Nom i Cognoms:_____

Contesteu en aquest mateix full en l'espai reservat de cada pregunta. Raoneu totes les respostes. Les respostes que no vagin acompanyades d'una mínima explicació no s'admetran.

1. La següent funció encén un led col.locat al pin CCP2 del PIC (amb Fosc=8MHz) durant exactament 10ms.

```
void EncenLED10ms ( void )
{
          T1CON=0x95;
          CCPR2=10000;
          T3CCP1=0;
          T3CCP2=0;
          TRISCbits.CCP2=0;
          CCP2CON=0x09;
}
```

Modifiqueu el que calgui perquè el LED s'encengui durant 120ms (2 punts). Justifiqueu la resposta.

Veiem que el prescaler configurat a T1CON és de 1:2 (al Timer 1 entra (fosc/4)/2-> 1us) i que per tant compta fins a 10000 amb el CCPR2 (10000us=10ms).

Com que 120000 no cap en 16 bits del CCPR2, tenim dues opcions: o posar el prescaler a 1:4 (T1CON = A5) i posar 60000 al CCPR2, o posar el prescaler a 1:8 (T1CON=B5) i posar 30000 al CCPR2. Les dues són vàlides.

 Amb 8MHz de CLK del sistema, podem usar la unitat CCP1 per tenir una interrupció CCP1IF cada 1 segon (mode 1010 del CCP)? Justifiqueu la resposta. (1 punt)

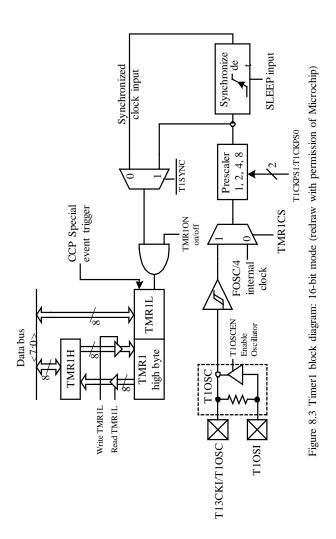
No, perquè hauríem de comptar 2.000.000 de tics. Aquest valor no cap en 16 bits del CCPR1 i el prescaler més gran que podem usar és de 8 que implicaria posar un valor de 250000 que tampoc cap en 16 bits.

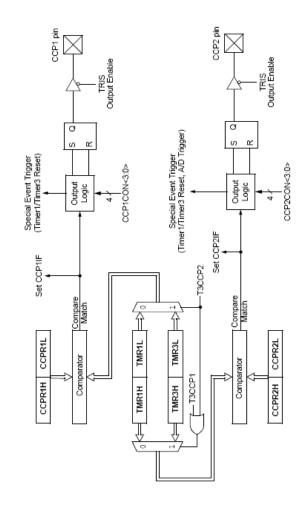
3. Emprant el mode Toggle (mode CCP = 0010) quina és la freqüència més baixa que podem generar al pin CCP1 amb el xip emprat a laboratori (oscil.lador a 8MHz)? Justifica la resposta (2 punts).

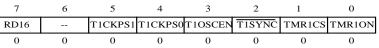
Si posem el màxim prescaler (1:8) i carreguem al registre CCP1R el màxim valor 65535, ens dóna una freqüència de:

```
(Fosc/4) \cdot (1/8) \cdot (1/65536) = 3,814 \text{ Hz}.
```

Com que emprem el mode toggle (cada cop canvia l'estat 1/0), la freqüència de sortida serà la meitat, per tant 1,9 Hz.







RD16: 16-bit read/write mode enable bit

0 = Enables read/write of Timer1 in two 8-bit operations

1 = Enable read/write of Timer1 in 16-bit operation

T1CKPS1:T1CKPS0: Timer1 input clock prescale select bits

00 = 1:1 prescale value

01 = 1:2 prescale value

10 = 1:4 prescale value

11 = 1:8 prescale value

T1OSCEN: Timer1 oscillator enable bit

0 = Timer1 oscillator is shut off 1 = Timer1 oscillator is enabled

T1SYNC: Timer1 external clock input synchronization select bit

When TMR1CS = 1

0 = Synchronize external clock input

1 = Do not synchronize external clock input

When TMR1CS = 0

This bit is ignored.
TMR1CS: Timer1 clock source select bit

0 = Instruction cycle clock (FOSC/4)

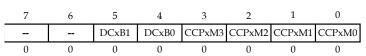
1 = External clock from pin RC0/T1OSO/T13CKI

TMR1ON: Timer1 on bit

0 = Stop Timer1

1 = Enables Timer1

Figure 8.4. T1CON contents (redraw with permission of Microchip)



DCxB1:DCxB0: PWM duty cycle bit 1 and bit 0 for CCP module x

capture mode:

unused compare mode:

unused

PWM mode:

These two bits are the lsbs (bit 1 and bit 0) of the 10-bit PWM duty cycle.

