

**FIB, Interfícies dels Computadors**  
**Segon parcial 15-12-2014 (1h30', Full 1/2)**

**COGNOMS:**

**NOM:**

**DNI:**

**Responen en aquest mateix full. Cal justificar totes les respostes.**  
**Respostes sense un mínim text explicatiu no es tindran en consideració.**

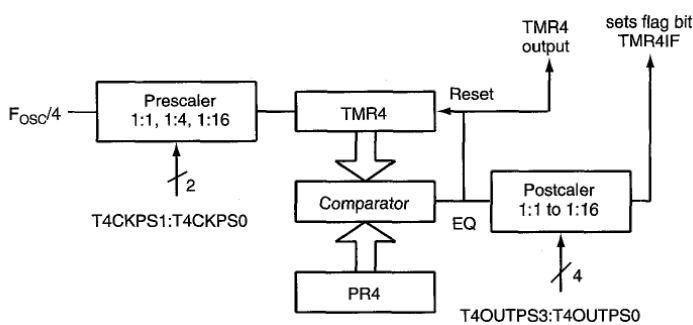
1. El codi que es mostra a continuació és part d'una rutina d'atenció de les interrupcions de baixa prioritat.

```
MOVWF W_TEMP ; W_TEMP is in virtual bank
MOVFF STATUS, STATUS_TEMP ; STATUS_TEMP located anywhere
MOVFF BSR, BSR_TEMP ; BSR_TEMP located anywhere
;
; USER ISR CODE
;
MOVFF BSR_TEMP, BSR ; Restore BSR
MOVFF STATUS_TEMP, STATUS ; Restore STATUS
MOVF W_TEMP, W ; Restore WREG
RETFIE LOW_PRIORITY
```

Hi ha alguna cosa errònia en el codi anterior si considerem que la instrucció MOVF modifica els flags Z i N del registre STATUS? (1 punt)

L'ordre de recuperació dels registres està equivocat, doncs al recuperar el registre W es poden modificar els bits del registre STATUS

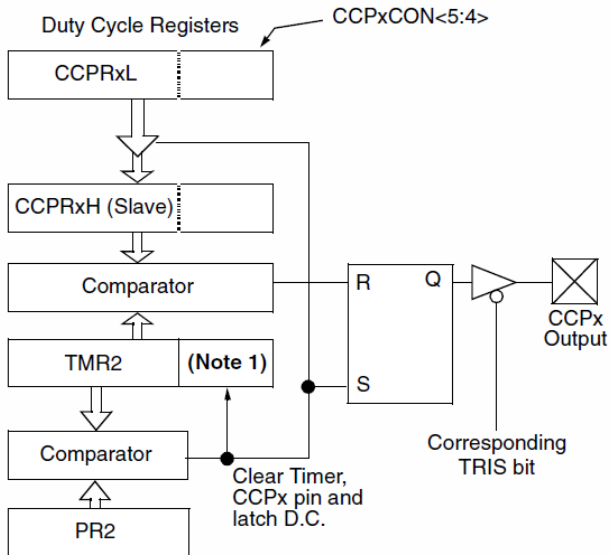
2. Quina és la freqüència mínima i màxima del senyal TMR4IF que es podrà generar amb el Timer que es mostra a continuació? Considereu  $F_{osc} = 10\text{MHz}$ . (2 punts)



Freqüència màxima =  $F_{osc}/4$ ; amb el Prescaler = 1, PR4 = 1 i Postcaler = 1.

Freqüència mínima =  $F_{osc}/(4 \cdot 16 \cdot 16 \cdot 255)$ ; amb el Prescaler = 16, PR4 = 255 i Postcaler = 16.

