X Timers ADC AVR

E M

K K , *el*21045 M Δ , *el*21170

29 O 2024

0.1 Z 3.1

Σ cseg duty cycle DC_VALUE. Θ PORTB PORTD TCCR1A, TCCR1B TMR1A Fast PWM, 8 bit, non-inverting output N = 256, BOTTOM = 0 TOP = 0x00ff = 255. Γ PD3, PD4 PCINT19 (PD3) PCINT20 (PD4) PCMSK2, o Pin Change Interrupt 2 PCICR. A i 6 (6 , duty cycle 50%) Z i- . T , . Σ duty cycle Z 10ms . H pc2isr PD3 PD4. E PIND PD3, PD4 . Σ , cur_PIND AND (NOT prev_PIND) bit PIND 1 0. A PD3 i Z 1 , PD4 , i Z 1 .

```
;
; Ex3_1.asm
;
; Created: 10/25/2024 5:06:29 PM
; Author : User
;

.include "m328PBdef.inc"

.org 0x0
        rjmp reset

.org 0xA
        rjmp pc2isr

.def temp = r16
.def DC_VALUE = r17
.def DC_INC = r18 ; DC_INC = 1 -> DC_VALUE increasing, DC_INC = 0 -> DC_VALUE decreasing
.def i = r19
.def prev_PIND = r20
.def cur_PIND = r21

```

```
DUTY: .DB 5, 26, 46, 66, 87, 107, 128, 148, 168, 189, 209, 230, 250 ; 255*(2+8k)/100, k = 0 ... 12. Initial k = 6.
.equ DUTY_LAST = 12
.equ DUTY_START = 6

pc2isr:
        push temp
        in temp, SREG
        push temp
        in cur_PIND, PIND                       ; Get the current set PIND bits
        mov temp, prev_PIND
        com temp                                ; Get the previous clear PIND bits
        and temp, cur_PIND                      ; Get the 1 bit that was just set and caused the PCINT2 interrupt
        sbrc temp, 3                            ; Skip next instruction if PD3 was not pressed
        rjmp increase
        sbrc temp, 4                            ; Skip next instruction if PD4 was not pressed
        rjmp decrease
        rjmp end
increase:
        cpi i, DUTY_LAST
        breq end
        inc i
        adiw Z, 1
        rjmp end
decrease:
        cpi i, 0
        breq end
        dec i
        sbiw Z, 1
end:
        mov prev_PIND, cur_PIND
        pop temp
        out SREG, temp
        pop temp
        rjmp reset

reset:
        ; Init stack pointer
        ldi temp, high(RAMEND)
        out SPH, temp
        ldi temp, low(RAMEND)
        out SPL, temp

        ; Set PORTB as output
        ser temp
        out DDRB, temp

        ; Set PORTD as input
        clr temp
        out DDRD, temp
```

```

    ; Fast PWM, 8 bit, non-inverting output, N = 256. BOTTOM = 0, TOP = 0x00ff = 255
    ldi temp, (1<<WGM10) | (1<<COM1A1)
    sts TCCR1A, temp
    ldi temp, (1<<WGM12) | (1<<CS12)
    sts TCCR1B, temp

    ; Enable PCINT19 (PD3), PCINT20 (PD4) interrupts
    ldi temp, (1<<PCINT19) | (1<<PCINT20)
    sts PCMSK2, temp

    ; Enable Pin Change Interrupt 2: PCINT[23:16]
    ldi temp, (1<<PCIE2)
    sts PCICR, temp

    ; Set previous state of PIND = 0
    ldi prev_PIND, 0

    ; Initialize i to duty cycle starting position: 6 (50%)
    ldi i, DUTY_START

    ; Load the starting address of the duty cycle value into Z
    LDI ZH, HIGH(2*DUTY+DUTY_START)
    LDI ZL, LOW(2*DUTY+DUTY_START)

    sei

; Replace with your application code
start:
    lpm DC_VALUE, Z
    sts OCR1AL, DC_VALUE

    ldi r24, LOW(10*16)        ;
    ldi r25, HIGH(10*16)       ; Set delay (10ms * 16)
    rcall delay_mS                      ; Delay for 10ms

    rjmp start

; delay of 1000*F1+6 cycles (almost equal to 1000*F1 cycles)
delay_mS:
; total delay of next 4 instruction group = 1+(249*4-1) = 996 cycles
    ldi r23, 249                    ; (1 cycle)
loop_inn:
    dec r23                                 ; 1 cycle
    nop                                         ; 1 cycle
    brne loop_inn                       ; 1 or 2 cycles

    sbiw r24, 1                             ; 2 cycles
    brne delay_mS                       ; 1 or 2 cycles

    ret                                         ; 4 cycles
```

0.2	Z	3.2
0.3	Z	3.3