

Nom i Cognoms: _____ **Una possible solució** _____

1) (1p.) Configurem la unitat PWM del PIC18F4550 amb el valor PR2 = 1, prescaler = 1:4.
Quants duty cycles diferents es poden programar amb aquesta configuració sense usar els bits CCPxCON<5:4>?

Podem configurar el registre CCPR amb els valors 0 o 1

Duty cycles = 2

Quants duty cycles diferents es poden programar amb aquesta configuració usant els bits CCPxCON<5:4>?

Si afegim 2 bits, multipliquem la resolució per 4

Duty cycles = 8

2) (1p.) Quines accions diferents implementa la unitat de control del micro si es troba un return, un retfie, o un retfie fast?

Completeu la taula següent amb les diferents accions en forma d'assignacions simples (com a l'exemple).
Qualsevol resposta a base de text no serà comptada com a vàlida.

return	retfie	retfie fast
PC ← (Top_of_Stack)	PC ← (Top_of_Stack) GIEx ← 1	PC ← (Top_of_Stack) GIEx ← 1 W ← W_shadow STATUS ← STATUS_shadow BSR ← BSR_shadow

3) (1p.) Si connectem un pin de sortida estàndar (TTL buffer) del PIC18F4550 alimentat a 5V a un pin d'entrada d'un altre micro igual,

Quin serà el marge de soroll si la sortida està a nivell baix?

$$V_{il} - V_{ol} = 0.8V - 0.6V = 0.2V$$

NM_L = 0.2V

Quin serà el marge de soroll si la sortida està a nivell alt?

$$V_{oh} - V_{ih} = 4.3V - 2V = 2.3V$$

NM_H = 2.3V

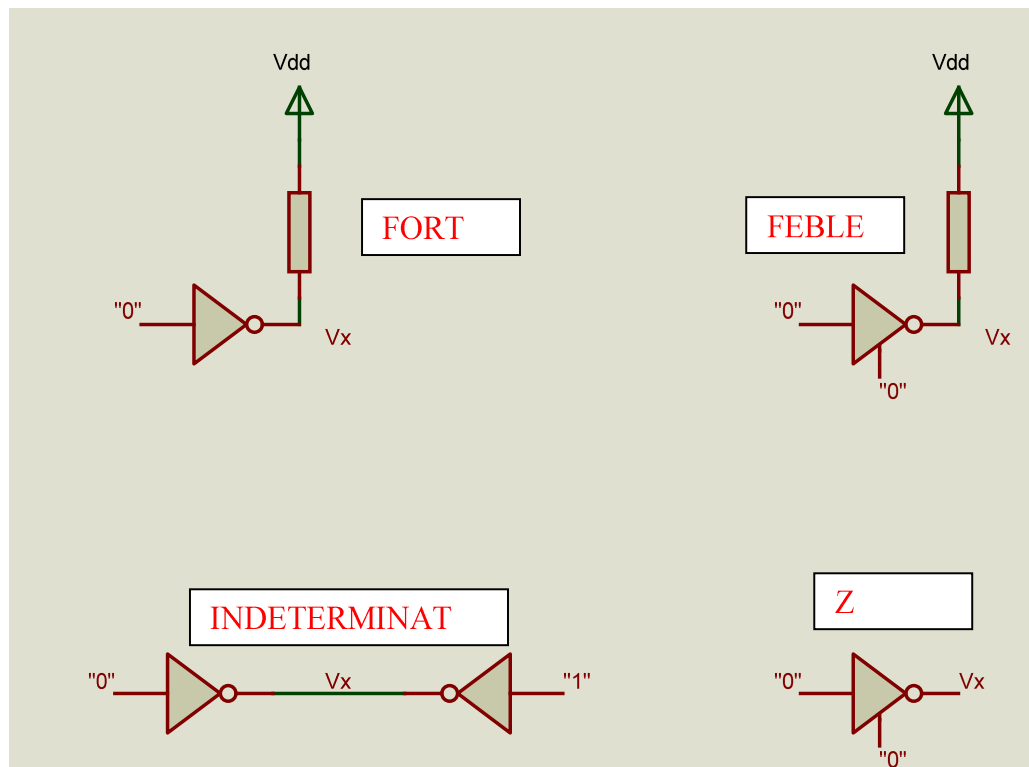
Per tant, podem garantir que el marge de soroll és de

Agafem el pitjor cas

NM = 0.2V

Nom i Cognoms: _____ **Una possible solució** _____

4) (1p.) Indiqueu si la sortida Vx dels següents 4 esquemes es troba en un estat lògic FORT, FEBLE, Z (alta impedància), o INDETERMINAT.