3. Quin *duty-cycle* es generarà amb la unitat PWM següent si el valor dels registres PR2 = 128, CCPRxL = 64, CCPxCON<5:4> = <0,0> ? (1 punt)



$$\text{DutyCycle} = \text{Ton} / \text{Tpwm} = (\text{CCPRxL} * 4 + \text{CCPxCON}\langle 5:4 \rangle) \text{ Tosc} / (\text{PR2} * 4 * \text{Tosc}) = 50\%$$

**Note 1:** The 8-bit TMR2 value is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler, to create the 10-bit time base.

4. Quin seria el temps total de mostreig donats els valors de ADSC2:ADSC0 = 001, ACQT2:ADCQT0 = 010 i ADFM = 1. Podem assegurar que TAD > 0.8 µseg i TACQ > 2.45 µseg? L'oscil·lador del sistema és de 8MHz. (1 punt)

$$\text{ADSC2:ADSC0} = 001 \rightarrow T_{AD} = (F_{osc}/8)^{-1} = 1 \mu\text{seg} > 0,8 \mu\text{seg}$$

$$\text{ACQT2:ADCQT0} = 010 \rightarrow T_{ACQ} = 4 * T_{AD} = 4 \mu\text{seg} > 2.45 \mu\text{seg}$$

$$T_{\text{Mostreig}} = 12 * T_{AD} + T_{ACQ} = 16 \mu\text{seg}$$

**FIB, Interfícies dels Computadors**  
**Segon parcial 15-12-2014 (1h30', Full 2/2)**

**COGNOMS:**

**NOM:**

**DNI:**

**Responen en aquest mateix full. Cal justificar totes les respostes.**  
**Respostes sense un mínim text explicatiu no es tindran en consideració.**

5. Escriu la inicialització dels registres necessaris per configurar els mecanismes d'interrupcions i les rutines de serveis a la interrupció amb la finalitat de:

- U  
n sensor **A** binari i està connectat a l'entrada INT1. La interrupció INT1 cal definir-la com a prioritat alta i la seva activació serà per flanc de baixada del senyal. El tractament de la interrupció INT1 consisteix en incrementar una variable de 0 fins a 65535.
- U  
n sensor **B** binari i està connectat a l'entrada INT2. La interrupció INT2 es definirà com a prioritat baixa, i la seva activació serà per flanc de pujada del senyal. El tractament de la INT2 consisteix en complementar el valor del bit 0 del PORTC.
- P  
ot haver altres fonts d'interrupcions tant altes com baixes

En certes condicions pot succeir que un flanc de baixada en el senyal associat a INT1 no produeix l'execució de la RSI de prioritat alta associada. Amb aquestes consideracions, quin(s) factor(s) limita(en) la freqüència màxima del senyal **A**? Quin(s) factor(s) limita(en) la freqüència màxima del senyal **B**? (Justifica les respostes.)

```
#include<xc.h>

unsigned int contador=0;

void interruptRSIHigh (){
    if (INT1IE && INT1IF) { // INT1 RSI
        INT1IF=0;
        contador++;
    };
    // Code for others high priority RSI
}

void interruptlow_priorityRSILow (){
    if (INT2IE && INT2IF) { // INT2 RSI
        INT2IF=0;
        PORTCbits.RC0=~PORTCbits.RC0;
    };
    // Code for others low priority RSI
}

void main(void)
{
    // Init interruption hardware
    INTCON2= 0x10; // INT1 falling edge INT2 rising edge
    INTCON3= 0x58; // Enable INT1 High, INT2 Low, Clear IF's
    RCONbits.IPEN= 1; // Enable priority levels
    TRISB= 0x0F; // RB3 a RB0 set as Digital IN
    ADCON1= 0x0F; // All IN are DIGITAL
    INTCON= 0xC0; // Enable High & Low priority INTS
    TRISC= 0x00; // PORT C Output
    PORTC= 0x00;
    while (1) {
        // Code for user application
    };
}
```

Si es compleix la condició que genera la interrupció (un flanc de baixada de INT1 o el flanc de pujada de INT2) el bit INTxIF associat s'activa (=1). Si es produeix una nova condició de interrupció i encara el INTxIF no s'ha desactivat (=0) a la RSI, aquell esdeveniment no serà tractat (es sobreescriu '1' sobre INTxIF que ja valia '1').

