

```

;*****
;* PROGRAMMNAME:      ERSTELLER: Kolmit      DATUM: 04.04.2010 *
;* RGB-PWM 12VDC - 3 Channel with ISP      (Version A-01.01) *
;* Interne Oszillatorfrequenz Attiny 2313 = 4 MHz *
;*****
;
.NOLIST
.include "D:\Mikrocontroller\AVR-Programme\RGB-Control\10-04-04-RGB-PWM-12VDC-3-Channel
\3-channel-tn2313def.inc"
.LIST
.CSEG

;*****
;* Definition Register *
;*****

/* Arbeitsregister */
.def register_0 = r16      ;Arbeitsregister service
.def register_rt = r17     ;Arbeitsregister red
.def register_gn = r18     ;Arbeitsregister green
.def register_bl = r19     ;Arbeitsregister blue
.def register_mo = r20     ;Arbeitsregister mode
.def register_ti = r21     ;Arbeitsregister time

/* Verzögerungsschleife */
.def DELAY1 = r22         ;Verzögerungsregister 1
.def DELAY2 = r23         ;Verzögerungsregister 2
.def DELAY3 = r24         ;Verzögerungsregister 3
.def DELAY4 = r25         ;Verzögerungsregister 4

;*****
;* Definition Variablen und Port's *
;*****

/* Definition Schaltwerte */
.equ MIN = 0x00           ;Minimalwert_1 0 (8-Bit)0x00
.equ MAX = 0xff           ;Maximalwert_1 255 (8-Bit)0xff
.equ MIN_1 = 0x01         ;Minimalwert_2 5 (8-Bit)0x01
.equ MAX_1 = 0x96         ;Maximalwert_2 150 (8-Bit)0x96

/* Definition PWM-Kanäle */
;
;          OCR1AL      ;PWM-Kanal PB3 "ROT"
;
;          OCR1BL      ;PWM-Kanal PB4 "GRUEN"
;
;          OCR0A       ;PWM-Kanal PB2 "BLAU"

/* Definition Bedienungs-Taster */
;
;          PORTD0      ;Drucktaster PWM "RED MAX"
;
;          PORTD1      ;Drucktaster PWM "RED MIN"
;
;          PORTD2      ;Drucktaster PWM "GREEN MAX"
;
;          PORTD3      ;Drucktaster PWM "GREEN MIN"
;
;          PORTD4      ;Drucktaster PWM "BLUE MAX"
;
;          PORTD5      ;Drucktaster PWM "BLUE MIN"
;
;          PORTD6      ;Drucktaster PWM "TIME FAST"
;
;          PORTB0      ;Drucktaster PWM "TIME SLOW"

;*****
;* STACK-POINTER definieren *
;*****

ldi register_0,LOW(RAMEND)
out SPL,register_0      ;Stapel low einrichten

;*****
;* Ein- / Ausgangsports definieren *
;*****

/* DEFINITION PORT B */

```

```

ldi    register_0,0xE3
out    portb,register_0        ;Pull-Up-Widerstände einschalten

ldi    register_0,0x1C
out    ddrb,register_0        ;Port B definieren

/* DEFINITION PORT D */
ldi    register_0,0xFF
out    portd,register_0        ;Pull-Up-Widerstände einschalten

ldi    register_0,0x00
out    ddrd,register_0        ;Port D definieren

/* Defaultwert zuweisen */
ldi    register_0,MIN          ;Anfangswert PWM-Ausgang
out    OCR1A,register_0        ;Port B3 "ROT"
out    OCR1B,register_0        ;Port B4 "GRUEN"
out    OCR0A,register_0        ;Port B2 "BLAU"

;*****
;* PWM-Einstellungen Ausgang Port B2 / B3 / B4          *
;*****

ldi    register_0,0b10100001    ;set PWM-Mode (B2)
out    TCCR0A,register_0        ;Channel BLAU

ldi    register_0,0b00000010    ;set Prescaler 2 (500 Hz)
out    TCCR0B,register_0        ;set CS00 - CS02 (TCCR0B)

ldi    register_0,0b10100001    ;set PWM-Mode (B3 / B4)
out    TCCR1A,register_0        ;Channel ROT / GRUEN

ldi    register_0,0b00000010    ;set Prescaler 2 (500 Hz)
out    TCCR1B,register_0        ;set CS10 - CS12 (TCCR1B)

;*****
;* Sprungbefehl Hauptprogramm                          *
;*****

ldi    register_mo,0x01         ;Programmwahl "Einzelfarbmodus"
