Computer Interfacing Segon Parcial 19/11/2018

Nom i Cognoms: _____

1) Configurem la unitat CCP1 en mode Compare amb els valors (**Fosc=8MHz**): CCP1CON=0x02, CCPTMRS0=0x02, CCPR1=0x0336, T5CON=0x73, TMR5GE=0 i CCP1TRIS=0.

Quína freqüència del senyal esperem tenir a la sortida del pin CCP1? *(1,5punts)*

2) Volem tenir una interrupció d'alta prioritat del Timer0 cada 25ms i comptar les interrupcions que van arribant. Configura tot el necessari si tenim un **Fosc=20MHz**. (*2,5punts)*

```
volatile int Count_INT;

void main(void)
{
    Count_INT=0;
```

```
void interrupt RSI_High(void)
{

        Count_INT++;

}
```

3) Si s'activessin en el mateix moment 4 flags d'interrupció (per ex. INT1IF, TMR1IF, ADIF, CCP3IF), i estan les 4 habilitades i configurades com a baixa prioritat, què determinarà en quin ordre les atendrem? *(1punt)*

REGISTER 14-3: CCPTMRS0: PWM TIMER SELECTION CONTROL REGISTER 0

R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
C3TSEL<1:0>		—	C2TSEL<1:0>		—	C1TSEL<1:0>	
bit 7							bit 0

bit 1-0 **C1TSEL<1:0>:** CCP1 Timer Selection bits
00 = CCP1 – Capture/Compare modes use Timer1, PWM modes use Timer2
01 = CCP1 – Capture/Compare modes use Timer3, PWM modes use Timer4
10 = CCP1 – Capture/Compare modes use Timer5, PWM modes use Timer6
11 = Reserved

REGISTER 14-1: CCPxCON: STANDARD CCPx CONTROL REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DCxB<1:0>		CCPxM<3:0>			
bit 7							bit 0

bit 7-6 **Unused**

bit 5-4 **DCxB<1:0>:** PWM Duty Cycle Least Significant bits
Capture mode:
Unused
Compare mode:
Unused
PWM mode:
These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPRxL.

bit 3-0 **CCPxM<3:0>:** ECCPx Mode Select bits
0000 = Capture/Compare/PWM off (resets the module)
0001 = Reserved
0010 = Compare mode: toggle output on match
0011 = Reserved

0100 = Capture mode: every falling edge
0101 = Capture mode: every rising edge
0110 = Capture mode: every 4th rising edge
0111 = Capture mode: every 16th rising edge

1000 = Compare mode: set output on compare match (CCPx pin is set, CCPxIF is set)
1001 = Compare mode: clear output on compare match (CCPx pin is cleared, CCPxIF is set)
1010 = Compare mode: generate software interrupt on compare match (CCPx pin is unaffected, CCPxIF is set)
1011 = Compare mode: Special Event Trigger (CCPx pin is unaffected, CCPxIF is set)
TimerX (selected by CxTSEL bits) is reset
ADON is set, starting A/D conversion if A/D module is enabled(1)
11xx =: PWM mode

7	6	5	4	3	2	1	0
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
1	1	1	1	1	1	1	1

TMR0ON: *Timer0 on/off control bit*
0 = stops Timer0
1 = Enables Timer0
T08BIT: *Timer0 8-bit/16-bit control bit*
0 = Timer0 is configured as a 16-bit timer
1 = Timer0 is configured as an 8-bit timer
T0CS: *Timer0 clock source select*
0 = Instruction cycle clock
1 = Transition on T0CKI pin
T0SE: *Timer0 source edge select bit*
0 = Increment on falling edge transition on T0CKI pin
1 = Increment on rising edge transition on T0CKI pin
PSA: *Timer0 prescaler assignment bit*
0 = Timer0 prescaler is assigned . Timer0 clock input comes from prescaler
1 = Timer0 prescaler is not assigned. Timer0 clock input bypasses prescaler
T0PS2:T0PS0: *Timer0 prescaler select bits*
000 = 1:2 prescaler value
001 = 1:4 prescaler value
010 = 1:8 prescaler value
011 = 1:16 prescaler value
100 = 1:32 prescaler value
101 = 1:64 prescaler value
110 = 1:128 prescaler value
111 = 1:256 prescaler value

