
        ldi     CL,24           ;
dp_9: ;
        lslw    A               ;
        rolw    B               ;
        cpi     BH,10           ;
        brcs    PC+3            ;/
        subi    BH,10           ;
        inc     AL              ;

dp_3:   ;
        dec     CL              ;<--
        brne    PC-9            ;//

        addi    BH,'0'  ;Push the remander (a decimal digit)
        push    BH              ;/

        cp      AL,_0           ;if(var1 =! 0)
        cpc     AH,_0           ; continue digit loop;
        cpc     BL,_0           ;
        brne    PC-18           ;/
        mov     AL, CH  ;Sign
        rcall   txmit           ;/

dpm3:   ;
        pop     AL              ;Transmit decimal string
        rcall   txmit           ;<-- Put a char to memory, console
                                ;    or any other display device
        dec     T0L             ;
        brne    PC-3            ;/
        ret                     ;/smc3/

;-tx-----------------------------------;
TX_romstr:
        ;Txmit ROM string
        ; IN:   Z = top of the string (ASCIZ)
        ; OUT:  Z = next string
        lpm     AL, Z+
        tst     AL
        brne    txr1    ;PC+2
        ret
txr1:
        rcall   txmit
        rjmp    TX_romstr               ;/smc3/

;-tx-----------------------------------;
TXecho:
        ; Transmit AL.
        sbrs    _Flags, 0       ;echo off?
        ret
TXmit:
        sbis    UCSRA, UDRE
        rjmp    TXmit   ;PC-1
        out     UDR, AL
        ret

;-tx-------------------------------------------------------;
;ROM Strings

m_prompt:       .db     13,10, "PP>", 0
m_error:        .db     13,10, " type ? for help.", 0
m_start:.db     13,10,13,10, "PowerPartner -- DC Motor Controller V1.0  [?:help]
",13,10, 0

m_help: .db     13,10,13,10
                .db "K - list TCNT1",13,10,"L - list pos",13,10
                .db "Z - zero pos",13,10,13,10
```

```
.db "F<pulse> - fwd cmd",13,10, "R<pulse> - rev cmd",13,10
.db "G<pos> - goto cmd ",13,10,13,10
.db "S - show T/U - fwd/rev slipset",13,10
.db "T<pulse> - T=fwd slip set",13,10,"U<pulse> - U=rev slip set",13,10
.db "? - help",13,10, 0,0
```