

Cognoms, Nom _____ DNI _____

És obligatori justificar la resposta dels problemes en aquest mateix full !

P1. (2 punts) Un estudiant fa una crida a una subrutina anomenada **smart_crash** programada a un PIC18F45K22. Indica durant quant de temps s'executarà la subrutina abans de que es produeixi un desbordament de la pila si Fosc=10Mhz. Suposeu que en el moment de fer la crida la pila està buida.

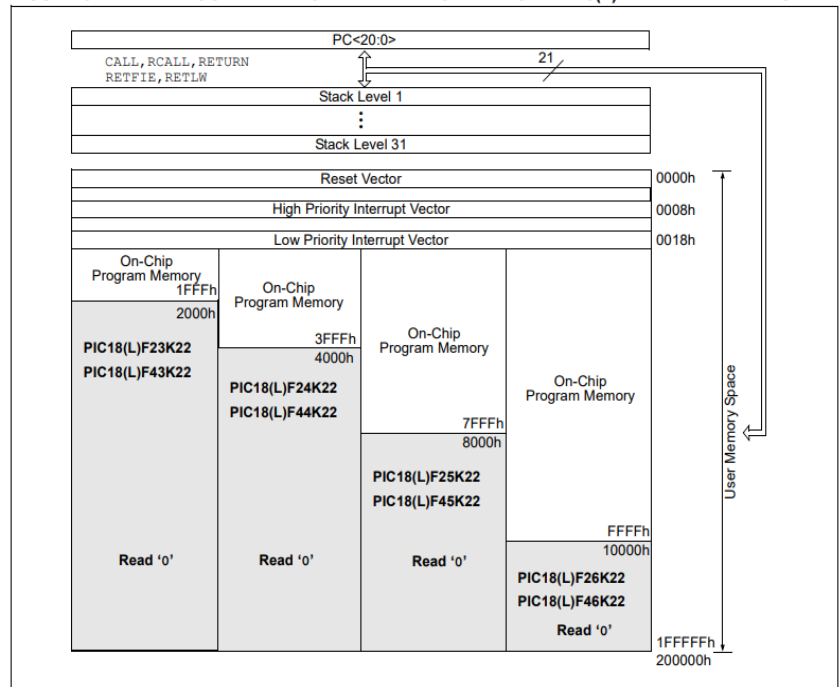
smart_crash

```
MOVLW 0xFF ;1c
BTFSC WREG, 0 ; no salta 1c
CALL smart_crash ;2 cicles
RETURN ;no s'executa
```

La pila del PIC18F45K22 té 31 nivells.
A cada crida a smart_crash es guarda el PC i es produirà un stack overflow quan fem la crida número 32.
La primera vegada que cridem smart_crash omple el nivell 1 de la pila. Falten 31 crides més:

$$4\text{cicles} \times 31 \times 4 / 10\text{MHz} = 49,6\mu\text{s}$$

FIGURE 5-1: PROGRAM MEMORY MAP AND STACK FOR PIC18(L)F2X/4XK22 DEVICES



P2. (2 punts) Omple la taula amb els valors resultants dels registres després d'executar el següent codi:

```
val equ 0x33
MOVLW val ;WREG=0x33
BTFSS WREG, 1 ;salta
CLRF 00, A ;no s'executa
CLRF 00, B ;[100]=0
MOVLB 0 ;BSR=0
INCF 00, W, A ;WREG=[000]+1
INCF 00, F, B ;[000]=[000]+1
BTFSC WREG, 7 ;salta
MOVFF 100h, 200h ;no s'executa
```

Reg	Valor inicial	Valor Final
WREG	00h	05h
BSR	1	00h
000h	04h	05h
100h	FFh	00h
200h	00h	00h

Quant ocupa el codi a memòria de programa?

8 instruccions single word + 1 instrucció 2-word = 10 Words = 20 Bytes = 160bits

Cognoms, Nom _____ DNI _____

És obligatori justificar la resposta dels problemes en aquest mateix full !

P3. (1 punt) Completeu el següent codi per a que sumi les **N** ($N \leq 255$) primeres posicions de memòria del banc 1 i deixi el resultat en la posició de memòria de dades 000h (LSB de la suma) i 001h (MSB de la suma).

```
N equ x ; x ≤ 255
sum_hi equ 01
sum_lo equ 00
i equ 02 ;iterator
CLRF sum_lo, A
CLRF sum_hi, A
CLRF i, A ;start from 00
LFSR FSR0, 0x100
```

loop

```
MOVF POSTINC0, W
ADDWF sum_lo, F, A
MOVLW 0
ADDWFC sum_hi, F, A
INCF i, F, A ; increment loop index
MOVLW N
CPFSGT i, A ; it is done when i>N
BRA loop
```

