|  |  |
| --- | --- |
| Universidad del Valle de Guatemala | Marco S. Alejandro Juárez Urrutia |
| Digital II | Carné 18516 |

**LABORATORIO #1**

\* File: Lab01.c

\* Author: Marco Juárez

\* Carné: 18516

\*/

//Librerías

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <xc.h>

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// 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)

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

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Variables

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#define LEDROJO PORTAbits.RA0

#define LEDAMARILLO PORTAbits.RA1

#define LEDVERDE PORTAbits.RA2

#define \_XTAL\_FREQ 8000000

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Prototipos de funciones

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Setup(void);

void Semaforo(void);

char J1 = 0;//Posicion inicial Jugador 1

char J2 = 0;//Posicion inicial Jugador 2

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

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

int run (unsigned int Cont); //Variable para conteo

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Configuración

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Setup(void){

ANSEL = 0;

ANSELH = 0;

//Puerto A como salida

TRISAbits.TRISA0 = 0; //led amarillo

TRISAbits.TRISA1 = 0; //led verde

TRISAbits.TRISA2 = 0; //led rojo

PORTA = 0;

//Puerto E como entrada de botones

PORTEbits.RE0 = 1;

PORTEbits.RE2 = 1;

PORTEbits.RE3 = 1;

//Puerto B como salida

TRISBbits.TRISB0 = 0; //Win J1

TRISBbits.TRISB1 = 0; //Win J2

PORTB = 0;

//Puerto C como salida

TRISCbits.TRISC0 = 0;

TRISCbits.TRISC1 = 0;

TRISCbits.TRISC2 = 0;

TRISCbits.TRISC3 = 0;

TRISCbits.TRISC4 = 0;

TRISCbits.TRISC5 = 0;

TRISCbits.TRISC6 = 0;

TRISCbits.TRISC7 = 0;

PORTC = 0;

//Puerto D como salida

TRISDbits.TRISD0 = 0;

TRISDbits.TRISD1 = 0;

TRISDbits.TRISD2 = 0;

TRISDbits.TRISD3 = 0;

TRISDbits.TRISD4 = 0;

TRISDbits.TRISD5 = 0;

TRISDbits.TRISD6 = 0;

TRISDbits.TRISD7 = 0;

PORTD = 0;

}

//Secuencia de semaforo

void Semaforo(void){

J1 = 0;

J2 = 0;

//Se enciende el led se toma un delay de 100ms y luego se apaga

LEDROJO = 1;

\_\_delay\_ms(100);

LEDROJO = 0;

LEDAMARILLO = 1;

\_\_delay\_ms(100);

LEDAMARILLO = 0;

LEDVERDE = 1;

\_\_delay\_ms(100);

LEDVERDE = 0;

PORTC = LED1[J1];

PORTD = LED2[J2];

}

//A la variable C se le ira sumando uno al momento de precionara el boton

//para tener un conteo de decada

int run (unsigned int Cont){

if (Cont == 1){

while(1){

if (PORTEbits.RE1 == 0){

//antirrebote

\_\_delay\_ms(50);

if (PORTEbits.RE1 == 1){

J1++;

PORTC = LED1[J1];

if (J1 == 8){

return (1);

}

}

}

if (PORTEbits.RE2 == 0){

\_\_delay\_ms(50);

if (PORTEbits.RE2 == 1){

J2++;

PORTD = LED2[J2];

if (J2 == 8){

return (2);

}

}

}

}

}

}

//rutina la cual revise el estado de los dos contadores para saber

//quién es el que llega antes a la meta y encender led.

void ganador (void){

if (J1 == 8){

PORTB = 0b00000001;

}

if (J2 == 8){

PORTB = 0b00000010;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Loop principal

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void main(void) {

Setup();

//Se configura para que al precionar el boton inicie el semaforo

//y luego llame a las otras funciones

while(1){

if(PORTEbits.RE0 == 0){

//Se implementa un antirrebote

\_\_delay\_ms(50);

if (PORTEbits.RE0 == 1){

//Se manda a llamar la funcion semaforo

Semaforo();

//Se manda a llamar la funcion run

run(1);

//Se manda a llamar la funcion ganador

ganador();

}

}

}

}

**Pseudocódigo Laboratorio #1**

* Se revisará si el botón esta presionado si lo está iniciará el semáforo.
* Habrán dos jugadores que tendrán su propio botón para avanzar para eso se crearán variables independientes para cada jugador que irán incrementando su valor.
* Al final se mostrará al ganador por medio de un led que lo indicará y el otro jugador ya no podrá avanzar.