En el cas de les prioritats altes, el temps que passa des de que es produeix la condició d'interrupció fins que es desactiva el IF és la latència del mecanisme d'interrupcions (3 o 4 Tinstrucció) més el temps d'execució de les instruccions de la RSI High\_priority fins que es desactiva el IF (INT1IF=0 en el nostre cas).

En el cas de les prioritats baixes, és el mateix, però en el pitjor cas hem de tenir present que si s'està executant la RSI High\_priority, la RSI Low\_priority no s'executa fins que aquesta acabi, i per tant el temps que passa des de que es produeix la condició d'interrupció fins que es desactiva el IF (en el pitjor cas) és el temps d'execució de la RSI High\_priority, més la latència del mecanisme d'interrupcions (3 o 4 Tinstrucció) més el temps d'execució de les instruccions de la RSI Low\_priority fins que es desactiva el IF (INT2IF=0 en el nostre cas).

6. Assigna els valors adequats als registres necessaris per a generar una interrupció d'alta prioritat al cap de 10 mil·lisegons, utilitzant el TIMER1. Fosc= 8 MHz. (1 punt)

$F_{osc} = 8 \text{ MHz} \Rightarrow F_{cycle} = 2 \text{ MHz} = T_{cycle} = 0.5 \mu s$        $10 \text{ ms} = 10000 \mu s$   
 $10000 / 0.5 = 20000$  (calen 20,000 períodes de 0.5  $\mu s$  per completar 10 ms)

Per tant cal el TIMER en mode 16 bits, i un prescaler de 1:1 és suficient.  
Cal inicialitzar el TIMER1 amb el valor  $65,535 - 20,000 = 35,535 = 0x8ACF$

```
PIR1= 0x00; // Initinterruption. Clear IF
IPR1= 0x01; // TIMER1 Highpriority
PIE1= 0x01; // Enable TIMER1 interruption
RCONbits.IPEN= 1; // EnableInterruptpriority
TMR1H= 0x8A; // 0x8ACF = 35,535
TMR1L= 0xCF;
T1CON= 0x81; // 16 bits, TIMER1 ON
INTCON= 0xC0; // Enableinterruptions
```

7. Es vol utilitzar un PIC18F4550 per monitoritzar els valors proporcionats per 6 sensors analògics connectats a les entrades AN0 a AN5. Les tensions subministrades pels sensors ( $V_{j=0 \text{ a } 5}$ ) estan dintre del rang VDD - VSS, i es desitja treballar amb una resolució de 8 bits. Fosc= 20MHz. (2 Punts)

Escriu el codi per a la configuració del conversor A/D del PIC18F4550 i per fer 40 cicles de lectura dels valors  $V_j$ , que s'emmagatzemaran en la matriu MOSTRA[i][j], amb l'objectiu que la freqüència de mostreig del sensors sigui el més gran possible.

Indicar per la solució proposada quin és el temps necessari per l'obtenció d'una mostra ( $T_{Adquisició} + T_{Conversió}$ )

Amb  $F_{osc} = 20 \text{ MHz}$ , el valor necessari per fer  $T_{ADQ}$  el més petit possible, tot respectant  $T_{ADQ} \geq 0.8 \mu s$  és  $AD_{conversionclock} = F_{osc} / 16 \Rightarrow T_{AD} = 0.8 \mu s$ . El valor per  $T_{ADQ} = 4 * T_{AD} > 2.45 \mu s$ .