5) (1p.) S'ens presenten dues versions d'un programa per escriure i llegir dades del PORTB. Indiqueu quin és el valor escrit en el PORTB en l'instant assenyalat amb **, i quin valor acabem llegint del PORTB pels dos codis presentats.

```
int valor ;
...
TRISB = 0; // PORTB de sortida
PORTB = 0x55; // **
nop();
TRISB = 0xFF; // PORTB d'entrada
valor = PORTB;
```

PORTB ** = 0x55

Valor = Estat de l'entrada

```
int valor ;
...
TRISB = 0; // PORTB de sortida
LATB = 0x55; // **
nop();
TRISB = 0xFF; // PORTB d'entrada
valor = LATB;
```

PORTB ** = 0x55

Valor = 0x55

Nom i Cognoms: _____ **Una possible solució** _____

6) (1p.) Si F_{osc} val 8MHz i en temps $t=0$, posem els valors als registres $TMR1L=0x12$, $TMR1H=0x00$ i $T1CON=0x95$, quin valor tindran $TMR1L$ i $TMR1H$ en temps $t=1$ segon? (és a dir, 1 segon més tard).

A partir de la configuració de $T1CON$ veiem que el Timer1 està en mode 16 bits, amb Prescaler de 2, que utilitza el clock d'instrucció ($F_{osc}/4$) i que està en marxa.

Per tant al Timer1 li arriba amb el Prescaler: $(F_{osc}/4)/2 = 1$ MHz. És a dir un pols cada 1us (microsegon).

En 1 segon haurà rebut 1000000 de polsos, per tant haurà fet 15 overflows i comptat 16960 ticks (1000000 mòdul 65536).

El seu valor final serà 18 (valor inicial en base 10) + 16960 = **16978**

Per tant els registres del TMR1 (ara en hexa) valdran **0x4252**.

7) (2p.) La unitat de Capture rep a l'entrada CCP1 i a l'entrada CCP2 un senyal de freqüència 2 KHz i Duty Cycle variable. Si configurem la CCP1 per a detectar cada flanc de pujada i la CCP2 per a detectar cada flanc de baixada, $T3CCP2=0$, $T3CCP1=0$ i el TIMER1 està configurat com en l'exercici anterior,

- Quina resolució tindrem per mesurar el Duty Cycle del senyal d'entrada?

El senyal d'entrada té una freqüència de 2KHz, per tant, un període de 500us.

De l'exercici anterior veiem que el Timer1 rep un tick cada 1us.

Per tant, podrem distingir **500 valors de DC diferents**.

Si expressem el DC com a percentatge, veiem que tenim una resolució del **0,2%**.

- Si el senyal d'entrada té un Duty Cycle del 76% i el registre CCPR1 val 0xF003, quin valor tindrà el registre CCPR2?

Si el DC està al 76%, vol dir que dels 500us, estem 380us a nivell lògic 1 i 120us a nivell lògic 0.

Si el CCP1R, en el moment del flanc de pujada captura un valor del Timer1 (que compta microsegons) de 0xF003, equivalent a 61443 en base 10, tenim que:

En l'instant del flanc de pujada, el registre CCPR2 encara tindrà el valor de **61443-120** (moment del darrer flanc de baixada).

Quan hi hagi el següent flanc de baixada el valor capturat al CCPR2 serà **61443+380**.

(qualsevol de les dues respostes s'ha considerat valida)

Nom i Cognoms: _____ **Una possible solució** _____
 8) (2p.) Tenim el PIC18F4550 amb un rellotge Fosc=12MHz. Programeu tot el que creieu necessari per tenir una interrupció periòdica TMR0IF cada 1ms.

```
void main (void)
{
```

```
    IPEN=1
    TMR0IP=1
    TMR0IF=0
    TMR0IE=1
    // Preparo interrupcions
    // a alta prioritat (també
    // pot ser baixa, però no
    // les dos alhora)
    T0CON=0x88
    // No cal Prescaler
    TMR0=65536-3000
    // En 1 ms ha de comptar
    // 3000 ticks a Fosc/4
    GIE=1
    //engeguem.
```

```
...
```

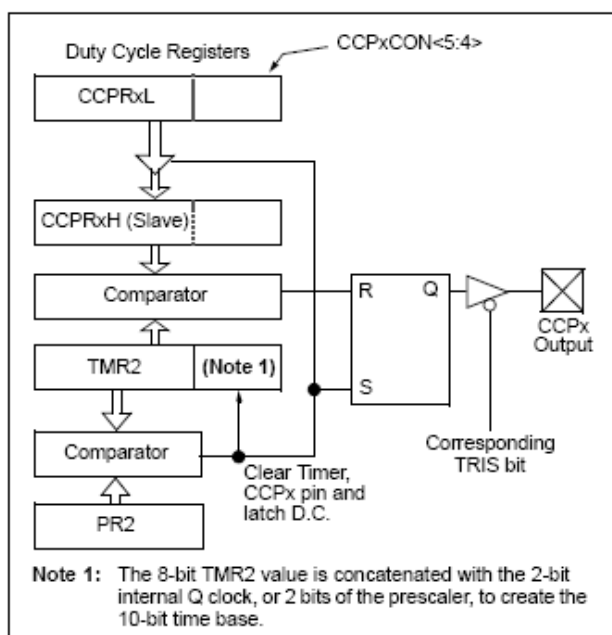
```
    // codi de programa, per
    // exemple, while(1)
```

```
void interrupt High Interr()
{
```

```
    if (TMR0IF && TMR0IE)
    {
        TMR0=65536-3000
        // Cal recarregar!!
        TMR0IF = 0
        // Cal resetejar el flag
    }
```

```
void interrupt Low Interr()
{
```

```
    // Res a dir
```