end bra end

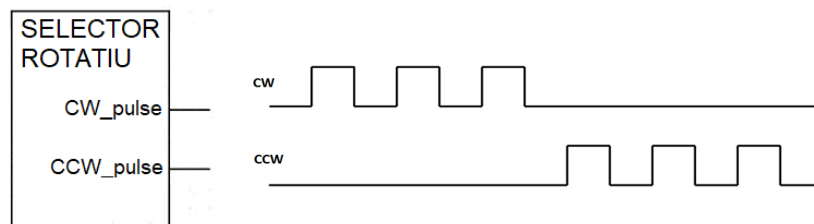
Hem de fer servir direccionament indirecte per a poder

recórrer la memòria en un bucle

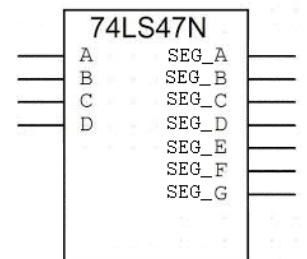
Cognoms, Nom _____ DNI _____

És obligatori justificar la resposta dels problemes en aquest mateix full !

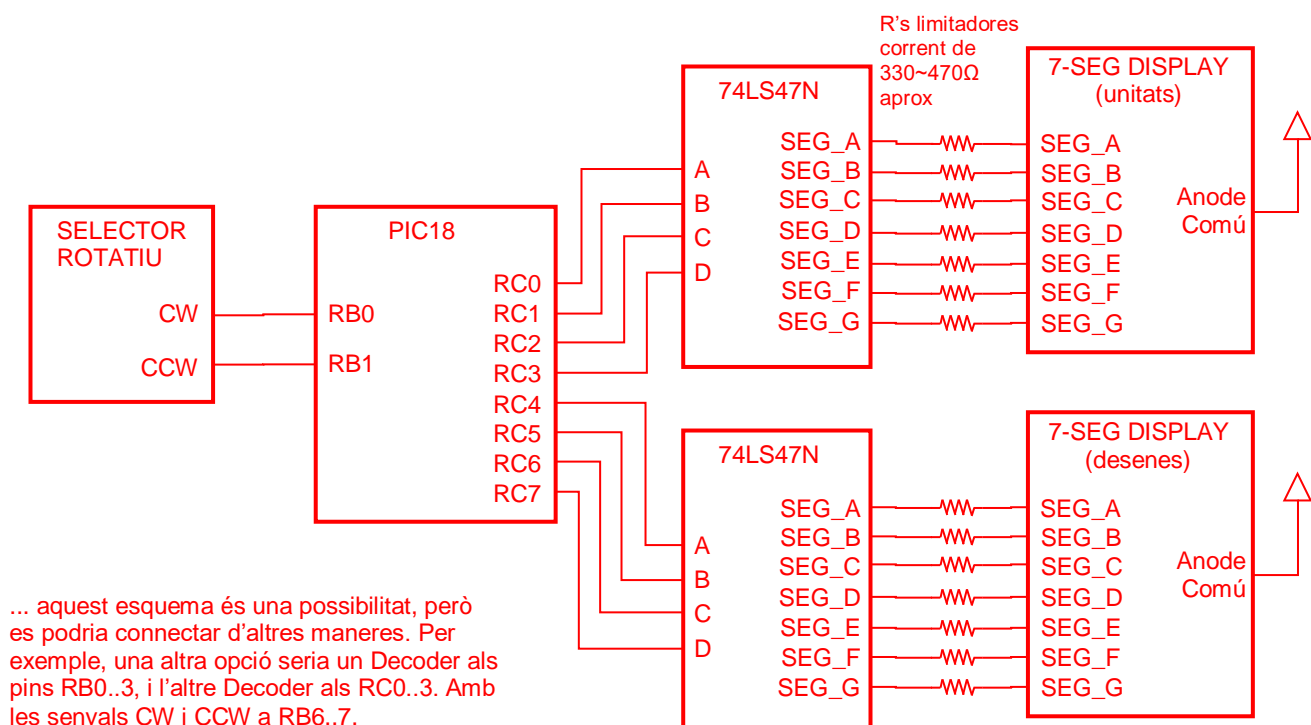
P4. (3 punts) La UPC vol renovar tots els microones que hi ha al Campus Nord per un nou model dissenyat a la mateixa universitat. A CI ens ha tocat dissenyar el subsistema de selecció del temps. Usarem una rodeta de selecció que ens permetrà saber si hem d'incrementar o decrementar 1 segon el temps del comptador. La rodeta té un component electrònic que ens dona 2 pins de sortida: un pin ens donarà els polsos en cas de gir horari (CW, clockwise), i l'altre pin ens donarà polsos en cas de gir antihorari (CCW, counterclockwise).



El nostre microones model **UPCWave** serà tant potent que només caldrà que compti 59 segons com a màxim. El temps el mostrarem amb 2 displays de 7-segments, de tipus Ànode Comú. A diferència del laboratori, no usarem un pin del PIC per controlar cada segment, sinó que utilitzarem un xip especial anomenat "Decodificador de BCD a 7-Segments" (referència 74LS47N). Al decodificador li passem un codi en binari de 4 bits, que representi el valor en decimal de la xifra que volem mostrar (això s'anomena BCD: Binary Coded Decimal). El decodificador activarà adequadament la sortida de 7 pins per controlar els 7 segments. El pin A és el LSB del codi BCD. Usarem dos xips 74LS47N, un per cada display.