ldi    register_ti,0x96         ;Zeitvorwahl Farbwechsel
rjmp   main                    ;Beginn Hauptprogramm

;*****
;* HAUPTPROGRAMM                                      *
;*****
main:

/* Programmanwahl Einzelfarbmodus / Automatikfarbwechsler */

sbis   PIND,0                   ;Abfrage Schalter RT MAX
rjmp   mode_colour              ;Programmablauf Einzelfarbmodus
sbis   PIND,1                   ;Abfrage Schalter RT MAX
rjmp   mode_colour              ;Programmablauf Einzelfarbmodus
sbis   PIND,2                   ;Abfrage Schalter RT MAX
rjmp   mode_colour              ;Programmablauf Einzelfarbmodus
sbis   PIND,3                   ;Abfrage Schalter RT MAX
rjmp   mode_colour              ;Programmablauf Einzelfarbmodus
sbis   PIND,4                   ;Abfrage Schalter RT MAX
rjmp   mode_colour              ;Programmablauf Einzelfarbmodus
sbis   PIND,5                   ;Abfrage Schalter RT MAX
rjmp   mode_colour              ;Programmablauf Einzelfarbmodus
sbis   PIND,6                   ;Abfrage Schalter FARBEWECHSEL
rjmp   mode_time                ;Programmablauf Farbwechsler

```

```

    sbis    PINB,0                ;Abfrage Schalter FARBWECHSEL
    rjmp   mode_time             ;Programmablauf Farbwechsler
    nop
    rjmp   mode

mode_colour:
    cpi    register_mo,0x01      ;Wurde von Farbwechselmodus
    breq   mode_colour_         ;nach Einzelfarbmodus gewechselt,
    ldi    register_0,MIN       ;werden alle Ausgänge auf Null
    out    OCR1AL,register_0    ;gesetzt.
    out    OCR1BL,register_0
    out    OCR0A,register_0

mode_colour_:
    ldi    register_mo,0x01      ;Programmwahl "Einzelfarbmodus"
    rjmp   mode

mode_time:
    ldi    register_mo,0x02      ;Wurde von Einzelfarbmodus nach
    brge   mode_time_          ;Wechselmodus gewechselt, werden
    ldi    register_0,MIN       ;alle Ausgänge auf Null gesetzt.
    out    OCR1AL,register_0
    out    OCR1BL,register_0
    out    OCR0A,register_0

mode_time_:
    ldi    register_mo,0x02      ;Programmwahl "Farbwechselmodus"

mode:
    cpi    register_mo,0x01      ;Ausführung "Einzelfarbmodus"
    breq   colour
    nop
    rjmp   time                 ;Ausführung "Farbwechselmodus"
                                ;Wird beim ersten Durchlauf kein Modus vorgewählt,
                                ;so wird automatisch der Einzelfarbmodus eingestellt

t!
colour:
;*****
;* UNTERPROGRAMM EINZELFARBMODUS "COLOUR" *
;*****

;*****
;* PROGRAMMDURCHLAUF EINZELFARBMODUS *
;*****

    sbis    PIND,0                ;Abfrage Schalter RT MAX
    call   sw_rt_max              ;Programmablauf Einzelfarbmodus
    nop
    sbis    PIND,1                ;Abfrage Schalter RT MIN
    call   sw_rt_min             ;Programmablauf Einzelfarbmodus
    nop
    sbis    PIND,2                ;Abfrage Schalter GN MAX
    call   sw_gn_max            ;Programmablauf Einzelfarbmodus
    nop
    sbis    PIND,3                ;Abfrage Schalter GN MIN
    call   sw_gn_min            ;Programmablauf Einzelfarbmodus
    nop
    sbis    PIND,4                ;Abfrage Schalter BL MAX
    call   sw_bl_max            ;Programmablauf Einzelfarbmodus
    nop
    sbis    PIND,5                ;Abfrage Schalter BL MIN
    call   sw_bl_min            ;Programmablauf Einzelfarbmodus
