01/02/2021
der Ovalle Barrios
18349

Universidad Del Valle De Guatemala

Digital 2 Helder Ovalle Barrios

Sección: 18349

## Labrotorio #1

## Link de github:

https://github.com/Helder1121/Labsdigitaldos

## 

#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)
```

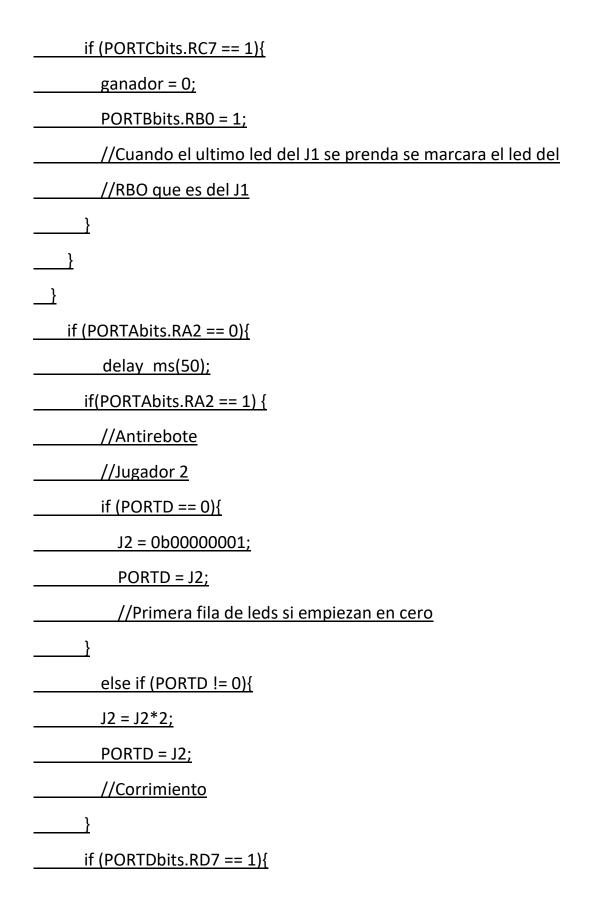
#pragma config BOR4V = BOR40V // Brown-out Reset Selection bit
(Brown-out Reset set to 4.0V)
#pragma config WRT = OFF // Flash Program Memory Self Write Enable
bits (Write protection off)
*******
// Variables
******
#define XTAL FREQ 8000000
#define LED_ROJO PORTEbits.RE0
#define LED_AMARILLO PORTEbits.RE1
#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

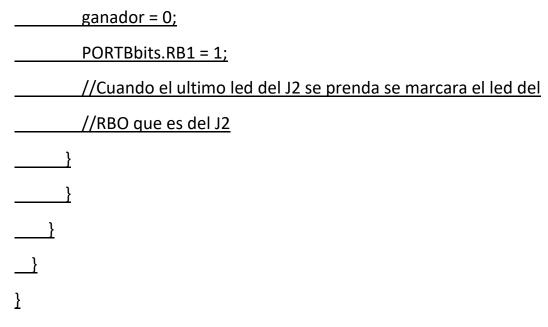
//************************************
*********
void semaforo(void);
void setup(void);
void contador(void);
//*************************************
********
// Ciclo principal
<u></u>
********
void main(void) {
//************************************
******
// Loop principal
******
setup();
<u>while (1) {</u>
if (PORTAbits.RA0 == 0){ //PushB del semaforo (antirebote))
delay_ms(50);
<u>if (PORTAbits.RA0 == 1) {</u>

semaforo();
contador();
//Llamo a los void
}
}
}
<u>}</u>
//*****************
*******
// Configuración
//*******************
*******
<pre>void setup(void){</pre>
TRISE = 0;
<u>PORTE = 0;</u>
<u>ANSEL = 0;</u>
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
<u>PORTA = 0;</u>
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;
```

```
<u>ganador = 1;</u>
//declaro las variables del semaforo con sus delays para el incio de la carrera
}
//Void contador para corrimiento de los leds, que se reincien e indiquen al
<u>puerto</u>
//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
  <u>if (PORTC == 0){</u>
   J1 = 0b00000001;
   PORTC = J1;
   //Primera fila de leds si empieza en cero
 else if (PORTC != 0){
 J1 = J1*2;
 PORTC = J1;
 //Corrimiento
```





## Diagrama de flujo:

