Universidad del Valle de Guatemala

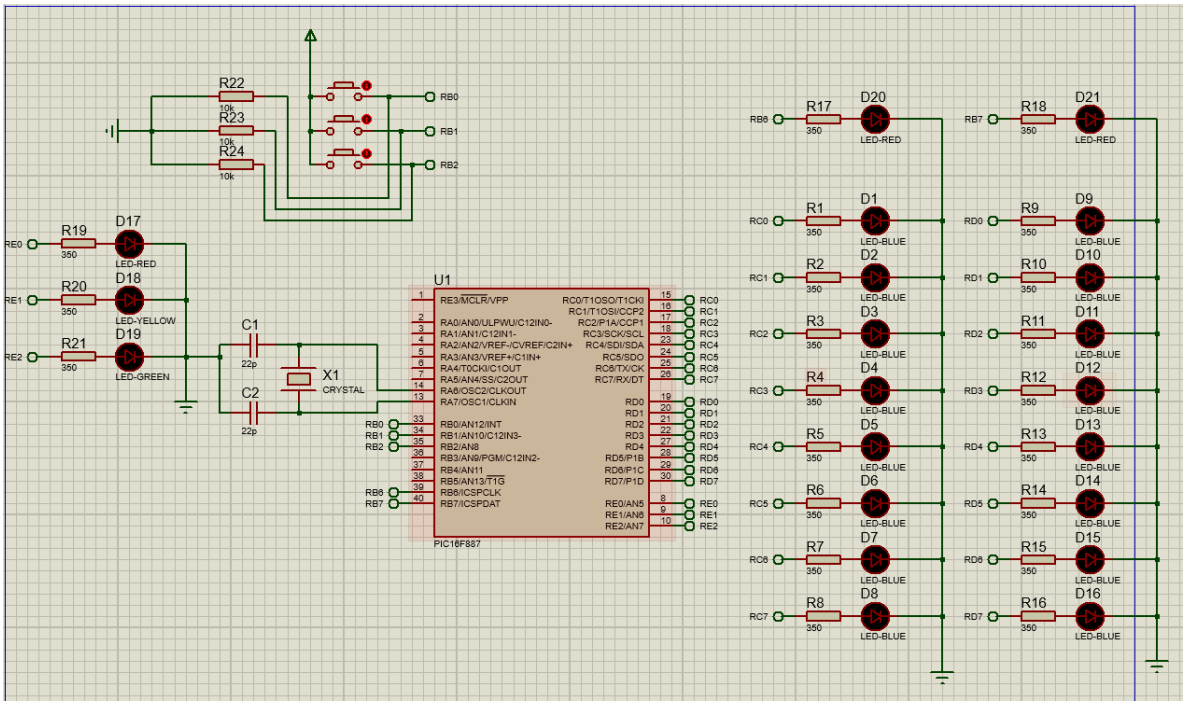
Digital 2 Sección 10

Rodrigo García 19085

Reporte Laboratorio 1

Link a Repositorio: https://github.com/gar19085/Digital2-Gar19085.git

Esquemático:

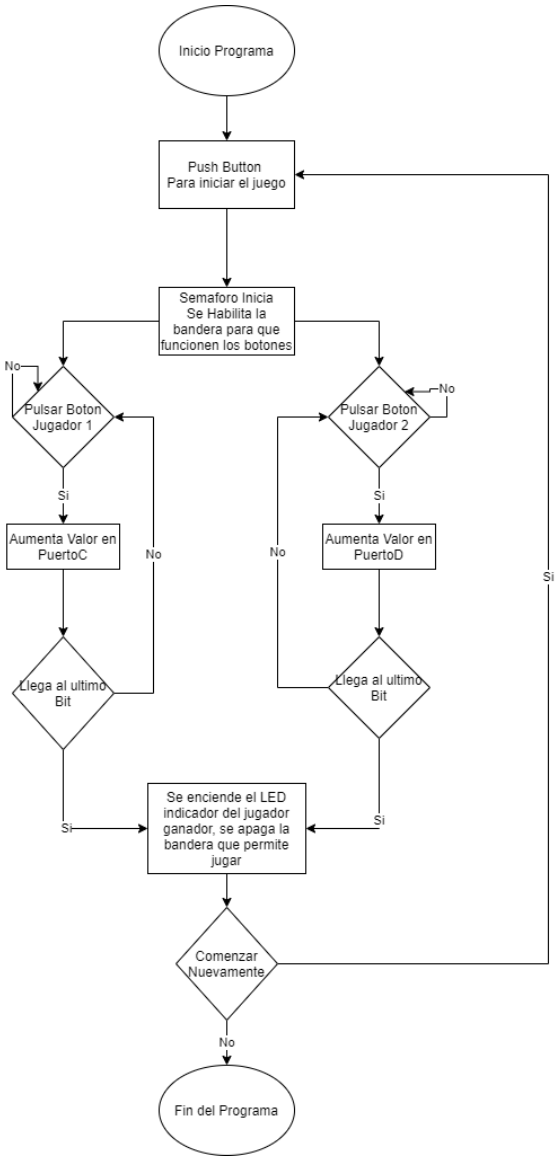


Descripción:

Configuración de Puertos:

Se Utilizaron los puertos C y D para los LEDs de cada jugador, se uso el puerto E para los LEDs del semáforo. Se utilizo el puerto B en el cual los pines RB0 a RB2 se configuraron como entradas para poder utilizar los push buttons. Los pines RB6 y RB7 se configuraron como salidas para indicar el LED que se enciende al momento que hay un ganador. En el puerto A se utilizaron los pines correspondientes para poder utilizar el cristal que funciona como reloj externo para el PIC.

Seudocódigo:



Código:

```
/*
 * File:   Lab01.c
 * Author: Rodrigo García 19085
 *
 * Created on 21 de enero de 2021
 */
```

```
// PIC16F887 Configuration Bit Settings


// 'C' source line config statements


// CONFIG1

#pragma config FOSC = EXTRC_CLKOUT// Oscillator Selection bits (RC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, RC on 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)


// #pragma config statements should precede project file includes.

// Use project enums instead of #define for ON and OFF.


#include <xc.h>


/*
```

```
 VARIABLES
 */
#define LEDR  RE0
#define LEDA  RE1
#define LEDV  RE2
#define START RB0
#define PJ1   RB1
#define PJ2   RB2
#define WIN1  RB6
#define WIN2  RB7
#define _XTAL_FREQ 8000000

//DEFINO NOMBRES ESPECIFICOS A PINES ESPECIFICOS ASÍ ES MÁS SENCILLO IDENTIFICAR LOS LEDS Y PUSH BUTTONS

char FLAG;
char DECADAJ1;
char DECADAJ2;
char J1, J2;
//CREO VARIABLES QUE LUEGO ME SERVIRAN COMO BANDERAS

/*
 FUNCIONES
 */
void Setup(void);
void Semaforo (void);
void PushSemaforo (void);
void CONTADORJ1(void);
void CONTADORJ2(void);
//GENERO MIS FUNCIONES
```

```
/*
 MAIN LOOP
 */


void main(void){
   Setup(); //LLAMO A LA FUNCIÓN SETUP
   while (1){
      if(START==1){ //REVISO SI MI PSUHBUTTON DE INICIO ESTA EN 1
         Semaforo(); //LLAMO LA FUNCIÓN DE SEMAFORO
         }


      if (FLAG == 1){ //REVISO SI MI BANDERA "FLAG" ESTA EN 1 ASI PUEDO COMENZAR EL JUEGO
         if (PJ1==1){ //REVISO SI ESTA PULSADO EL BOTON DEL JUGADOR 1
         J1 = 1;     //LE INDICO A LA BANDERA DEL JUGADOR 1 QUE ESTA SE HABILITE
      }
         else {
            if (J1==1 && PJ1==0){ //SI YA NO SE ESTQA PULSANDO EL BOTON PERO LA BANDERA DEL JUGADOR SI ESTA ENCENDIDA SE REALIZA LO SIGUIENTE
               J1=0;   //APAGO LA BANDERA DEL JUGADOR 1
               DECADAJ1++; //LE SUMO 1 A MI VARIABLE DEL CONTADOR EN DECADA
               CONTADORJ1(); //LLAMO LA FUNCIÓN CONTADOR
         }
         }
   }
      //MISMO PROCEDIMIENTO QUE PARA EL JUGADOR 1
      if (FLAG==1){
         if (PJ2==1){
         J2=1;
      }
         else {
```

```
            if (J2==1 && PJ2==0){
            J2=0;
            DECADAJ2++;
            CONTADORJ2();
          }
          }
       }
   }
}

/*
 SETUP
 */

void Setup(void){
   ANSEL = 0;
   ANSELH = 0; //INDICO EN 0 ANSEL Y ANSELH, PARA ASÍ INDICAR LOS PUERTOS COMO DIGITALES
   TRISA = 1;  //INDICO LOS PINES DEL PUERTOA COMO ENTRADAS
   TRISB = 0b00000111; //INDICO LOS PINRES RB0-RB2 COMO ENTRADAS Y LOS DEMAS COMO SALIDAS
   TRISC = 0; //INDICADOS COMO SALIDAS
   TRISD = 0;
   TRISE = 0;
   PORTA = 0; //LIMPIEZA DE PUERTOS
   PORTB = 0;
   PORTC = 0;
   PORTD = 0;
   PORTE = 0;
   OPTION_REG = 0b10000000; //APAGO PULLUPS
```

```
}


void Semaforo(void){
   PORTC=0; //LIMPIEZA DE PUERTOS NUEVAMENTE
   PORTD=0;
   WIN1=0; //LIMPIEZA DE INDICADORES DE GANADOR
   WIN2=0;
      LEDR = 1;
      LEDA = 0;
      LEDV = 0;
      __delay_ms(100);
      LEDR = 0;
      LEDA = 1;
      LEDV = 0;
      __delay_ms(100);
      LEDR = 0;
      LEDA = 0;
      LEDV = 1;
      FLAG=1; //ENCIENDO MI BANDERA PARA PERMITIR EL INICIO DEL JUEGO
      __delay_ms(100);
      LEDR = 0;
      LEDA = 0;
      LEDV = 0;
}


void CONTADORJ1(void){
   if (DECADAJ1==1){ //SI MI VARIABLE DECADAJ1 ES IGUAL A 1
      PORTC=1;      //ENCIENDO EL PRIMER BIT DEL PUERTOC
```

```
   }
      else if (DECADAJ1>1 && DECADAJ1<8){ //SI MI VARIABLE DECADAJ1 ES MAYOR A 1 Y MENOR QUE 8
         PORTC=PORTC<<1;                  //REALIZO UN SHIFT A LA IZQUIERDA EN EL PUERTOC PARA MOVER EL BIT ENCENDIDO AL SIGUIENTE
   }
      else if (DECADAJ1==8){  //SI MI VARIABLE DECADAJ1 ES IGUAL A 8
         DECADAJ1=0;          //INDICO A MI VARIABLE QUE NUEVAMENTE TENGA EL VALOR 0
         RC6=0;           //APAGO EL PIN RC6
         RC7=1;           //ENCIENDO EL PIN RC7
         WIN1=1;           //ENCIENDO MI INDICADOR DEL JUGADOR 1 QUE GANO
         FLAG=0;                  //APAGO MI BANDERA "FLAG" PARA BLOQUEAR LOS BOTONASOS DEL JUGADOR2 Y ASÍ FINALIZAR EL JUEGO HASTA QUE SE PRESIONE "START"
   }
}


//MISMO FUNCIONAMIENTO QUE EN CONTADORJ1
void CONTADORJ2(void){
   if (DECADAJ2==1){
      PORTD=1;
   }
   else if (DECADAJ2 > 1 && DECADAJ2 < 8){
      PORTD=PORTD<<1;
   }
   else if (DECADAJ2 == 8){
      DECADAJ2=0;
      RD6=0;
      RD7=1;
      WIN2=1;
      FLAG=0;
```

```
    }
}
```