    nop
    rjmp   main

;*****
;* EINZELFARBMODUS ROT *
;*****

```

```
    /* PWM-Ausgang erhöhen */
sw_rt_max:
    in     register_rt,OCR1AL      ;PWM LED ROT
    ldi   register_0,MAX
    cpse  register_rt,register_0  ;Grenzwert MAX
    inc   register_rt
    out   OCR1AL,register_rt
    rjmp  sw_rt_max_signal

    /* PWM-Ausgang reduzieren */
sw_rt_min:
    in     register_rt,OCR1AL      ;PWM LED ROT
    ldi   register_0,MIN
    cpse  register_rt,register_0  ;Grenzwert MIN
    dec   register_rt
    out   OCR1AL,register_rt
    rjmp  sw_rt_min_signal

    /* PWM-Ausgang Maximalwert erreicht (Farbe blinkend)*/
sw_rt_max_signal:
    in     register_rt,OCR1AL      ;PWM LED ROT
    cpi   register_rt,MAX         ;Grenzwert MAX
    brlo  sw_rt_end
    ldi   register_rt,MIN         ;PWM LED ROT
    out   OCR1AL,register_rt
    call  DLY
    nop
    ldi   register_rt,MAX         ;PWM LED ROT
    out   OCR1AL,register_rt
    rjmp  sw_rt_end

    /* PWM-Ausgang Minimalwert erreicht (Farbe blinkend)*/
sw_rt_min_signal:
    in     register_rt,OCR1AL      ;PWM LED ROT
    cpi   register_rt,MIN         ;Grenzwert MIN
    brne  sw_rt_end
    ldi   register_rt,MAX         ;PWM LED ROT
    out   OCR1AL,register_rt
    call  DLY
    nop
    ldi   register_rt,MIN
    out   OCR1AL,register_rt

sw_rt_end:
    rjmp  DLY

;*****
;*  EINZELFARBMODUS GRUEN                                     *
;*****

    /* PWM-Ausgang erhöhen */

sw_gn_max:
    in     register_gn,OCR1BL      ;PWM LED GRUEN
    ldi   register_0,MAX
    cpse  register_gn,register_0  ;Grenzwert MAX
    inc   register_gn
    out   OCR1BL,register_gn
    rjmp  sw_gn_max_signal

    /* PWM-Ausgang reduzieren */

sw_gn_min:
    in     register_gn,OCR1BL      ;PWM LED GRUEN
    ldi   register_0,MIN
    cpse  register_gn,register_0  ;Grenzwert MIN
    dec   register_gn
    out   OCR1BL,register_gn
```

```
    rjmp     sw_gn_min_signal

    /* PWM-Ausgang Maximalwert erreicht (Farbe blinkend)*/
sw_gn_max_signal:
    in      register_gn,OCR1BL      ;PWM LED GRUEN
    cpi    register_gn,MAX         ;Grenzwert MAX
    brlo   sw_gn_end
    ldi    register_gn,MIN         ;PWM LED GRUEN
    out    OCR1BL,register_gn
    call   DLY
    nop
    ldi    register_gn,MAX
    out    OCR1BL,register_gn
    rjmp   sw_gn_end

    /* PWM-Ausgang Minimalwert erreicht (Farbe blinkend)*/
sw_gn_min_signal:
    in      register_gn,OCR1BL      ;PWM LED GRUEN
    cpi    register_gn,MIN         ;Grenzwert MIN
    brne   sw_gn_end
    ldi    register_gn,MAX         ;PWM LED GRUEN
    out    OCR1BL,register_gn
    call   DLY
    nop
    ldi    register_gn,MIN
    out    OCR1BL,register_gn

sw_gn_end:
    rjmp   DLY

;*****
;*  EINZELFARBMODUS BLAU                *
;*****

    /* PWM-Ausgang erhöhen */

sw_bl_max:
    in      register_bl,OCR0A      ;PWM LED BLAU
    ldi    register_0,MAX
    cpse   register_bl,register_0  ;Grenzwert MAX
    inc    register_bl
    out    OCR0A,register_bl
    rjmp   sw_bl_max_signal

    /* PWM-Ausgang reduzieren */

sw_bl_min:
    in      register_bl,OCR0A      ;PWM LED BLAU
    ldi    register_0,MIN
    cpse   register_bl,register_0  ;Grenzwert MIN
    dec    register_bl
    out    OCR0A,register_bl
    rjmp   sw_bl_min_signal

    /* PWM-Ausgang Maximalwert erreicht (Farbe blinkend)*/