Nom i Cognoms:

Una possible solució

	7	6	5	4	3	2	1	0
value after reset	RD16	--	TICKPS1	TICKPS0	TIOSCEN	TISYNC	TMR1CS	TMR1ON
	0	0	0	0	0	0	0	0

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

TICKPS1:TICKPS0: 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

TIOSCEN: Timer1 oscillator enable bit
 0 = Timer1 oscillator is shut off
 1 = Timer1 oscillator is enabled

TISYNC: 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/T1CKI

TMR1ON: Timer1 on bit
 0 = Stop Timer1
 1 = Enables Timer1

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

	TMR0ON	08BIT	CS	SI	PS1	PS0	PS1	PS0
value after reset								

TMR0ON: Timer0 on/off control bit
 0 = stops Timer
 1 = Enables Timer

08BIT: 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

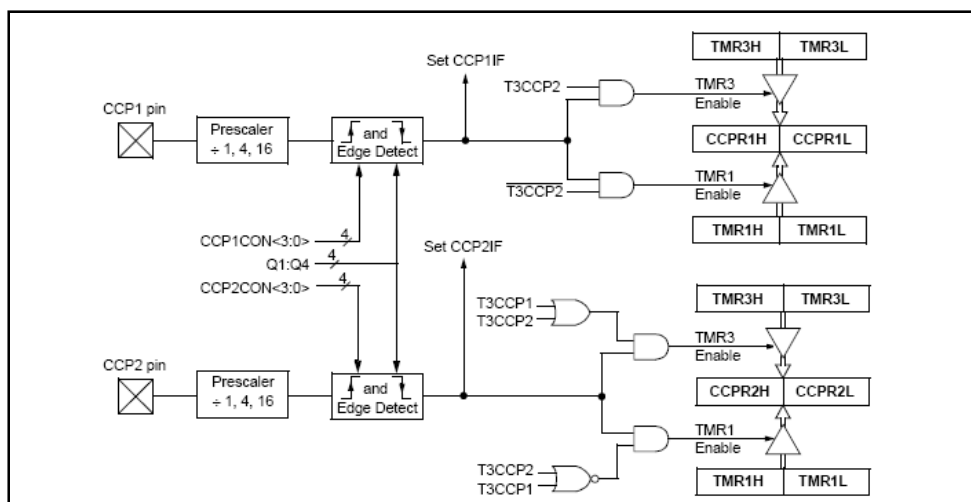
CS: Timer0 clock source select
 0 = Instruction cycle clock
 1 = Transition on T1CKI pin

SI: Timer0 source edge select bit
 0 = Increment on falling edge transition on T1CKI pin
 1 = Increment on rising edge transition on T1CKI pin

PS1:PS0: Timer0 prescaler assignment bit
 0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output
 1 = Timer0 prescaler is not assigned. Timer0 clock input bypasses prescaler

PS3:PS0: Timer0 prescaler select bits
 000 = 2 prescaler value
 001 = 4 prescaler value
 010 = 8 prescaler value
 011 = 16 prescaler value
 100 = 32 prescaler value
 101 = 64 prescaler value
 110 = 128 prescaler value
 111 = 256 prescaler value

Figure 2 T1CON register reprint with permission of Microchip



Una possible solució

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial			
Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
D080	VOL	Output Low Voltage I/O Ports (except RC4/RC5 in USB mode)	—	0.6	V	IO _L = 8.5 mA, VDD = 4.5V, -40°C to +85°C
D083		OSC2/CLKO (EC, ECIO modes)	—	0.6	V	IO _L = 1.6 mA, VDD = 4.5V, -40°C to +85°C
D090	VOH	Output High Voltage⁽³⁾ I/O Ports (except RC4/RC5 in USB mode)	VDD - 0.7	—	V	IO _H = -3.0 mA, VDD = 4.5V, -40°C to +85°C
D092		OSC2/CLKO (EC, ECIO, ECPIO modes)	VDD - 0.7	—	V	IO _H = -1.3 mA, VDD = 4.5V, -40°C to +85°C