- 1) Dibuixeu un esquema amb els components i les connexions que seran necessàries per llegir i mostrar la selecció del temps amb dues xifres mitjançant els 7-segments. Utilitzeu únicament pins del PortB i PortC. Indiqueu clarament els pins del microcontrolador als que connectareu els diferents components.



Cognoms, Nom _____ DNI _____

És obligatori justificar la resposta dels problemes en aquest mateix full !

- 2) Ompliu la següent subrutina de **inicialitzacioPorts()** en C, per a configurar adequadament els pins dels ports involucrats, per a fer funcionar el nostre sistema.

void inicialitzacioPorts() {

Configurem els ports seguint l'esquema dibuixat abans.
Altres esquemes requeririen configuracions diferents!

```
ANSELB = ANSELB & 0xFC; // port B pins 0 i 1 com a Digital I/O
ANSELC = 0; // tot port C com a Digital I/O
TRISBbits.TRISB0 = 1; // pin RB0 com a Input
TRISBbits.TRISB1 = 1; // pin RB1 com a Input
TRISC = 0; // tot Port C com a Output
}
```

- 3) Escriviu una funció en codi C que mostri pels displays el valor del comptador de segons (0 <= segons <= 59) usant els xips 74LS47N connectats al nostre micro:

void showSeconds (int segons) {

Codi basat en l'esquema dibuixat abans.
Altres esquemes requeririen codis diferents, per deixar
les unitats i desenes en altres pins del micro!

```
int unitats, desenes;
unitats = segons % 10;
desenes = segons / 10;

LATC = unitats + desenes << 4; // deixem unitats als pins RC0..3, i desenes als RC4..7
// també podríem haver fet: LATC = unitats + desenes * 16;
// o bé: LATC = unitats | desenes << 4;
}
```

Cognoms, Nom _____ DNI _____

És obligatori justificar la resposta dels problemes en aquest mateix full !

P5. (2 Punts) Es vol gestionar l'activació periòdica d'una subrutina **MySub()** cada **DNI** microsegons aproximadament, on la variable **DNI** correspon al número que figura en el vostre document d'identitat o passaport. S'ha connectat un circuit oscil·lador que genera 30 polsos per segon a l'entrada del pin RB2 i es vol que es generi una interrupció de baixa prioritat associada cada flanc de pujada.

Es demana:

- 1) Indiqueu com queden afectats els bits dels registres de configuració RCON, INTCON, INTCON2, INTCON3 i TRISB.

Hi hauria dues possibles solucions: via la interrupció INT2 que està connectada a RB2 o via Port B Interrupt on Change. Aquesta ultima opció, Port B Interrupt on change només funciona amb els bits 4 a 7, i a més donaria interrupcions en el flanc de pujada i de baixada.

Via INT2, els bits afectats són:

```
RCONbits.IPEN = 1; // habilitem prioritats alta i baixa
INTCON2bits.INTEDG2 = 1; // interrupció en flanc de pujada
INTCON3bits.INT2IP = 0; // INT2 interrupció de baixa prioritat
INTCON3bits.INT2IE = 1; // habilitem INT2
TRISBbits.RB2 = 1; // Port B pin 2 d'entrada
INTCONbits.GIEL = 1; // habilitem baixes
INTCONbits.GIEH = 1; // habilitem interrupcions
```

- 2) Completeu el codi de la RSI següent per a que la subrutina **MySub()** s'executi aproximadament cada **DNI** µseg.

```
unsigned int Count;
```

```
void interrupt low_priority LowIsr(void) //Low priority interrupt
```

```
{
    if(INTCON3bits.INT2IF && INTCON3bits.INT2IE) // Si INT2 està habilitada i s'ha produït
    {
        Count++;
        if (Count == ((DNI*30)/1000000) ) // ja han passat DNI/1000000 segons
        {
            MySub(); // cridem a la subrutina
            Count = 0; // tornem a comptar
        }
        INTCON3bits.INT2IF = 0; // baixem el flag
    }
}
```

- 3) Si en el codi del programa principal volem posar en pausa l'execució periòdica de la subrutina **My_Sub**, de quines maneres ho podríeu fer?

La manera més senzilla és deshabilitar les interrupcions INT2 incloent la instrucció

```
INTCON3bits.INT2IE = 0;
```

Una altre manera enginyosa de fer-ho seria configurant el bit RB2 de sortida **TRISBbits.RB2 = 0**; En aquest cas cal posar una resistència per a que no es produeixi un corrent excessiu en RB2 al coincidir dues sortides amb valors lògics diferents.

Cognoms, Nom _____ DNI _____

És obligatori justificar la resposta dels problemes en aquest mateix full !

Altres maneres, serien deshabilitar les interrupcions de baixa prioritats (GIEL), deshabilitar prioritats (IPEN) o deshabilitar totes les interrupcions (GIEH).