

CODI DEL PREVI DEL L3a

```
/* Main.c file generated by New Project wizard
```

```
*
```

```
* Created: dj. oct. 17 2024
```

```
* Processor: PIC18F45K22
```

```
* Compiler: MPLAB XC8
```

```
*/
```

```
#define _XTAL_FREQ 8000000
```

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

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

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

```
//Declaració variables globals
```

```
char config7segments[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
```

```
int prevRC[4] = {0, 0, 0, 1}; //permetran saber l'estat dels pins RC4, RC5, RC6 i RC7 del bucle anterior, enters ja  
que no cal fer-los servir per indicar cap número al 7-segments, RC5 a 1 perquè el seu estat natural és un 1 lògic  
unsigned long counter = 0; //comptador que permet generar semilles aleatòries tota l'estona, un unsigned long  
perquè volem que pugui almagatzemar número molts grans
```

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

```
int aux = 0; //per anar fent el mòdul del numero
```

```
void configPIC(){
```

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

```
    ANSELC=0x00;
```

```
    ANSELD=0x00;
```

```
    TRISA=0x00; //RA0, RA1, RA2, RA3 output
```

```
    TRISC=0xF0; //RC4, RC5, RC6, RC7 input
```

```
    TRISD=0x00; //tots són outputs
```

```
    PORTA = 0x00; //netejem els ports
```

```
    PORTC = 0x00;
```

```
    PORTD = 0x00;
```

```
}
```

```
void config_flancs(){
```

```
    prevRC[0] = PORTCbits.RC4; //actualitzem l'estat de prevRC4 per a la següent iteració
```

```
    prevRC[1] = PORTCbits.RC5; //actualitzem l'estat de prevRC5 per a la següent iteració
```

```
    prevRC[2] = PORTCbits.RC6; //actualitzem l'estat de prevRC6 per a la següent iteració
```

```
    prevRC[3] = PORTCbits.RC7; //actualitzem l'estat de prevRC7 per a la següent iteració
```

```
}
```

```
void encendre_leds(){
```

```
    aux = numero; //número auxiliar per no perdre l'original
```

```
    PORTD = config7segments[aux%10]; //posem al 7-segments les unitats
```

```
    aux = aux / 10; //dividim entre 10 per després agafar les desenes
```

```
    PORTA = encendre_7_segments[3]; //per saber quin set_segments encendre
```

```
    __delay_ms(1);
```

```
    PORTD = 0x00;
```

```
    PORTD = config7segments[aux%10]; //posem al 7-segments les desenes
```

```
    aux = aux / 10;
```

```
    PORTA = encendre_7_segments[2]; //per saber quin set_segments encendre
```

```
    __delay_ms(1);
```

```
PORTD = 0x00;

PORTD = config7segments[aux%10]; //posem al 7-segments les centenes
aux = aux / 10;
PORTA = encendre_7_segments[1]; //per saber quin set_segments encendre
__delay_ms(1);
PORTD = 0x00;

PORTD = config7segments[aux%10]; //posem al 7-segments els milers
PORTA = encendre_7_segments[0]; //per saber quin set_segments encendre
__delay_ms(1);
PORTD = 0x00;
}

void main(void){
    configPIC();
    while(1) {
        if (PORTCbits.RC6 == 1 && prevRC[2] == 0) { //al detectar flanc de pujada a RC6 es genera número aleatori
            srand(counter);
            numero = 0 + rand() % 9999;
        }

        if (PORTCbits.RC7 == 0 && prevRC[3] == 1) { //al detectar flanc de baixada a RC7 es posa a 0 els
7-segments
            numero = 0;
        }

        if (PORTCbits.RC5 == 1 && prevRC[1] == 0) { //al detectar flanc de pujada a RC5 sumem una unitat al
número
            if (numero < 9999) ++numero;
        }

        if (PORTCbits.RC4 == 1 && prevRC[0] == 0) { //al detectar flanc de pujada a RC4 restem una unitat al
número
            if (numero > 0) --numero;
        }

        encendre_leds();
        config_flancs();
        ++counter;
        __delay_ms(40);
    }
}
```