

## Computer Interfacing. Segon parcial, 23/11/2017

Respondeu les preguntes a l'espai reservat. **Justifiqueu** les respostes.

*Poseu el nom a tots els fulls. Totes les preguntes valen 1 punt.*

1. Volem calcular la velocitat de rotació d'un motor mitjançant un PIC18F4550 funcionant a 8Mhz. Per això hem col·locat un sensor que produeix un flanc de pujada a cada volta del motor i hem configurat la unitat CCP1 del PIC per poder mesurar el temps entre polsos. Volem fer servir únicament interrupcions per obtenir la dada. Sabent que el règim del motor estarà entre els 0,5Hz i els 50 Hz, quantes interrupcions haurem d'habilitar i tractar per resoldre aquest problema?. Justifiqueu la resposta amb els càlculs pertinents.

Una interrupció que segur que haurem de configurar es la interrupció associada al CCP1 en mode captura. Recordem que en aquest mode, cada cop que es compleix la condició de captura (segons la configuració del bits CCP1M3..CCP1M0 del registre CCP1CON) capturem el valor del timer1/3 en el registres CCPR1H..CCPR1L.

Ara be, seria possible que el timer1/3 es desbordessin abans de que es produís la condició de captura. En tal cas també seria necessari programar la interrupció associada al desbordament del timer que hem configurat per treballar amb el CCP.

Mirem si serà necessari. Observem que el pitjor cas seria el corresponent a haver de mesurar el període més gran (o la freqüència més baixa) que s'especifica per enunciat, es a dir, 0,5Hz que es correspon amb un període de 2 segons.

La pregunta és: **Es desbordara el timer abans de que passin aquest 2 segons?**

Veiem que el la freqüència d'increment del timer1/3 es de 2MHz i el màxim prescaler que podem aplicar tan en el timer 1 com en el timer 3 es de 8. Es a dir, que com a màxim podem reduir la freqüència d'increment de timer a  $2.000.000 / 8$  ticks per segon, es a dir = 250.000 ticks per segon. Per poder mesurar 2 segons per tant es necessari poder comptar fins a  $250.000 \times 2 = 500.000$  ticks. Com que tenim registres de 16 bits, podem comptar com a màxim fins a 65536 sense que el timer es desbordi. **Per tant podem afirmar que TAMBÉ serà necessari programar la interrupció de timer1/3.**

**FIGURE 14-1:      TIMER3 BLOCK DIAGRAM**

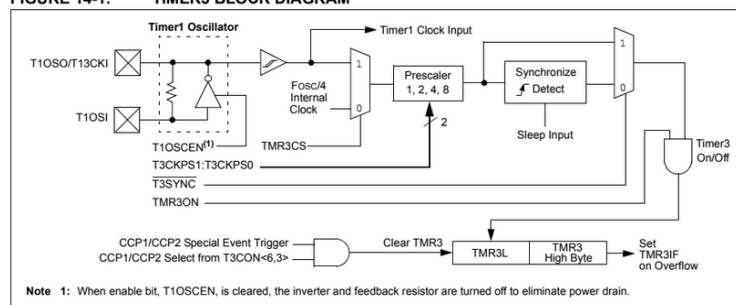
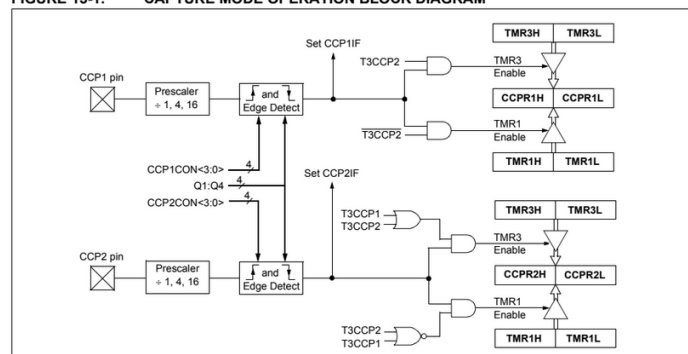