CCPxM3:CCPxM0: CCP module x mode select bits

0000 = capture/compare/PWM disabled (resets CCPx module)

0001 = reserved

0010 = compare mode, toggle output on match (CCPxIF bit is set)

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, initialize CCP pin low, on compare match force CCP pin high (CCPxIF bit is set)

1001 = compare mode, initialize CCP pin high, on compare match force CCP pin low (CCPxIF bit is set)

1010 = compare mode, generate software interrupt on compare match (CCP pin unaffected, CCPxIF bit is set).

1011 = compare mode, trigger special event (CCPxIF bit is set)

For CCP1 and CCP2: Timer1 or Timer3 is reset on event

For all other modules: CCPx pin is unaffected and is configured as an I/O port. $11xx = PWM \mod e$

Figure 8.10 CCPxCON register (x = 1,...,5) (redraw with permission of Microchip)

Nom	i Cognoms:	

Contesteu en aquest mateix full en l'espai reservat de cada pregunta. Raoneu totes les respostes. Les respostes que no vagin acompanyades d'una mínima explicació no s'admetran.

- 4. Si s'utilitza una entrada connectada a una interrupció (per exemple la **INTO**) per, en la subrutina d'interrupció, 'capturar' l'instant de temps precís en que hi ha un flanc de pujada, quines seran les principals fonts de variabilitat en els resultats obtinguts? Quin és el maquinari adequat per aquests tipus de tasques? (1 punt)
 - Les principals fonts que farien variar els resultats són l'efecte d'atendre a altres possibles interrupcions que puguin co-existir amb la INTO i la variabilitat en latència de les interrupcions (3 o 4 cicles). Recordar que un retard constant no provocaria una variabilitat en els resultats, només un offset que es podria compensar. El maquinari adient és el CCP en mode Capture.
- 5. En el PIC18 cal respectar l'ordre de lectura de la part baixa (TMRL) i alta (TMRHighByte) dels comptadors de 16 bits. Quin és l'error màxim (mesurat en *ticks* del comptador) que es podria atribuir al fet de no respectar l'esmentat ordre de lectura? (1 punt)
 - Com que el timer és de 16 bits s'han de fer dues lectures per llegir el registre TMR complet. Si no es respecta l'ordre correcte (recordar que llegir el registre TMRL realitza una copia de la part alta en el registre auxiliar TMRH) llavors es llegirà qualsevol valor emmagatzemat anteriorment en aquest registre. L'error pot arribar a ser dels 8 bits més significatius del registre de 16 bits és a dir 65280.
- 6. Atesa la següent nota (extreta del full tècnic del PIC18), en quin cas es generaria una interrupció immediatament després d'habilitar les interrupcions? (1 punt)

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

Cas que s'hagi produït una interrupció mentre les interrupcions estaven inhabilitades i que no s'hagin netejat els flags d'interrupció associats (o amb posterioritat a netejar els flags).

7. Quina seria el nombre màxim de consignes diferents de potències PWM que es podrien generar, si el període del clock que entra en el Timer 2 és de 1 μseg (amb prescaler 1:1) i el valor del PR2 és 200? I si el valor que entra en el Timer 2 fos de 3 μseg i prescaler 1:4 ? (2 punts)

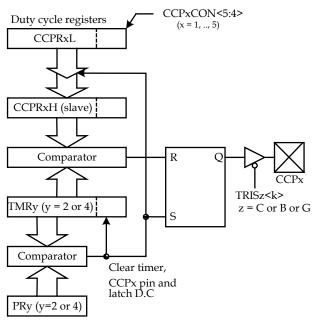


Figure 8.24 Simplified PWM block diagram (redraw with permission of Microchip)

Si el registre del PR2 és igual a 200, 200 tics del Timer defineixen el període del PWM, en conseqüència només es podran ajustar *duty cycles* de 0 a com a molt 200. Com que el registre CCPR afegeix dos bits extra (CCPxCOn<5:4>) per augmentar la resolució del PWM obtenim 804 *duty cycles* diferents. Aquest valors de duty cicles diferents és INDEPENDENT de la freqüència que li entra al TIMER (segona qüestió).