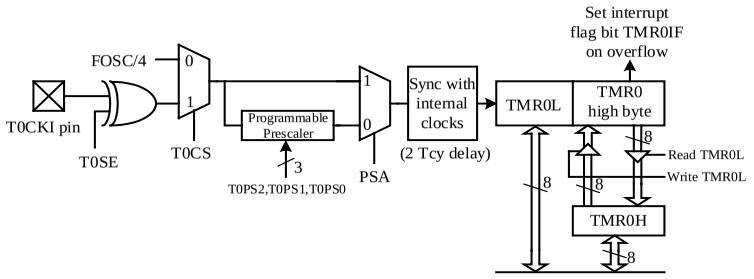
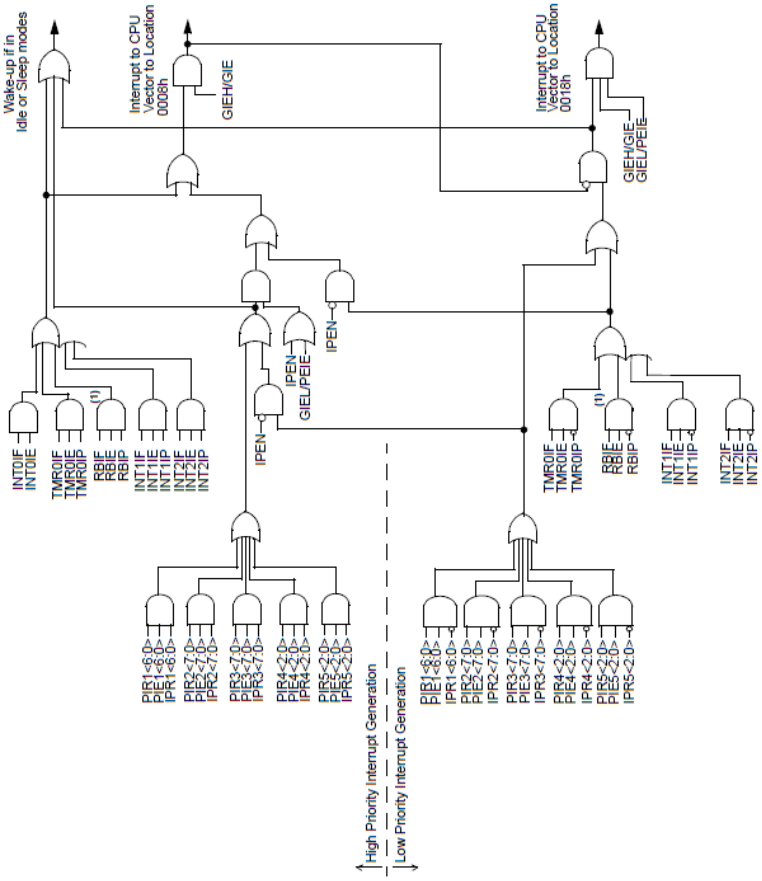


Figure 8.2 T0CON register (reprint with permission of Microchip)

TABLE 9-1: REGISTERS ASSOCIATED WITH INTERRUPTS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	INT2IP	RBIP
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP
IPR2	OSCFIP	C1IP	C2IP	EEIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	CTMUIP	TMR5GIP	TMR3GIP	TMR1GIP
IPR4	—	—	—	—	—	CCP5IP	CCP4IP	CCP3IP
IPR5	—	—	—	—	—	TMR6IP	TMR5IP	TMR4IP
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
PIE2	OSCFIE	C1IE	C2IE	EEIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	CTMUIE	TMR5GIE	TMR3GIE	TMR1GIE
PIE4	—	—	—	—	—	CCP5IE	CCP4IE	CCP3IE
PIE5	—	—	—	—	—	TMR6IE	TMR5IE	TMR4IE
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
PIR2	OSCFIF	C1IF	C2IF	EEIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	CTMUIF	TMR5GIF	TMR3GIF	TMR1GIF
PIR4	—	—	—	—	—	CCP5IF	CCP4IF	CCP3IF
PIR5	—	—	—	—	—	TMR6IF	TMR5IF	TMR4IF
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
RCON	IPEN	SBOREN	—	RI	TO	PD	POR	BOR



