Universidad del Valle de Guatemala
Electrónica Digital 2 – Sección 11
Pablo Mazariegos

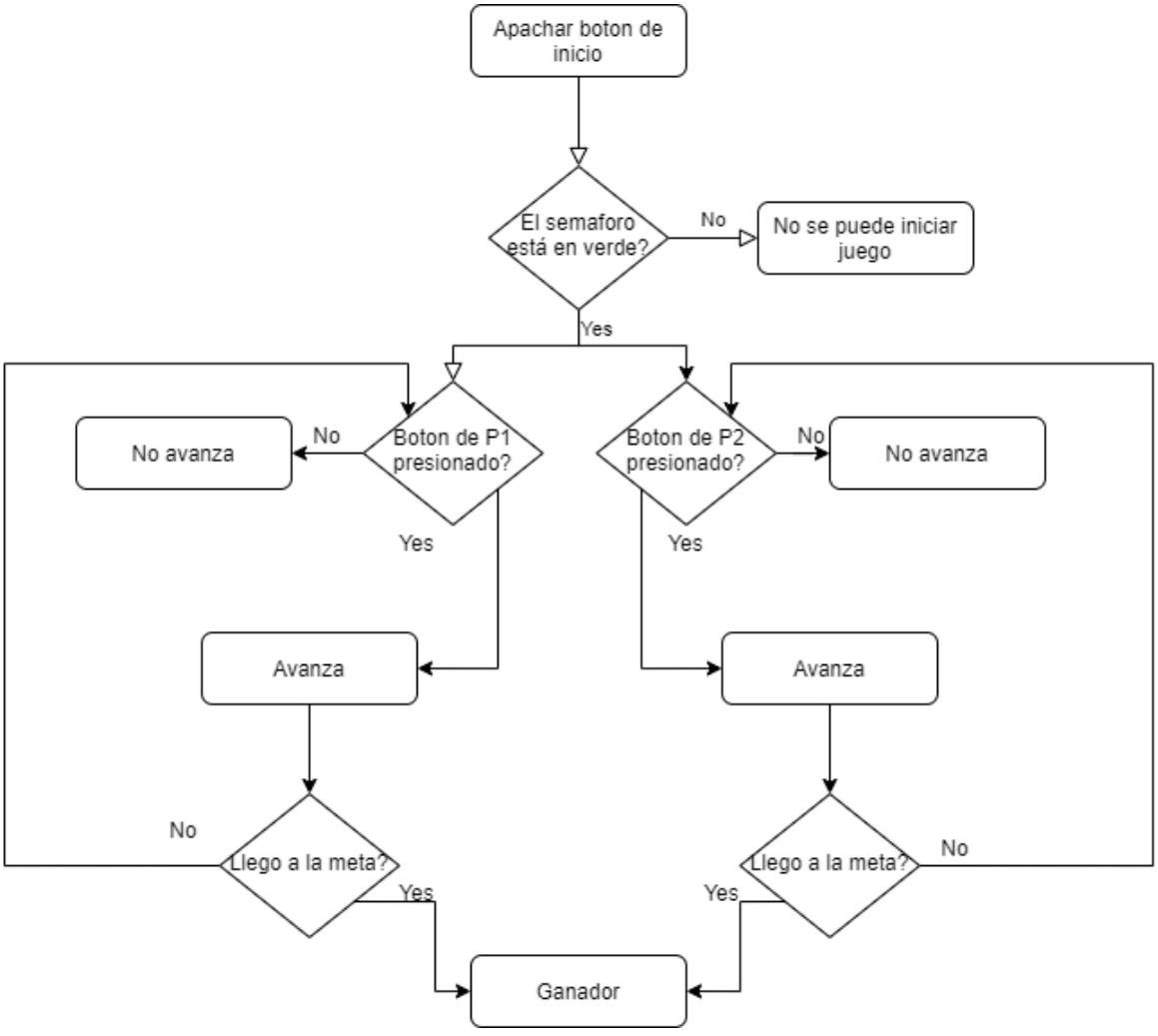
Francisco Montúfar
Carné 19379
28/01/2020

