

Labrotorio #1

Link de github:

<https://github.com/Helder1121/Labsdigitaldos>

Progra comentada:

/*

* File: lab1.c

* Author: Helder Ovalle

*

* Created on 30 de enero de 2021, 02:55 PM

*/

//*****

// Importación de librerías

//*****

#include <xc.h>

```
//*****  
*****
```

// Palabra de configuración

```
//*****  
*****
```

// CONFIG1

#pragma config FOSC = XT // Oscillator Selection bits (XT oscillator:
Crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN)

#pragma config WDTE = OFF // Watchdog Timer Enable bit (WDT
disabled and can be enabled by SWDTEN bit of the WDTCON register)

#pragma config PWRTE = OFF // Power-up Timer Enable bit (PWRT
disabled)

#pragma config MCLRE = OFF // RE3/MCLR pin function select bit
(RE3/MCLR pin function is MCLR)

#pragma config CP = OFF // Code Protection bit (Program memory code
protection is disabled)

#pragma config CPD = OFF // Data Code Protection bit (Data memory
code protection is disabled)

#pragma config BOREN = OFF // Brown Out Reset Selection bits (BOR
disabled)

#pragma config IESO = OFF // Internal External Switchover bit
(Internal/External Switchover mode is disabled)

#pragma config FCMEN = OFF // Fail-Safe Clock Monitor Enabled bit (Fail-
Safe Clock Monitor is disabled)

#pragma config LVP = OFF // Low Voltage Programming Enable bit (RB3
pin has digital I/O, HV on MCLR must be used for programming)

// CONFIG2

#pragma config BOR4V = BOR40V // Brown-out Reset Selection bit
(Brown-out Reset set to 4.0V)

#pragma config WRT = OFF // Flash Program Memory Self Write Enable
bits (Write protection off)

//*****

// Variables

//*****

#define XTAL FREQ 8000000

#define LED_ROJO PORTEbits.RE0

#define LED_AMARILLO PORTEbits.RE1

#define LED_VERDE PORTEbits.RE2

//Definiendo las variables del semaforo y el delay para su uso

unsigned char ganador = 0;

unsigned char J1 = 0;

unsigned char J2 = 0;

//variables tipo char

//J1 y J2 las utilizare para el incremento de bits multiplicando x2

//ganador indicando el ganangodr de los dos jugadores

//*****

// Prototipos de funciones

```
//*****  
*****
```

void semaforo(void);

void setup(void);

void contador(void);

```
//*****  
*****
```

// Ciclo principal

```
//*****  
*****
```

void main(void) {

```
//*****  
*****
```

// Loop principal

```
//*****  
*****
```

setup();

while (1) {

if (PORTAbits.RA0 == 0){ //PushB del semaforo (antirebote)}

delay_ms(50);

if (PORTAbits.RA0 == 1) {

```

    semaforo();

    contador();

    //Llamo a los void

    }

}

}

}

//*****
*****

// Configuración

//*****
*****

void setup(void){

    TRISE = 0;

    PORTE = 0;

    ANSEL = 0;

    ANSELH = 0;

    //Steo todo el puerto E donde esta el semaforo para que no se queden on

    TRISA = 0b00000111;

    //Estan en uno porque son de los push del puerto A

    PORTA = 0;

    TRISC = 0;

    PORTC = 0;

    TRISD = 0;

    PORTD = 0;

```

//Steo puerto c y d de los contadores de los jugadores

TRISB = 0;

PORTB = 0;

}

//*****

// Funciones

//*****

//Void del incio del semaforo

void semaforo(void) {

PORTC = 0;

PORTD = 0;

PORTB = 0;

//steo los puertos

LED ROJO = 1;

delay ms(500);

LED ROJO = 0;

LED AMARILLO = 1;

delay ms(500);

LED AMARILLO = 0;

LED VERDE = 1;

delay ms(500);

LED VERDE = 0;

```
ganador = 1;  
//declaro las variables del semaforo con sus delays para el incio de la carrera  
}
```

```
//Void contador para corrimiento de los leds, que se reinicien e indiquen al  
puerto
```

```
//B el ganador
```

```
void contador(void) {
```

```
while (ganador == 1){
```

```
//La variable de ganador hara entrar al ciclo while
```

```
if (PORTAbits.RA1 == 0){
```

```
delay_ms(50);
```

```
if(PORTAbits.RA1 == 1) {
```

```
//Antirebote
```

```
//Jugador 1
```

```
if (PORTC == 0){
```

```
J1 = 0b00000001;
```

```
PORTC = J1;
```

```
//Primera fila de leds si empieza en cero
```

```
}
```

```
else if (PORTC != 0){
```

```
J1 = J1*2;
```

```
PORTC = J1;
```

```
//Corrimiento
```

```
}
```

```

_____ if (PORTCbits.RC7 == 1){
_____     ganador = 0;
_____     PORTBbits.RB0 = 1;
_____     //Cuando el ultimo led del J1 se prenda se marcara el led del
_____     //RBO que es del J1
_____ }
_____ }
_____ }
_____ if (PORTAbits.RA2 == 0){
_____     delay_ms(50);
_____     if(PORTAbits.RA2 == 1) {
_____         //Antirebote
_____         //Jugador 2
_____         if (PORTD == 0){
_____             J2 = 0b00000001;
_____             PORTD = J2;
_____             //Primera fila de leds si empiezan en cero
_____         }
_____         else if (PORTD != 0){
_____             J2 = J2*2;
_____             PORTD = J2;
_____             //Corrimiento
_____         }
_____         if (PORTDbits.RD7 == 1){

```