FIGURE 15-1: CAPTURE MODE OPERATION BLOCK DIAGRAM

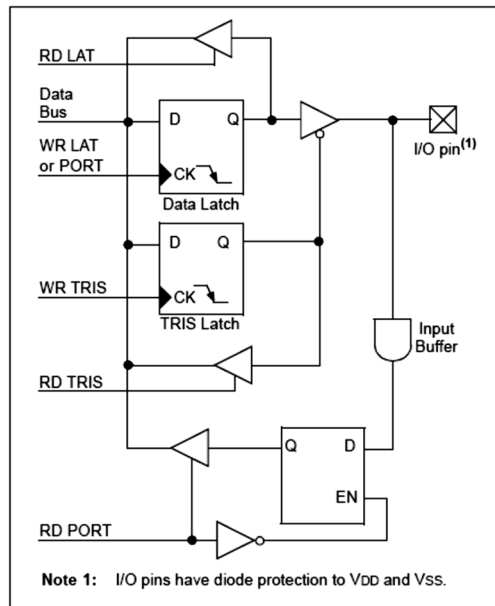


Nom i Cognoms: \_\_\_\_\_

2. Segons l'esquema següent corresponent a un pin d'un port d'E/S, penseu que és possible trobar al biestable LAT un valor diferent que en el biestable PORT?

Si configurem el pin com un pin de sortida totes les escriptures en el registre LAT també afecten al registre PORT a través del buffer tri-state i l'input buffer. Per tant els valors són coherents.

Ara bé, quan estem en mode entrada, les lectures del PORT (RDPORT) no afecten al biestable LATCH, per tant un codi com:  $TRISX=1$  (entrada),  $var=PORTX$  i  $LATX=\text{not}(var)$  farà que els valors siguin diferents (X pot ser qualsevol pin de qualsevol port).

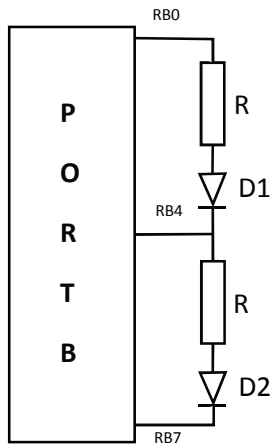


3. Marca les següents afirmacions com a verdaderes (V) o falses (F).

- ☒ V Quan es produeix una interrupció, en TOTS els casos el context d'execució es guarda als *shadow registers* (o a la pila ràpida de registres).
- ☐ F El bit IF associat a una interrupció només s'activarà en el cas que haguem habilitat la interrupció amb el bit IE associat.
- ☐ F La interrupció externa INT0 està mapejada en el pin RB1
- ☐ F El context d'execució que es guarda als *shadow registers* està format pels registres WREG, STATUS, BSR i PC.
- ☒ V Una interrupció, sigui quina sigui la seva prioritat, mai pot ser interrompuda per una interrupció de la mateixa prioritat.

