

CODI L3b INTERRUPCIONS

```
#define _XTAL_FREQ 8000000
```

```
#define seed 1234
```

```
#include <xc.h>
```

```
#include <stdlib.h>
```

```
#include "config.h"
```

```
char encendre_el[4] = {0x3F, 0x3F, 0x3F, 0xBF}; //indica quin 7-segment tindrà el dp. Un char perquè amb 8 bits ja tenim prou per indicar quin tindrà el dp activat
```

```
char numeros[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x67}; //números del 0 al 9 en hexadecimal, hem escollit un char ja que són 8 bits, perfectes per indicar un número
```

```
char encendre_7_segments[4] = {0x08, 0x04, 0x02, 0x01}; //indica quin 7-segments cal encendre, un char perquè amb 4 dels 8 bits ja tenim prou i no cal més
```

```
uint8_t numero = 0; //indicarà quin número del 0 al 9 sortirà al 7-segments. Un uint8_t perquè ja ens permetrà veure els números del 0 al 9 de sobres
```

```
uint8_t activat = 3; //el de més a la dreta
```

```
void interrupt service_routine_HighP (void){
```

```
    if (INTCONbits.INT0IF && INTCONbits.INT0IE) {
```

```
        INTCONbits.INT0IF=0;
```

```
        encendre_el[activat] = numeros[(0 + rand() % 9)] + 0x80;
```

```
    }
```

```
}
```

```
void interrupt low_priority service_routine_LowP (void){
```

```
    if (INTCON3bits.INT1IF && INTCON3bits.INT1IE) {
```

```
        INTCON3bits.INT1IF=0;
```

```
        if(activat == 3) {
```

```
            encendre_el[0] = encendre_el[3]; //movem el número a l'esquerra
```

```
            encendre_el[3] = 0x3F; //posem un 0
```

```
            activat = 0;
```

```
        }
```

```
    else {
```

```
        encendre_el[activat+1] = encendre_el[activat]; //movem el número a la dreta
```

```
        encendre_el[activat] = 0x3F; //posem un 0
```

```
        ++activat;
```

```
    }
```

```
}
```

```
if (INTCON3bits.INT2IF && INTCON3bits.INT2IE) {
```

```
    INTCON3bits.INT2IF=0;
```

```
    if(activat == 0) {
```

```
        encendre_el[3] = encendre_el[0]; //movem el número a la dreta
```

```
        encendre_el[0] = 0x3F; //posem un 0
```

```
        activat = 3;
```

```
    }
```

```
    else {
```

```
        encendre_el[activat-1] = encendre_el[activat]; //movem el número a l'esquerra
```

```
        encendre_el[activat] = 0x3F; //posem un 0
```

```
        --activat;
```

```
    }  
}  
return;  
}  
  
void encendre_leds(){  
    PORTD = 0x00;  
    __delay_ms(1);  
    PORTA = encendre_7_segments[3]; //per saber quin set_segments encendre  
    PORTD = encendre_el[3]; //posem al 7-segments les unitats  
    __delay_ms(1);  
    PORTD = 0x00;  
  
    __delay_ms(1);  
    PORTA = encendre_7_segments[2]; //per saber quin set_segments encendre  
    PORTD = encendre_el[2]; //posem al 7-segments les decenes  
    __delay_ms(1);  
    PORTD = 0x00;  
  
    __delay_ms(1);  
    PORTA = encendre_7_segments[1]; //per saber quin set_segments encendre  
    PORTD = encendre_el[1]; //posem al 7-segments les centenes  
    __delay_ms(1);  
    PORTD = 0x00;  
  
    __delay_ms(1);  
    PORTA = encendre_7_segments[0]; //per saber quin set_segments encendre  
    PORTD = encendre_el[0]; //posem al 7-segments els milers  
    __delay_ms(1);  
    PORTD = 0x00;  
}  
  
void configPIC(){  
    ANSELA = 0x00; // Tots els bits a digital  
    ANSELB = 0x00;  
  
    TRISA = 0x00; //RA0, RA1, RA2, RA3 output  
    TRISD = 0x00; //tots són outputs  
  
    PORTA = 0x00; //netejem els ports  
    PORTD = 0x00;  
  
    //interrupcions  
    ANSELB &= 0xF8; //INT0, INT1, INT2 digital  
    TRISB |= 0x07; //INT0, INT1, INT2 inputs  
    RCONbits.IPEN = 1; //activem interruptor de prioritat  
    INTCON = 0xD0; //activem el bit 7 'GIEH', el bit 6 'GIEL' i el bit 4 'INT0IE'  
    INTCON3 = 0x18; //activem bit 4 'INT2IE' i el bit 3 'INT1IE'  
  
    INTCON2 |= 0x81; // configura INT1 per ser flanc de pujada i configura INT2 per ser flanc de pujada  
}
```

```
void main(void){  
    configPIC();  
    srand(seed);  
    while(1) {  
        encendre_leds();  
        __delay_ms(15);  
    }  
}
```