Nom i Cognoms: _____POSSIBLE SOLUCIÓ_____

1) Volem configurar un PIC18F45K22 de tal manera que sigui capaç de gestionar les interrupcions provinents de la interrupció externa 2 i del CCP1 configurat en mode Compare quan es produeix una igualtat amb el valor del temporitzador associat al CCP1.

La interrupció externa 2 ha de ser d'alta prioritat i s'ha de disparar per flanc de pujada. La interrupció del CCP1 ha de ser de baixa prioritat.

Indica en la llista els registres què creus que has d'escriure i indica el valor del bits que creus que has de modificar (només els bits que calgui modificar) (1.5 Punts)

Nom del registre	**Bits a modificar del registre 0/1**
_____ _ _ _ _ _ _ _	RCON 1XXXXXXX (IPEN A 1 PRIORITATS)
_____ _ _ _ _ _ _ _	INTCON 11XXXXXX (ENABLE INT I ENABLE H. PRI)
_____ _ _ _ _ _ _ _	INTCON2 XXX1XXXX (INT2 EDGE UP)
_____ _ _ _ _ _ _ _	INTCON3 1XX1XX0X (ENABLE INT2 HIGH PRIO)
_____ _ _ _ _ _ _ _	PIE1 XXXXX1XX (ENABLE INT CCP1)
_____ _ _ _ _ _ _ _	IPR1 XXXXX0XX (PRIORITAT BAIXA CCP1)

2) Tenim un PIC18F45K22 funcionant amb un oscil·lador a **10MHz**. Hem configurat el Timer2 de manera que el registre T2CON te el valor 0bx01001111 (valor en binari) i el registre PR2 és igual a 231. Indica exactament el temps que trigarà el Timer2 a provocar que el flag TMR2IF s'activi. (1.5 Punts)

Com que el timer 2 s'incrementa cada 4 tics de l'oscil·lador tenim que el timer s'incrementarà cada 0,4 microsegons. Multipliquem aquest temps pel valor del preescaler i del post escaler que segons la configuració de T2CON es correspon a 16 i 10 respectivament. Com que el flag s'activarà quan el comparador detecti la igualtat entre PR2 i TMR2 tenim que això es produirà al cap de

0,4 microsegons * 10 * 16 * (231+1) = 14,848 milisegons

3) Volem utilitzar el mòdul CCP1 del PIC18F45K22 funcionant amb un oscil·lador a **8MHz** per generar un senyal PWM de període 0,25 ms i un *duty cicle* del 20,25%. Hem configurat el PIC per tal de que el Timer2 funcioni como a *timer* associat al CCP1.- Quin és el valor amb el que hem de inicialitzar el registres PR2 i CCPR1L i els bits extres DC1B1-DC1B0 per tal d'obtenir aquest senyal amb la màxima precisió possible? (2 Punts) **(Respon l'exercici darrera)**

El timer2 s'incrementa a ¼ de la freqüència de l'oscil·lador principal, per tant a 2MHz o cada 0,5 microsegons. Per fixar el període del senyal del PWM hem de fixar el valor del CCPR1L a 0,25 ms / 0,5 microS = 500. Com que 500 es superior al 2^8 necessitem fixar un prescaler de 4 (el mínim valor del prescaler garanteix la màxima precisió) al Timer2 i fixar el valor de PR2 a (500 / 4) - 1 = 124

Per fixar el valor de CCPR1L i DC1B1 i DC1B0 Aplicant la formula:

Cicle Treball = ([CCPR1L;CCP2CON[5..4])/(4 * PR2 + 1)

Nom i Cognoms: _____POSSIBLE SOLUCIÓ_____

Per tant
0,2025 = (valor que volem trobar en 10 bits) /(4 * 125)
Valor = 101,25
En binari 0001100101
El dos últims bits pertanyen es guarden a DC1B1 i DC1B0 respectivament. Els 8 mes als a CCPR1L
Per tant CCPR1L = 00011001
i
DC1B1 = 0
DC1B0 = 1

4) Si **Fosc=20MHz**, quina és la freqüència més baixa a la que podem rebre interrupcions del Timer0? Indica els càlculs realitzats. (1p)

Fosc = 20 MHz -> Fcycle = 5 MHz
El prescaler més gran és de 256, Fpre = 5MHz / 256 = 19531,25 Hz
Finalment ens queda el Timer0 configurat a 16 bits i sense valor de precàrrega farà una interrupció cada cop que es desbordi (65536 tics), per tant:
Fint = Fpre/65536=0,298Hz

5) Com sabeu, la unitat de CCP pot tenir tres modes diferents i dins d'algun d'ells fer diferents funcions. Indica per cadascun dels modes i funcions indicats quin valor ha de tenir el bit TRIS_CPP (0 / 1 / X = no importa) (1 punt)

CCP1	Mode Compare per generar so.	TRIS_CCP1 = 0 (OUT)
CCP2	Mode Capture d'un senyal extern.	TRIS_CCP2 = 1 (IN)
CCP3	Mode PWM per controlar un motor.	TRIS_CCP3 = 0 (OUT)
CCP4	Mode PWM pel llum de la GLCD.	TRIS_CCP4 = 0 (OUT)
CCP5	Mode Compare per generar una interrupció.	TRIS_CCP5 = X (No afecta el port d'E/S)

6) Volem programar una funció bloquejant *DelayCCP (int micros)* que utilitzi la unitat de CCP per fer-ho amb la màxima precisió possible.

6.1 Quin mode de CCP triaries per aquesta tasca? Quin valor posaries a quin registre per configurar-ho així? (0,5p)
El mode compare, que compara un timer amb un valor, i d'entre els compare, el millor és el que posa el IF a 1 i no afecta els pins (ja que no cal). Per tant CCP1CON=xxxx1010

6.2 Tria un Timer per associar-lo a la unitat de CCP. Indica quin valor posaries a quin registre. (0,5p)
Serveix qualsevol timer (dels associats a CCP, no el 0). Triem per exemple el Timer 1 modificant el resgistre de selecció. CCPTMRS0 = xxxxxx00.

Nom i Cognoms: _____POSSIBLE SOLUCIÓ_____

6.3 Si treballem amb un oscil·lador de **16MHz**, com configuraries el Timer triat perquè la seva base de temps sigui de 1 us (1 microsegon)? (0,5 p)

En aquest cas voldrem que el Timer1 compti microsegons. La única combinació possible amb un Fosc de 16MHz és triar Fosc/4 com a base i posar un prescaler de 4. De passada, el posem a 16 bits per tenir més rang.

T1CON = 00100x10

6.4 Escriu **(al darrera)** el codi de la funció *DelayCCP (int micros)* que no surti fins que hagi passat el temps indicat a *micros* (no volem utilitzar interrupcions per això). (1,5p)

Com que el Timer compta microsegons i la unitat de CCP en mode compare compara el valor del Timer amb el registre CCPR1, és tan fàcil com:

```
void DelayCCP (int micros)
{
        CCPR1=micros;
        TMR1=0;          // no sabem com podia estar...
        T1CON |=1;       // engeguem si no ho està...
        while (!CCP1IF); // esperem
        CCP1IF=0;
}
```

Si considerem que micros pot ser un valor gran (major que int), caldrà comptar diversos blocs de CCPIF (per la divisió i el residu).