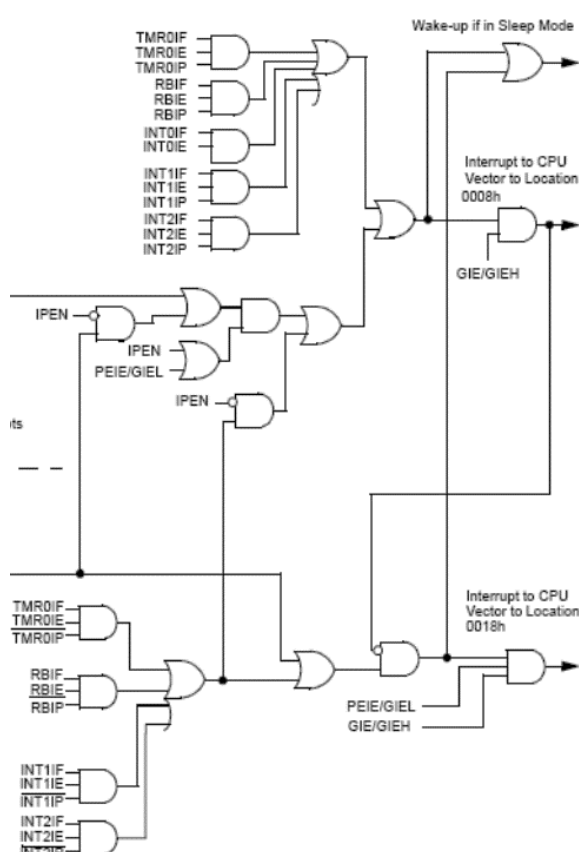
Nom i Cognoms: \_\_\_\_\_

4. Existeix alguna configuració dels registres TRISB i PORTB per a que s'encengui el diode led D1 i el diode led D2 resti apagat? I per a que s'encenguin alhora tots dos diodes (D1 i D2)? En cada cas, si existeix, indiqueu els valors amb els que es configurarien els registres anteriors.



Per encendre D1 i no D2, existeixen diverses opcions, una d'elles és  $TRISB = 0b11101110$  (RB0 i RB4 de sortida, RB7 d'entrada).  $PORTB = 0b00000001$  (RB0 = 5v i RB4= 0v). Per encendre D1 i D2, existeix només una opció,  $TRISB = 0b01111110$  (RB0 i RB7 de sortida, RB4 d'entrada).  $PORTB = 0b00000001$  (RB0 = 5v i RB7= 0v, el corrent no deriva cap a RB4 doncs està en alta impedància).

5. Indiqueu els valors dels bits dels registres de configuració següents per a que es produeixi una interrupció de baixa prioritat associada a l'entrada INT1. Justifiqueu breument la resposta.



$GIE = 1$  (Interrupcions habilitades)  
 $GIEL = 1$  (interrupcions de baixa prioritat habilitades)  
 $IPEN = 1$  (habilitem prioritats en les interrupcions)  
 $INT1IF = 0 \rightarrow 1$  (El posem a 0. Al passar a 1 per hardware o per software (en el codi nostre fem  $INT1IF=1$ ) es provocarà la interrupció. )  
 $INTIE = 1$  (Interrupció habilitada)  
 $INTIP = 0$  (Interrupció en baixa prioritat)

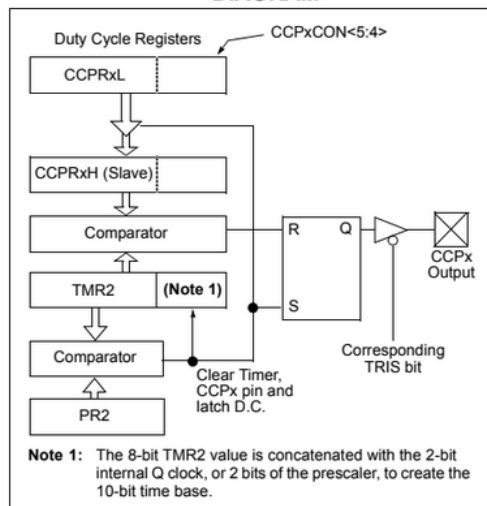
Nom i Cognoms: \_\_\_\_\_

6. Utilitzant la unitat CCP del PIC, quina és la freqüència de PWM més baixa possible que es pot generar amb el PIC18F funcionant a 8MHz si es vol que en aquest PWM tinguem una resolució de 100 divisions (és a dir, poder generar 100 DC diferents)?

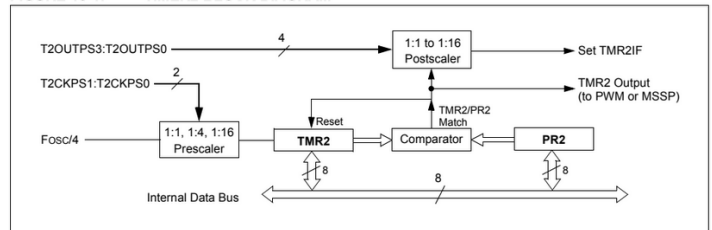
El nombre de potències o DCs diferents que es poden generar amb el PWM del pic 18F és igual a  $(PR2 + 1) \cdot 4$  (utilitzant els dos bits extra del registre CCPR). Això implica que  $PR2 = 24$  per tenir 100 divisions diferents.

