

CODI SOBRE L2b

```
#define _XTAL_FREQ 8000000

#include <xc.h>

#include "config.h"

void configPIC(){

    ANSELA=0x00; // Tots els bits a digital

    ANSELB=0x00;

    ANSELD=0x00;

    TRISA=0x00; //RA0 output

    TRISB=0x1F; //RB0, RB1, RB2 input

    TRISD=0x00; //tots són outputs

    PORTA = 0x00; //netegem els ports

    PORTB = 0x00;

    PORTD = 0x00;

}

void main(void)

{

    configPIC();

    int numero = 0; //el número a mostrar pel 7-segments, un int perquè no farà falta posar-ho en els leds

    int comptador = 4; //el comptador permet saber quin 7-segments encendre, és un int perquè no farà falta posar-ho als leds

    char config7segments[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0xFF, 0x67}; //números del 0 al 9 en hexadecimal, hem escollit un char ja que són 8 bits, perfectes per indicar un número

    int prevRB[3] = {0, 0, 0}; //permetran saber l'estat dels pins RB0 i RB1 del bucle anterior

    char encendre_7_segments[4] = {0x08, 0x04, 0x02, 0x01};

    while(1)

    {

        if (PORTBbits.RB2 == 1) PORTAbits.RA0 = 0; //apagar pantalla si RB2 = 1
```

```
else if (PORTBbits.RB4 == 1) { //apagar tots els segments si RB4 = 1

    PORTAbits.RA0 = 1;

    PORTAbits.RA1 = 1;

    PORTAbits.RA2 = 1;

    PORTAbits.RA3 = 1;

}

else {

    if (prevRB[0] == 0 && PORTBbits.RB0 == 1 && numero < 9 ) { //sumem 1 si al bucle anterior RB0 era
0, ara és 1, i el numero no es major igual a 9

        ++numero;

    }

    if (prevRB[1] == 0 && PORTBbits.RB1 == 1 && numero > 0) { //restem 1 si al bucle anterior RB1 era 0,
ara és 1, i el numero no es menor igual a 0

        --numero;

    }

    if (prevRB[2] == 0 && PORTBbits.RB3 == 1) {

        if (comptador == 1) comptador = 4;

        else --comptador;

    }

    PORTD = config7segments[numero]; //posem al 7-segments el numero que toca

    PORTA = encendre_7_segments[comptador-1]; //per saber quin set_segments encendre

    prevRB[0] = PORTBbits.RB0; //actualitzem l'estat de prevRB0 per a la següent iteració

    prevRB[1] = PORTBbits.RB1; //actualitzem l'estat de prevRB1 per a la següent iteració

    prevRB[2] = PORTBbits.RB3; //actualitzem l'estat de prevRB2 per a la següent iteració

    __delay_ms(200.000);

}

}

}
```