sw_bl_max_signal:
    in      register_bl,OCR0A      ;PWM LED BLAU
    cpi    register_bl,MAX         ;Grenzwert MAX
    brlo   sw_bl_end
    ldi    register_bl,MIN         ;PWM LED BLAU
    out    OCR0A,register_bl
    call   DLY
    nop
    ldi    register_bl,MAX
    out    OCR0A,register_bl
    rjmp   sw_bl_end

    /* PWM-Ausgang Minimalwert erreicht (Farbe blinkend)*/
```

```

sw_bl_min_signal:
    in     register_bl,OCR0A        ;PWM LED BLAU
    cpi   register_bl,MIN         ;Grenzwert MIN
    brne  sw_bl_end
    ldi   register_bl,MAX         ;PWM LED BLAU
    out   OCR0A,register_bl
    call  DLY
    nop
    ldi   register_bl,MIN
    out   OCR0A,register_bl

sw_bl_end:
    rjmp  DLY

time:

;*****
;*  UNTERPROGRAMM FARBWECHELBMODUS  *
;*****

;*****
;*  PROGRAMMDURCHLAUF WECHSELFARBMODUS  *
;*****
sw_ti_flash_start:

    /* Abfrage SLOW / FAST-Taster */

    sbis  PINB,0                 ;Abfrage Schalter ZEIT MIN
    rjmp  sw_ti_slow            ;Programmablauf Farbwechselmodus

    sbis  PIND,6                 ;Abfrage Schalter ZEIT MAX
    rjmp  sw_ti_fast            ;Programmablauf Einzelfarbmodus

    rjmp  sw_ti_flash_1

    /* Farbwechselzeit erhöhen */

sw_ti_slow:
    ldi   register_0,MAX-0x02
    cpse  register_ti,register_0
    inc   register_ti
    rjmp  sw_ti_slow_max

    /* Farbwechselzeit reduzieren */

sw_ti_fast:
    ldi   register_0,MIN+0x02
    cpse  register_ti,register_0
    dec   register_ti
    rjmp  sw_ti_slow_min

sw_ti_slow_max:
    cpi   register_ti,MAX-0x02
    brlo  sw_ti_flash
    ldi   register_0,MIN
    out   OCR1AL,register_0
    out   OCR1BL,register_0
    out   OCR0A,register_0
    call  DLY
    nop
    ldi   register_0,MAX
    out   OCR1AL,register_0
    out   OCR1BL,register_0
    out   OCR0A,register_0
    rjmp  sw_ti_flash

sw_ti_slow_min:
    cpi   register_ti,MIN+0x02

```

```

brne    sw_ti_flash
ldi     register_0,MAX
out     OCR1AL,register_0
out     OCR1BL,register_0
out     OCR0A,register_0
call    DLY
nop
ldi     register_0,MIN
out     OCR1AL,register_0
out     OCR1BL,register_0
out     OCR0A,register_0
rjmp    sw_ti_flash

sw_ti_flash:
call    DLY
nop
rjmp    sw_ti_flash_start

sw_ti_flash_1:

;*****
;* FARBEWECHSELMODUS FARBEWAHL *
;*****

flash_1_1:
cpi     register_mo,0x03      ;Phase I - blau fallend
brsh    flash_1_2
rjmp    sw_ti_phase_1_1      ;Phase I - rot steigend

flash_1_2:
cpi     register_mo,0x04      ;Phase I - blau fallend
brsh    flash_2_1
rjmp    sw_ti_phase_1_2      ;Phase I - rot steigend

flash_2_1:
cpi     register_mo,0x05      ;Phase II - rot fallend
brsh    flash_2_2
rjmp    sw_ti_phase_2_1      ;Phase II - gruen steigend

flash_2_2:
cpi     register_mo,0x06      ;Phase II - rot fallend
brsh    flash_3_1
rjmp    sw_ti_phase_2_2      ;Phase II - gruen steigend

flash_3_1:
cpi     register_mo,0x07      ;Phase III - gruen fallend
brsh    flash_3_2
rjmp    sw_ti_phase_3_1      ;Phase III - blau steigend

flash_3_2:
cpi     register_mo,0x08      ;Phase III - gruen fallend
brsh    DLY_1
rjmp    sw_ti_phase_3_2      ;Phase III - blau steigend

;*****
;* FARBEWECHSELMODUS PHASE I (ALLE FARBEN) *
;*****

sw_ti_phase_1_1:
ldi     register_0,MIN
ldi     register_rt,MIN
ldi     register_gn,MIN
out     OCR1AL,register_0      ;Vorbelegung Farbe "ROT"