Nom i Cognoms: _____

4) Volem utilitzar la unitat de Capture de dos CCPs per saber l'amplada d'un pols. La CCP1 estarà configurada per trobar el flanc de pujada i la CCP2 per trobar el flanc de baixada. Fosc=8MHz, el timer associat utilitza Fosc com a entrada i PRE=1. Després de l'arribada dels dos flancs tenim que CCPR1=17 i CCPR2=17, què podem saber respecte l'amplada del pols? *(2 punts)*

5) Se'ns demana, a partir d'un Fosc=12MHz, generar un PWM de freqüència 12,5 KHz. Quínes configuracions ens permetran tenir com a mínim 6 bits de resolució? *(2 punts)*

6) Volem utilitzar el timer 4 com a base de temps, tot generant una interrupció periòdica basada en TMR4IF. Si Fosc=8MHz, quína serà la freqüència mínima amb que ens poden arribar les interrupcions? *(1 punt)*

REGISTER 12-1: TXCON: TIMER1/3/5 CONTROL REGISTER

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/0	R/W-0/u
TMRxCS<1:0>		TxCKPS<1:0>		TxSOSCEN	$\overline{\text{TxSYNC}}$	TxRD16	TMRxON
bit 7							bit 0

bit 7-6 **TMRxCS<1:0>:** Timer1/3/5 Clock Source Select bits
11 = Reserved. Do not use.
10 = Timer1/3/5 clock source is pin or oscillator:
If TxSOSCEN = 0:
External clock from TxCKI pin (on the rising edge)
If TxSOSCEN = 1:
Crystal oscillator on SOSCI/SOSCO pins
01 = Timer1/3/5 clock source is system clock (FOSC)
00 = Timer1/3/5 clock source is instruction clock (FOSC/4)

bit 5-4 **TxCKPS<1:0>:** Timer1/3/5 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **TxSOSCEN:** Secondary Oscillator Enable Control bit
1 = Dedicated Secondary oscillator circuit enabled
0 = Dedicated Secondary oscillator circuit disabled

bit 2 **$\overline{\text{TxSYNC}}$:** Timer1/3/5 External Clock Input Synchronization Control bit
TMRxCS<1:0> = 1X
1 = Do not synchronize external clock input
0 = Synchronize external clock input with system clock (FOSC)

TMRxCS<1:0> = 0X
This bit is ignored. Timer1/3/5 uses the internal clock when TMRxCS<1:0> = 1x.

bit 1 **TxRD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer1/3/5 in one 16-bit operation
0 = Enables register read/write of Timer1/3/5 in two 8-bit operation

bit 0 **TMRxON:** Timer1/3/5 On bit
1 = Enables Timer1/3/5
0 = Stops Timer1/3/5
Clears Timer1/3/5 Gate flip-flop