Amb prescaler a 16 podem generar el periode del PWM el més gran possible (obtenint la freqüència més baixa).  $T_{pwm} = 100 \cdot 1 / 8\text{MHz} \cdot 16 = 200 \text{ useg}$  (5KHz)

**FIGURE 15-3: SIMPLIFIED PWM BLOCK DIAGRAM**



**FIGURE 13-1: TIMER2 BLOCK DIAGRAM**



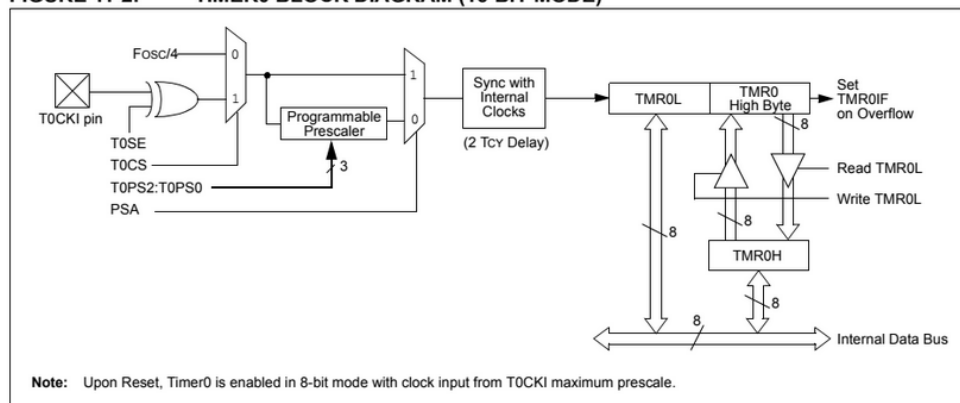
**EQUATION 15-1:**

$$\text{PWM Period} = [(PR2) + 1] \cdot 4 \cdot T_{osc} \cdot (\text{TMR2 Prescale Value})$$

**EQUATION 15-2:**

$$\text{PWM Duty Cycle} = (CCPRxL:CCPxCON<5:4>) \cdot T_{osc} \cdot (\text{TMR2 Prescale Value})$$

**FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)**



Nom i Cognoms: \_\_\_\_\_

**REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER**

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

<b>Legend:</b>			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7	<b>TMR0ON:</b> Timer0 On/Off Control bit 1 = Enables Timer0 0 = Stops Timer0
bit 6	<b>T08BIT:</b> Timer0 8-Bit/16-Bit Control bit 1 = Timer0 is configured as an 8-bit timer/counter 0 = Timer0 is configured as a 16-bit timer/counter
bit 5	<b>T0CS:</b> Timer0 Clock Source Select bit 1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (CLKO)
bit 4	<b>T0SE:</b> Timer0 Source Edge Select bit 1 = Increment on high-to-low transition on T0CKI pin 0 = Increment on low-to-high transition on T0CKI pin
bit 3	<b>PSA:</b> Timer0 Prescaler Assignment bit 1 = Timer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler. 0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.
bit 2-0	<b>T0PS2:T0PS0:</b> Timer0 Prescaler Select bits 111 = 1:256 Prescale value 110 = 1:128 Prescale value 101 = 1:64 Prescale value 100 = 1:32 Prescale value 011 = 1:16 Prescale value 010 = 1:8 Prescale value 001 = 1:4 Prescale value 000 = 1:2 Prescale value

7. Amb  $F_{osc}=8\text{MHz}$ , quina freqüència tindrà el senyal que apareixerà al **pin D0** si executem el següent codi?

```
#include "config.h"

main()
{
    T0CON=0x83; TRISD=0xFE;
    while(1)
        if (TMR0IF) {
            PORTD=PORTD+1;
            TMR0IF=0;
        }
}
```

