

```

        .include "D:\Program Files\Atmel\AVR Studio\Appnotes\tn25def.inc"

;Автор: Кравченко А.В.
;Дата: 18.06.2010
;Версия: 1.1.1
;Проект Робоход
;Имя файла: Robohod.asm
;Микроконтроллер: AVR tiny25.
;Тактовая частота: 6,4 МГц
;Питание: стабилизированное 5,0 В

```

```

.def     tmp      = r16
.def     tmp2     = r17
.def     tmp3     = r18
.def     tempo    = r19
.def     adc2     = r22
.def     adc1     = r23
.def     Yo       = r20
.def     Xo       = r21
.def     Mo       = r26
.def     adr      = r0
.def     d1       = r24
.def     d2       = r25
.def     s1       = r27
.def     s2       = r28

```

```

.cseg
.org 0

        rjmp      RESET
        nop
        ;rjmp     EXT_INT0 прерывание не используется
        ;rjmp     EXT_PCINT прерывание не используется
        ;rjmp     TIMER1_COMPA совпадение таймер 1
        ;rjmp     TIMER1_OVF0 прерывание от таймера
        ;rjmp     TIMER0_OVF0 прерывание от таймера
        ;rjmp     EE_RDY прерывание от таймера
        ;rjmp     ANA_COMP прерывание не используется
        ;rjmp     ADC прерывание не используется
        ;rjmp     TIMER1_COMPB прерывание от таймера
        ;rjmp     TIMER0_COMPA прерывание от таймера
        ;rjmp     TIMER0_COMPB прерывание от таймера
        ;rjmp     WDT тайм аут сторожевой таймен
        ;rjmp     USI_START состояние START
        ;rjmp     USI_OVF прерывание не используется

```

```

.org 20
;настройка порта В
RESET:  nop
        clr      tmp
        out      DDRB, tmp
        out      PORTB, tmp      ;обнулить порт В
        clr      tmp
        ldi      tmp, $0B        ;PB0,PB1,PB4- выходы
        out      DDRB, tmp
        nop
nStart: nop
;Один шаг левый
        clr      tmp
        ldi      tmp, $04
        out      PORTB, tmp      ;обнулить порт В

```

Страница

```

        rcall    zader
        clr      tmp
        out      PORTB, tmp          ;обнулить порт В
;Один шаг правый
        clr      tmp
        ldi      tmp, $02
        out      PORTB, tmp          ;обнулить порт В
        rcall    zader
        clr      tmp
        out      PORTB, tmp          ;обнулить порт В
;Проверка фото датчиков
;вход в подпрограмму опроса АЦП1 правые датчики
        nop
        ldi      tmp3, $93 ;источник питания, ADC2,ADMUX
        ldi      tmp2, $C6 ;одиночное, запуск, 1/64,ADCSRA
        ldi      tempo, $00 ;ADCSRB
;опрос уровня правых датчиков
        rcall    adcn
        mov      S1, tmp2             ;загрузка младшего разряда
        mov      S2, tmp3             ;загрузка старшего разряда
        cln
        clz
        cls
        cpi      S1, $10
        brsh     dalee
        nop
        cpi      S2, $00
        brne     dalee
        nop
        rjmp     nazadlevo
dalee:  nop
;вход в подпрограмму опроса АЦП1 левые датчики
        nop
        ldi      tmp3, $91 ;источник питания, ADC2,ADMUX
        ldi      tmp2, $C6 ;одиночное, запуск, 1/64,ADCSRA
        ldi      tempo, $00 ;ADCSRB
;опрос уровня левых датчиков
        rcall    adcn
        mov      d1, tmp2             ;загрузка младшего разряда
        mov      d2, tmp3             ;загрузка старшего разряда
        cln
        clz
        cls
        cpi      d1, $10
        brsh     dalee2
        nop
        cpi      d2, $00
        brne     dalee2
        nop
        rjmp     nazadpravo
        nop
dalee2: nop
;Вычитаем значения левых датчиков от значения правых
        cln
        cls
        clz
        sub      d1, S1               ;Вычитание младших байтов
;проверка на ноль старшего байта
        clr      tmp
        rol      tmp