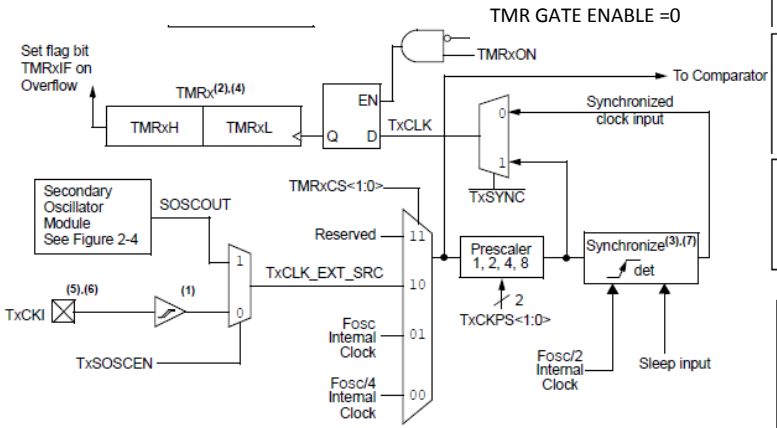
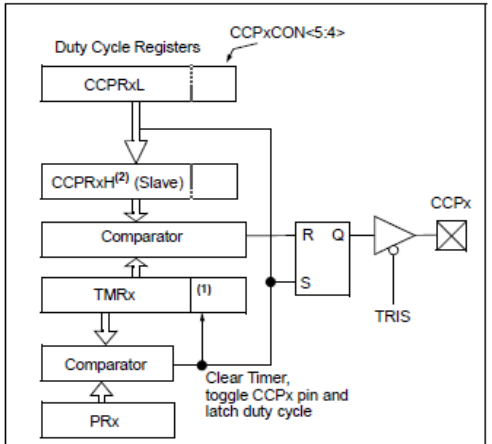


FIGURE 14-4: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: The 8-bit timer TMRx register is concatenated with the 2-bit internal system clock (FOSC), or 2 bits of the prescaler, to create the 10-bit time base.
2: In PWM mode, CCPRxH is a read-only register.

$$PWM\ Period = [(PRx) + 1] \bullet 4 \bullet Tosc \bullet (TMRx\ Prescale\ Value)$$

Note 1: TOSC = 1/FOSC

$$Pulse\ Width = (CCPRxL{:}CCPxCON{<}5{:}4{>}) \bullet Tosc \bullet (TMRx\ Prescale\ Value)$$

$$Resolution = \frac{log[4(PRx + 1)]}{log(2)}\ bits$$

FIGURE 14-1: CAPTURE MODE OPERATION BLOCK DIAGRAM

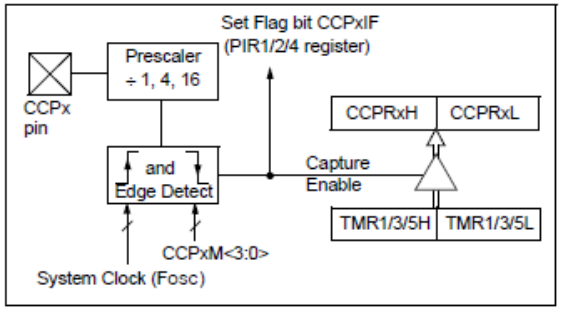
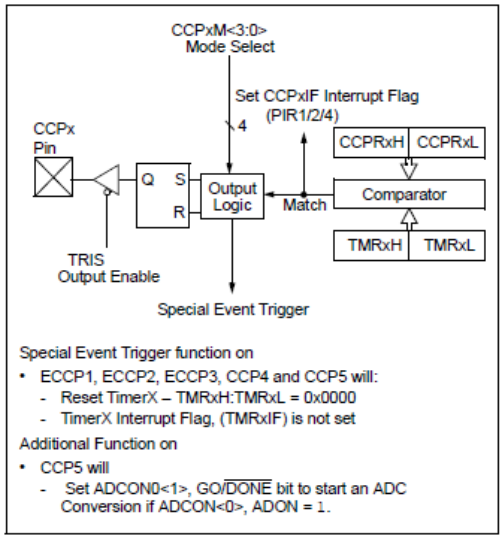


FIGURE 14-2: COMPARE MODE OPERATION BLOCK DIAGRAM



Special Event Trigger function on
- ECCP1, ECCP2, ECCP3, CCP4 and CCP5 will:
  - Reset TimerX – TMRxH:TMRxL = 0x0000
  - TimerX Interrupt Flag, (TMRxIF) is not set

Additional Function on
- CCP5 will
  - Set ADCON0<1>, GO/$\overline{\text{DONE}}$ bit to start an ADC Conversion if ADCON<0>, ADON = 1.

FIGURE 13-1: TIMER2/4/6 BLOCK DIAGRAM