A T0CON hem posat un 0x83, això vol dir timer a 16 bits, prescaler activat a 16. Per tant:

$T_{osc} = 125 \text{ ns}$

$T_{cicle} = 500 \text{ ns}$  (4  $T_{osc}$ )

$T_{timer} = 8 \text{ us}$  (16  $T_{cicle}$ )

$T_{overflow} = 524288 \text{ us}$  (65536 ticks de timer)

Donat que el codi incrementa el port D, el seu valor serà 0,1,2... per tant el bit D0 farà 0 1 0 1 0 1 0 1 i el període resultant serà 1.04 s

**La freqüència és doncs: 0,953 Hz**

8. Quina és la freqüència més baixa que podríem generar en el cas anterior tocant només el registre T0CON? Fes el càlcul!

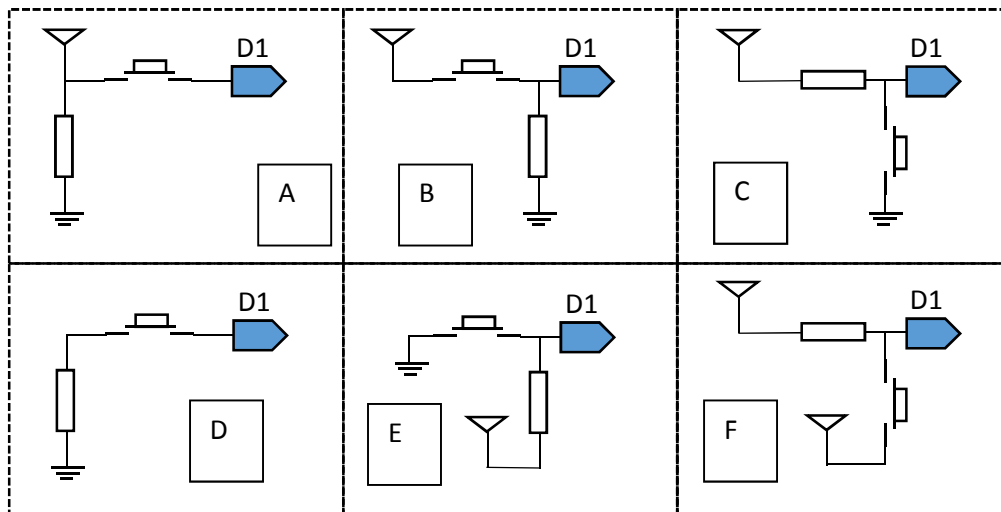
Amb el T0CON sols podem tocar el PREscaler (si el posem en mode 8 bits la freqüència serà més alta, no més baixa ja que compta menys polsos). Posant el PRE a 256, reduïm 16 cops la freqüència d'abans fins a **0,059 Hz**.

Nom i Cognoms: \_\_\_\_\_

9. Podríem amb el PIC18F4550 amb  $F_{osc}=8\text{MHz}$ , atendre una interrupció del Timer0 cada 1 microsegon? Per què?

No. Sols el temps de latència de la interrupció (3 o 4 cicles) ja serà superior a 1us. (3 cicles de  $500\text{us} = 1,5\text{us}$ ).

10. Indica quina o quines d'aquestes configuracions són correctes per detectar per software el funcionament (prémer, deixar anar, tornar a prémer...) d'un botó connectat al pin D1.



- A. No funciona. Tenim aire (indeterminat) si no prémem el botó i Vcc (1 lògic) si si.
- B. Funciona. Tenim 0 lògic si no premem (pull down) i 1 lògic si premem.
- C. Funciona. Tenim 1 lògic si no premem (pull up) i 0 si premem.
- D. No funciona. Tenim indeterminat si no premem i 0 si premem.
- E. Funciona. És igual que el C.
- F. No funciona. En tots els casos tenim un 1 lògic.