```

```

sub      d2, S2 ;Вычитание старшего байта с заемом
sub      d2, tmp ;Вычитание заема
brmi npr ;переход если результат отрицательный
nop
;Один шаг левый
clr      tmp
ldi      tmp, $04
out      PORTB, tmp      ;обнулить порт В
rcall    zader
clr      tmp
out      PORTB, tmp      ;обнулить порт В
nop
rjmp     con
npr:     nop
;Один шаг правый
clr      tmp
ldi      tmp, $02
out      PORTB, tmp      ;обнулить порт В
rcall    zader
clr      tmp
out      PORTB, tmp      ;обнулить порт В
nop
con:     nop
rjmp     nStart

```

;Подпрограмма назад, влево вперед

nazadlevo:

```

;назад
clr      tmp
ldi      tmp, $01
out      PORTB, tmp      ;обнулить порт В
nop
rcall    zader
nop
rcall    zader
clr      tmp
out      PORTB, tmp      ;обнулить порт В
;Один шаг левый
clr      tmp
ldi      tmp, $04
out      PORTB, tmp      ;обнулить порт В
nop
rcall    zader
nop
rcall    zader
clr      tmp
out      PORTB, tmp      ;обнулить порт В
ret

```

;Подпрограмма назад, вправо вперед

nazadpravo:

```

;назад
clr      tmp
ldi      tmp, $01
out      PORTB, tmp      ;обнулить порт В
nop
rcall    zader
nop
rcall    zader
clr      tmp

```

```

        out     PORTB, tmp      ;обнулить порт В
        ;Один шаг левый
        clr     tmp
        ldi     tmp, $02
        out     PORTB, tmp      ;обнулить порт В
        nop
        rcall   zader
        nop
        rcall   zader
        clr     tmp
        out     PORTB, tmp      ;обнулить порт В
        ret
;Настройка АЦП
adcn:    nop
        nop
        clr     tmp
awe:     nop
        cpi     tmp, $04
        brsh    ass
        out     ADMUX, tmp3     ;коммутация входов АЦП
        nop
        out     ADCSRA, tmp2    ;запуск АЦП на преобразование
        nop
        out     ADCSRB, tmp0    ;запуск АЦП на преобразование
;Сохранение данных АЦП
        in      tmp2, ADCL
        in      tmp3, ADCH
        mov     r2, tmp2
        mov     r3, tmp3
        mov     r4, r2
        mov     r5, r3
        mov     r6, r4
        mov     r7, r5
        inc     tmp
        rjmp    awe
;среднее арифметическое действие
ass:     nop
        clr     r8
        clc
        add     tmp2, r2
        adc     r9, r8
        add     tmp2, r4
        adc     r9, r8
        add     tmp2, r6
        adc     r9, r8
        clc
        lsr     r9
        ror     tmp2
        lsr     r9
        ror     tmp2
        clr     r9
        clc
        add     tmp3, r3
        adc     r9, r8
        add     tmp3, r5
        adc     r9, r8
        add     tmp3, r7
        adc     r9, r8
        clc
        lsr     r9

```

```

ror    tmp3
lsr    r9
ror    tmp3
nop
ret

```

;Подпрограмма задержки включения

```

zader: nop
      wdr
      clr    Mo
      clr    Yo
      ldi    Yo, $AF
      ldi    Mo, $04
      add    Mo, tmp3
dm:   nop
      ldi    Xo, $04
      add    Xo, tmp2
      nop
goq:  nop
      mov    Xo, tmp2
dv:   nop
      wdr
      nop
dx:   dec    Yo
      brpl   dx
      cln
      dec    Xo
      brpl   dv
      cln
      dec    Mo
      brne   dm
      ret

```

.EXIT