;File: pwm41w16.asm

; Author: Eckart Hartmann Date:15/10/2003

; Development progress: Pwm834.df

;

;PwmW16==========Write 16 bit:

;C Function prototype: char PwmW16(char cCh, int iVal).

;Description of Function: Write 16 bit channel value.

;User interface: Set parameter1 to 0 to write PWM0H/L or non zero to

; write PWM1H/L. Put the value to write in parameter2.

; See Description for more details.

; Call PwmW16. If in 16 bit mode or mode 0, writes value and

; returns 1 else just returns 0.

;Robustness: If selected mode is not 16 bit, wrong timing results on both

; channels. Changes only become effective at the end of the

; current PWM cycle.

;Side effects: Overwrites a, P, c.

;

NAME PWMW16

$NOMOD51

$IC(..kei841.inc) ; Parameter passing registers for Keil .

$IC(..kei841.dat) ; SFR definition for Keil .

;

public \_PwmW16

?PR?\_PwmW16?PWMW16 SEGMENT CODE

RSEG ?PR?\_PwmW16?PWMW16

;

\_PwmW16:

mov a,PWMCON ;if(16 bit mode)

anl a,#070h

jz Pw6Do

cjne a,#10h,Pw6Q3

sjmp Pw6Do

Pw6Q3: cjne a,#30h,Pw6Q4

sjmp Pw6Do

Pw6Q4: cjne a,#40h,Pw6Q6

sjmp Pw6Do

Pw6Q6: cjne a,#60h,Pw6R0 ; {

Pw6Do: mov a,cP1l ; if(cCh==0)

jnz Pw6C

mov PWM0L,iP2lc ; PWM0L/H = iP2l/h;

mov PWM0H,iP2hc

sjmp Pw6R1

Pw6C: mov PWM1L,iP2lc ; else PWM1L/H = iP2l/h;

mov PWM1H,iP2hc

Pw6R1: mov cP1l,#1 ; return 1;

sjmp Pw6R ; }

Pw6R0: mov cP1l,#0 ;else return 0;

Pw6R: ret

;

;Function End==========================================================Function End

END

;