$AD_{convesrsionclock} = F_{osc} / 16 = 20 \text{ M} / 16 = 1.25 \text{ MHz} \Rightarrow T_{AD} = 0.8 \mu s \Rightarrow T_{CONV} = 12 * T_{AD} = 9.6 \mu s$  i  $T_{ADQ} = 4 * T_{AD} = 3.2 \mu s$   
 $T_{mostreig} = T_{ADQ} + T_{CONV} = 3.2 \mu s + 9.6 \mu s = 12.8 \mu s \Rightarrow F_{mostreig} = 78,125 \text{ KHz}$

// Codi per exercici CAD

```
ADCON0= 0x01; // Channel 0 AD ON
ADCON1= 0x09; // ADC Voltage reference VddVss. RA0 to RA5 are analog input
```

```

TRISA= 0xFF; //
ADCON2= 0x15; // Leftjustified, Fosc/16, 4 TAD

for (i=0; i<40; i++) {
    ADCON0= 0x01;
    for (j=1; j<7; j++) {
        ADCON0bits.GO=1; // Startconversion
        while (ADCON0bits.GO=1); // Waituntilconversioncomplete
        ADCON0= (j<<2) + 1; // Select Channel j, AD ON
        MOSTRA[i][j]= ADRESH; // Store conversionresult
    };
};

```

Encara que en el codi solució no està fet així, es pot fer de forma concurrent el canvi del canal d'entrada del CAD i el treball del conversió del CAD.

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF <sup>(1)</sup>

bit 7	<b>GIE/GIEH:</b> Global Interrupt Enable bit When <b>IPEN</b> = 0: 1 = Enables all unmasked interrupts 0 = Disables all interrupts When <b>IPEN</b> = 1: 1 = Enables all high-priority interrupts 0 = Disables all interrupts
bit 6	<b>PEIE/GIEL:</b> Peripheral Interrupt Enable bit When <b>IPEN</b> = 0: 1 = Enables all unmasked peripheral interrupts 0 = Disables all peripheral interrupts When <b>IPEN</b> = 1: 1 = Enables all low-priority peripheral interrupts (if GIE/GIEH = 1) 0 = Disables all low-priority peripheral interrupts
bit 5	<b>TMR0IE:</b> TMR0 Overflow Interrupt Enable bit 1 = Enables the TMR0 overflow interrupt 0 = Disables the TMR0 overflow interrupt
bit 4	<b>INT0IE:</b> INT0 External Interrupt Enable bit 1 = Enables the INT0 external interrupt 0 = Disables the INT0 external interrupt
bit 3	<b>RBIE:</b> RB Port Change Interrupt Enable bit 1 = Enables the RB port change interrupt 0 = Disables the RB port change interrupt
bit 2	<b>TMR0IF:</b> TMR0 Overflow Interrupt Flag bit 1 = TMR0 register has overflowed (must be cleared in software) 0 = TMR0 register did not overflow
bit 1	<b>INT0IF:</b> INT0 External Interrupt Flag bit 1 = The INT0 external interrupt occurred (must be cleared in software) 0 = The INT0 external interrupt did not occur
bit 0	<b>RBIF:</b> RB Port Change Interrupt Flag bit <sup>(1)</sup> 1 = At least one of the RB7:RB4 pins changed state (must be cleared in software) 0 = None of the RB7:RB4 pins have changed state

REGISTER 9-3: INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF

bit 7	<b>INT2IP:</b> INT2 External Interrupt Priority bit 1 = High priority 0 = Low priority
bit 6	<b>INT1IP:</b> INT1 External Interrupt Priority bit 1 = High priority 0 = Low priority
bit 5	<b>Unimplemented:</b> Read as '0'
bit 4	<b>INT2IE:</b> INT2 External Interrupt Enable bit 1 = Enables the INT2 external interrupt 0 = Disables the INT2 external interrupt
bit 3	<b>INT1IE:</b> INT1 External Interrupt Enable bit 1 = Enables the INT1 external interrupt 0 = Disables the INT1 external interrupt
bit 2	<b>Unimplemented:</b> Read as '0'
bit 1	<b>INT2IF:</b> INT2 External Interrupt Flag bit 1 = The INT2 external interrupt occurred (must be cleared in software) 0 = The INT2 external interrupt did not occur
bit 0	<b>INT1IF:</b> INT1 External Interrupt Flag bit 1 = The INT1 external interrupt occurred (must be cleared in software) 0 = The INT1 external interrupt did not occur