out     OCR1BL,register_0      ;Vorbelegung Farbe "GRUEN"
ldi     register_0,MAX_1+0x01
ldi     register_bl,MAX_1+0x01
out     OCR0A,register_0      ;Vorbelegung Farbe "BLAU"
inc     register_mo

```

```

sw_ti_phase_1_2:
    in      register_rt,OCR1AL      ;Farbwechsel Phase I
    cpi    register_rt,MAX_1      ;Abfrage auf Maxwert
    breq   sw_ti_phase_1_end
    nop
    inc    register_rt            ;LED ROT steigend
    dec    register_bl            ;LED BLAU fallend
    out    OCR1AL,register_rt      ;Ausgabe PWM LED ROT
    out    OCR0A,register_bl      ;Ausgabe PWM LED BLAU
    rjmp   DLY_1

sw_ti_phase_1_end:                ;Farbwechsel Phase I Ende
    inc    register_mo
    rjmp   DLY_1

;*****
;* FARBECHSELMODUS PHASE II (ALLE FARBE) *
;*****

sw_ti_phase_2_1:
    ldi    register_0,MAX_1+0x01
    out    OCR1AL,register_0      ;Vorbelegung Farbe "ROT"
    ldi    register_0,MIN
    out    OCR1BL,register_0      ;Vorbelegung Farbe "GRUEN"
    out    OCR0A,register_0      ;Vorbelegung Farbe "BLAU"
    inc    register_mo

sw_ti_phase_2_2:
    in      register_gn,OCR1BL     ;Farbwechsel Phase II
    cpi    register_gn,MAX_1      ;Abfrage auf Maxwert
    breq   sw_ti_phase_2_end
    nop
    inc    register_gn            ;LED GRUEN steigend
    dec    register_rt            ;LED ROT fallend
    out    OCR1AL,register_rt      ;Ausgabe PWM LED ROT
    out    OCR1BL,register_gn      ;Ausgabe PWM LED GRUEN
    rjmp   DLY_1

sw_ti_phase_2_end:                ;Farbwechsel Phase II Ende
    inc    register_mo
    rjmp   DLY_1

;*****
;* FARBECHSELMODUS PHASE III (ALLE FARBE) *
;*****

sw_ti_phase_3_1:
    ldi    register_0,MIN
    out    OCR1AL,register_0      ;Vorbelegung Farbe "ROT"
    ldi    register_0,MAX_1+0x01
    out    OCR1BL,register_0      ;Vorbelegung Farbe "GRUEN"
    ldi    register_0,MIN
    out    OCR0A,register_0      ;Vorbelegung Farbe "BLAU"
    inc    register_mo

sw_ti_phase_3_2:
    in      register_bl,OCR0A     ;Farbwechsel Phase III
    cpi    register_bl,MAX_1      ;Abfrage auf Maxwert
    breq   sw_ti_phase_3_end
    nop
    inc    register_bl            ;LED BLAU steigend
    dec    register_gn            ;LED GRUEN fallend
    out    OCR0A,register_bl      ;Ausgabe PWM LED BLAU
    out    OCR1BL,register_gn      ;Ausgabe PWM LED GRUEN
    rjmp   DLY_1

sw_ti_phase_3_end:                ;Farbwechsel Phase III Ende

```



```

    ldi    register_mo,0x02          ;Farbwechselmodus Startadresse
    rjmp   main

;*****
;* VERZÖGERUNGSSCHLEIFEN *
;*****

    /* VERZÖGERUNGSSCHLEIFE 2*/

DLY_1:

    ldi    register_0,0xff          ;variable Verzögerung 1
    mov    Delay3,register_0
    mov    Delay4,register_ti

DLY_2:                                ;variable Verzögerung 2

;    dec    Delay1
;    brne   DLY_2
    dec    Delay2
    brne   DLY_2
    dec    Delay3
    brne   DLY_2
    dec    Delay4
    brne   DLY_2

    rjmp   main

    /* VERZÖGERUNGSSCHLEIFE 1*/

DLY:

    ldi    register_0,0xff          ;variable Verzögerung 1
    mov    Delay2,register_0

DLY_0:                                ;Verzögerungsschleife 1
    dec    Delay1
    brne   DLY_0
    dec    Delay2
    brne   DLY_0

    ret

;*****
;* PROGRAMMENDE - zurück zur Hauptschleife *
;*****

end:
    rjmp   main

.EXIT

```