Allison Estuardo Aguilar Chocooj --- 18834
Universidad del Valle de Guatemala

Laboratorio #1

Código Comentado

```
//******************************************************************************
//Autor: Allison Chocooj 18834
//******************************************************************************
// 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 digital input, MCLR internally tied to VDD)
#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)
```

Allison Estuardo Aguilar Chocooj --- 18834
Universidad del Valle de Guatemala

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

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

// Variables

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

#define LEDROJO PORTEbits.RE0

#define LEDAMARILLO PORTEbits.RE1

#define LEDVERDE PORTEbits.RE2

#define _XTAL_FREQ 8000000

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

// Prototipos de funciones

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

void Setup(void);

void Semaforo(void);

char LED1 [8]={0b10000000,0b01000000,0b00100000,0b00010000,0b00001000,0b00000100,0b00000010,0b00000001};//leds de jugador 1

//en esta variable puedo definir que estados tendra a la hora de cambiar de posicion

char LED2 [8]={0b10000000,0b01000000,0b00100000,0b00010000,0b00001000,0b00000100,0b00000010,0b00000001};//leds de jugador 2

//en esta variable puedo definir que estados tendra a la hora de cambiar de posicion

char Jugador1 = 0;//para saber la posicion del jugador 1 y pueda contener a LED1

char Jugador2 = 0;//para saber la posicion del jugador 2 y pueda contener a LED2

int carrerita (unsigned int C); //es una variable de asignacion a C



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

// Configuración

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

void Setup(void){
```

Allison Estuardo Aguilar Chocooj --- 18834
Universidad del Valle de Guatemala

```
ANSEL = 0;

ANSELH = 0;

//aca declaro que el puerto E sera salida para el semaforo

TRISEbits.TRISE0 = 0; //led rojo

TRISEbits.TRISE1 = 0; //led amarillo

TRISEbits.TRISE2 = 0; //led verde

PORTE = 0;


//aca declaro que los primero 3 puertos del A seran mis

//entradas para los botonoes

PORTAbits.RA1 = 1; //entrada de boton 1

PORTAbits.RA2 = 1; //entrada de boton 2

PORTAbits.RA3 = 1; //entrada de boton 3


//aca declaro que los primeros 2 puertos del B seran mis

//salidas para el led de los ganadores

TRISBbits.TRISB0 = 0; //led de ganador 1

TRISBbits.TRISB1 = 0; //led de ganador 2

PORTB = 0;


//aca declaro los leds para la carrera de jugador 1

TRISCbits.TRISC0 = 0; //led 1

TRISCbits.TRISC1 = 0; //led 2

TRISCbits.TRISC2 = 0; //led 3

TRISCbits.TRISC3 = 0; //led 4

TRISCbits.TRISC4 = 0; //led 5

TRISCbits.TRISC5 = 0; //led 6

TRISCbits.TRISC6 = 0; //led 7

TRISCbits.TRISC7 = 0; //led 8
```

```
   PORTC = 0;


   //aca declaro los leds para la carrera de jugador 2
   TRISDbits.TRISD0 = 0; //led 1
   TRISDbits.TRISD1 = 0; //led 2
   TRISDbits.TRISD2 = 0; //led 3
   TRISDbits.TRISD3 = 0; //led 4
   TRISDbits.TRISD4 = 0; //led 5
   TRISDbits.TRISD5 = 0; //led 6
   TRISDbits.TRISD6 = 0; //led 7
   TRISDbits.TRISD7 = 0; //led 8
   PORTD = 0;


}
//sera el modulo de semaforo que hará el conteo del semafor para poder iniciar
//la carrera pero primero pongo a jugador 1 y 2 en 0 antes de iniciar el conteo
void Semaforo(void){
   Jugador1 = 0;
   Jugador2 = 0;
   PORTD = LED2[Jugador2];
   PORTC = LED1[Jugador1];


   LEDROJO = 1;
    __delay_ms(250);
    LEDROJO = 0;


    LEDAMARILLO = 1;
    __delay_ms(250);
    LEDAMARILLO = 0;
```

Allison Estuardo Aguilar Chocooj --- 18834
Universidad del Valle de Guatemala

```
    LEDVERDE = 1;
    __delay_ms(250);
    LEDVERDE = 0;
}



//la variable de asignacion para C donde ire sumando jugador1 en +1 con lo que
//a la hora que defini el char led1 con variables ya establecidas este ira
//moviendose en estos y no contar de forma binaria para que el conteo se
//de forma secuancial y lo mismo se hace para el jugador 2 para la carrera 2
//a cada uno se le tiene un antirrebote con 50 milisegundo para
//evitar que salte de un solo a otra posicion y vaya pasando de uno a uno
   int carrerita (unsigned int C){
      if (C == 1){
         while(1){
         if (PORTAbits.RA1 == 0){
            __delay_ms(50);
            if (PORTAbits.RA1 == 1){
               //Programa para botón 1
               Jugador1++;
               PORTC = LED1[Jugador1];
               if (Jugador1 == 8){
                  return (1);
                }
              }
            }
         if (PORTAbits.RA2 == 0){
            __delay_ms(50);
```

```
            if (PORTAbits.RA2 == 1){
                //Programa para botón 1
                Jugador2++;
                PORTD = LED2[Jugador2];
                if (Jugador2 == 8){
                    return (2);
                  }
                }
              }
            }

      }
    }

    //en este pequeño modulo hago un pequeño analisis de quien fue el primero
    //en llegar con lo que verifico si llegaron a la meta y encender un led
    //en uno de los peurtos del B
    void ganador (void){
      if (Jugador1 == 8){
          PORTB = 0b00000001;
          }
      if (Jugador2 == 8){
          PORTB = 0b00000010;
          }
    }

//**********************************************************************************
// Ciclo principal
//**********************************************************************************
```

Allison Estuardo Aguilar Chocooj --- 18834
Universidad del Valle de Guatemala

```
void main(void) {
   Setup();
   //*****************************************************************************
   // Loop principal
   //*****************************************************************************
//tengo el ciclo principal del codigo donde hago un pequeño antirreboto a la hora
   //de hacer funcionar el semaforo para que este empiece el ciclo y se repita
   //llamando las funciones que tengo arriba y llamandolas en orden para que
   //se ejecuten una por una.
   while(1){
      if(PORTAbits.RA0 == 0){
         __delay_ms(50);
         if (PORTAbits.RA0 == 1){
         Semaforo();
         carrerita(1);
         ganador();
         }
      }
   }
}
```

Seudocódigo