REGISTER 9-4: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
SPPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF

REGISTER 9-6: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE

REGISTER 9-8: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
SPPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP

REGISTER 9-2: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
RBPU	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP

bit 7	<b>RBPU:</b> PORTB Pull-up Enable bit 1 = All PORTB pull-ups are disabled 0 = PORTB pull-ups are enabled by individual port latch values
bit 6	<b>INTEDG0:</b> External Interrupt 0 Edge Select bit 1 = Interrupt on rising edge 0 = Interrupt on falling edge
bit 5	<b>INTEDG1:</b> External Interrupt 1 Edge Select bit 1 = Interrupt on rising edge 0 = Interrupt on falling edge
bit 4	<b>INTEDG2:</b> External Interrupt 2 Edge Select bit 1 = Interrupt on rising edge 0 = Interrupt on falling edge
bit 3	<b>Unimplemented:</b> Read as '0'
bit 2	<b>TMR0IP:</b> TMR0 Overflow Interrupt Priority bit 1 = High priority 0 = Low priority
bit 1	<b>Unimplemented:</b> Read as '0'
bit 0	<b>RBIP:</b> RB Port Change Interrupt Priority bit 1 = High priority 0 = Low priority

REGISTER 9-10: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1 <sup>(1)</sup>	U-0	R/W-1	R-1	R-1	R/W-0 <sup>(2)</sup>	R/W-0
IPEN	SBOREN	—	RI	TO	PD	POR	BOR

bit 7	<b>IPEN:</b> Interrupt Priority Enable bit 1 = Enable priority levels on interrupts 0 = Disable priority levels on interrupts
bit 6	<b>SBOREN:</b> BOR Software Enable bit <sup>(1)</sup> For details of bit operation, see Register 4-1.
bit 5	<b>Unimplemented:</b> Read as '0'
bit 4	<b>RI:</b> RESET Instruction Flag bit For details of bit operation, see Register 4-1.
bit 3	<b>TO:</b> Watchdog Time-out Flag bit For details of bit operation, see Register 4-1.
bit 2	<b>PD:</b> Power-Down Detection Flag bit For details of bit operation, see Register 4-1.
bit 1	<b>POR:</b> Power-on Reset Status bit <sup>(2)</sup> For details of bit operation, see Register 4-1.
bit 0	<b>BOR:</b> Brown-out Reset Status bit For details of bit operation, see Register 4-1.

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON

```

0000 = Channel 0 (AN0)
0001 = Channel 1 (AN1)
0010 = Channel 2 (AN2)
0011 = Channel 3 (AN3)
0100 = Channel 4 (AN4)
0101 = Channel 5 (AN5)(1,2)
0110 = Channel 6 (AN6)(1,2)
0111 = Channel 7 (AN7)(1,2)
1000 = Channel 8 (AN8)
1001 = Channel 9 (AN9)
1010 = Channel 10 (AN10)
1011 = Channel 11 (AN11)
1100 = Channel 12 (AN12)
1101 = Unimplemented(2)
1110 = Unimplemented(2)
1111 = Unimplemented(2)

```

When ADON = 1:  
1 = A/D conversion in progress  
0 = A/D Idle

U-0	U-0	R/W-0	R/W-0	R/W-0 <sup>(1)</sup>	R/W <sup>(1)</sup>	R/W <sup>(1)</sup>	R/W <sup>(1)</sup>
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0

bit 4 VCFG0: Voltage Reference Configuration bit (VREF+ source)  
1 = VREF+ (AN3)  
0 = VDD

[illegible]

D = Digital I/O

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON

1 = Enables Timer1  
0 = Stops Timer1

**Note 1:** When enable bit, T1OSCEN, is cleared, the inverter and feedback resistor are turned off to eliminate power drain.