Laboratorio No.1

Juego de carreras

Link de GitHub: https://github.com/mon19379/DIGITAL2.git

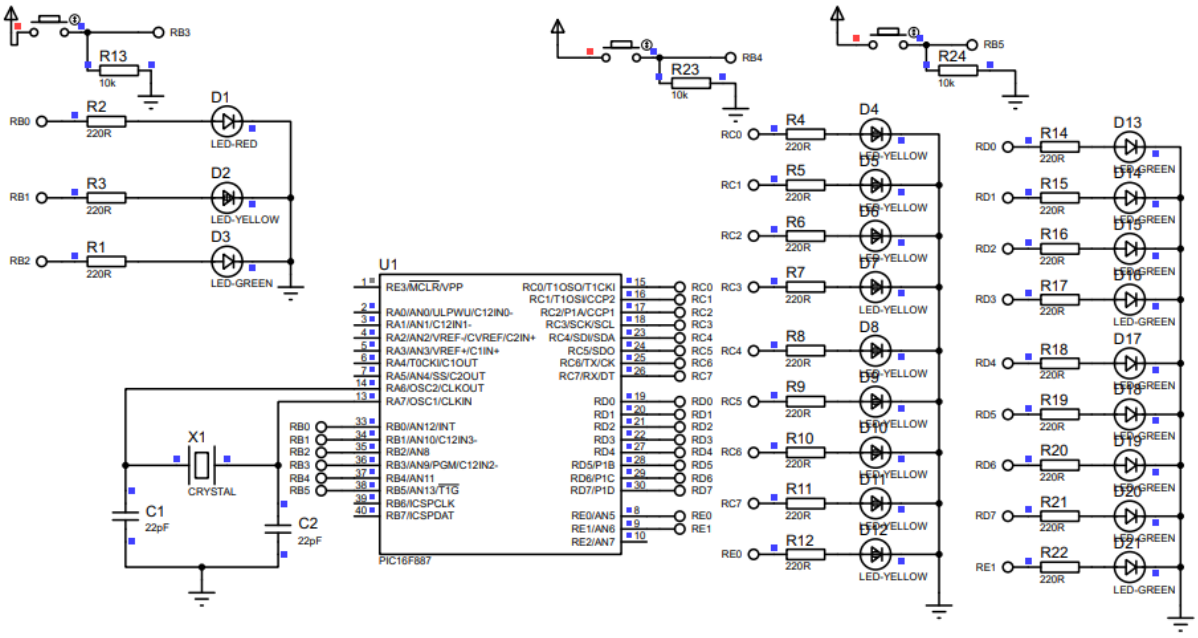
Pseudocódigo:



Descripción: El laboratorio consiste en realizar un juego de carreras para dos jugadores. El juego consiste en encender un semáforo y cuando este llega al color verde, empieza la carrera. Los jugadores

deben presionar rápidamente sus respectivos botones para poder llegar a la meta. El juego termina una vez un jugador llegue a la última LED y se encienda el indicador de victoria.

Esquemático:



Código:

```
/*FRANCISCO MONTÚFAR
 *CARNET 19379
 *ELECTRÓNICA DIGITAL 2
 *LABORATORIO #1
 *JUEGO DE CARRERAS
 *
 */
//*****************************************************************************
// Importación de librerias
//*****************************************************************************

#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)
#pragma config WRT = OFF        // Flash Program Memory Self Write Enable bits (Write protection off)
#define _XTAL_FREQ 8000000      //SE CONFIGURA EL OSCILADOR EXTERNO

//*****************************************************************************
//Variables
//*****************************************************************************
```

```
unsigned char sem = 0; //VARIABLE DE ANTIREBOTE PARA BOTON DEL SEMAFORO
unsigned char FLAG = 0; //BANDERA DEL SEMAFORO
unsigned char P1 = 0;//BANDERA CORREDOR 1
unsigned char P2 = 0;//BANDERA CORREDOR 2
unsigned char CONT = 0;//CONTADOR DEL CORREDOR 1
unsigned char CONT1 = 0;//CONTADOR DEL CORREDOR 2



//*****************************************************************************
// Prototipos de funciones
//*****************************************************************************
void Setup(void);
void semaforo(void); //FUNCIÓN DEL SEMAFORO
void player1 (void);//FUNCION DEL CORREDOR 1
void player2 (void);//FUNCION DEL CORREDOR 2



//*****************************************************************************
//Ciclo pincipal
//*****************************************************************************

void main(void) {

  Setup();
  //*****************************************************************************
  // Loop principal
  //*****************************************************************************
  while(1){
```

```
  if (PORTBbits.RB3 == 0){
     sem = 0; //SI EL BOTON ESTA EN CERO LA VARIABLE DEL SEMAFORO TAMBIEN
  }

  else{
     if (sem == 0){ //SI EL BOTON ESTA EN UNO Y LA VARIABLE DE SEMAFORO
        sem = 1;   //EN CERO, SE PONE EN UNO Y SE LLAMA LA RUTINA
        semaforo();

     }
  }

  if (FLAG == 1){ //CUANDO SE ENCIENDE LA BANDERA SE HABILITAN LOS BOTONES
     if (PORTBbits.RB4 == 1){ //ANTIREBOTE, SE PRESIONA EL BOTON
     P1 = 1; // SE ENCIENDE LA BANDERA DEL CORREDOR 1
  }

     else{
        if (P1 == 1 && PORTBbits.RB4 == 0 ){ //SE DEJA DE PRESIONAR EL
        P1 = 0;                          //BOTON
        CONT ++;// SE INCREMENTA UN CONTADOR
        player1();//SE LLAMA A LA FUNCION DEL CORREDOR1
     }

  }


}
```

```
      if (FLAG == 1){ //CUANDO SE ENCIENDE LA BANDERA SE HABILITAN LOS BOTONES
         if (PORTBbits.RB5 == 1){ //ANTIREBOTE, SE PRESIONA EL BOTON
         P2 = 1; //SE ENCIENDE LA BANDERA DEL JUGADOR2
      }

         else{
            if (P2 == 1 && PORTBbits.RB5 == 0 ){ //SE DEJA DE PRESIONAR EL
            P2 = 0;                          //BOTON
            CONT1 ++;//SE INCREMENTA UN CONTADOR
            player2();//SE LLAMA A LA FUNCION DEL CORREDOR 2
         }

      }


   }




   }
}



//*****************************************************************************
//Configuracion
//*****************************************************************************
```

```
void Setup(void) {
   ANSEL = 0;  // ENTRADAS DIGITALES
   ANSELH = 0;
   PORTA = 0; //PUERTO A EN 0
   PORTB = 0; //PUERTO B EN 0
   PORTC = 0; //PUERTO C EN 0
   PORTD = 0; //PUERTO D EN 0
   PORTE = 0; //PUERTO E EN 0
   //PINES RA0,RA1 Y RA2 COMO ENTRADAS, LOS DEMAS COMO SALIDAS
   TRISA = 0;
   TRISB = 0b00111000; //PUERTO B SALIDAS
   TRISC = 0; //PUERTO C SALIDAS
   TRISD = 0; //PUERTO D SALIDAS
   TRISE = 0; //PUERTO E SALIDAS
   OPTION_REG = 0b10000000; //SE APAGAN LAS PULLUPS DEL PUERTO B




}

//*****************************************************************************
// Subrutinas
//*****************************************************************************
void semaforo (void){
   if (sem == 1){ //SI LA VARIABLE DE SEMAFORO = 1 SE CORRE LA FUNCION
      PORTD = 0; //SE REINICIA CORREDOR 2
      PORTC = 0; //SE REINICIA CORREDOR 1
```

```
        PORTE = 0; //SE REINICIA INDICADOR DE VICTORIA
        PORTBbits.RB0 = 1; //SE ENCIENDE LA LED ROJA DEL SEMAFORO
        PORTBbits.RB1 = 0; //LAS OTRAS DOS SE MANTIENEN APAGADAS
        PORTBbits.RB2 = 0;
        __delay_ms(50);  //DELAY DE 500 ms

        PORTBbits.RB0 = 0;
        PORTBbits.RB1 = 1; //SE ENCIENDE LA LED AMARILLA DEL SEMAFORO
        PORTBbits.RB2 = 0; //LAS OTRAS DOS SE MANTIENEN APAGADAS
        __delay_ms(50);

        PORTBbits.RB0 = 0;
        PORTBbits.RB1 = 0;//LAS OTRAS DOS SE MANTIENEN APAGADAS
        PORTBbits.RB2 = 1;//SE ENCIENDE LA LED VERDE DEL SEMAFORO
        FLAG = 1;

        __delay_ms(100);

        PORTBbits.RB0 = 0;
        PORTBbits.RB1 = 0;
        PORTBbits.RB2 = 0; //SE APAGAN TODAS LAS LEDS DEL SEMAFORO
            //SE ENCIENDE LA BANDERA PARA QUE SE HABILITEN LOS
    }               //CORREDORES


}

void player1(void){
    if(CONT == 1){
```

```
    PORTC = 1; //SI EL CONTADOR ESTA EN 1, SE ENCIENDE LA PRIMER LED
  }
    else if (CONT >1 && CONT < 8){ //SI EL CONTADOR SE ENCUENTRA ENTRE 1 Y 8
      PORTC = PORTC<<1;         //SE HACE UN SHIFT AL UNICO BIT ACTIVO
  }

  else if(CONT == 8){
    CONT = 0;   //SI EL CONTADOR ES = 8 SE RESETEA
    PORTCbits.RC6 = 0; //SE APAGA LA PENULTIMA LED
    PORTCbits.RC7 = 1;//SE ENCIENDE LA ULTIMA LED
    PORTEbits.RE0 = 1;//SE ACTIVA EL INDICADOR DE VICTORIA
    FLAG = 0; //SE APAGA LA BANDERA QUE HABILITA LOS BOTONES
  }
}

void player2 (void){
  if (CONT1 == 1){
    PORTD = 1; //SI EL CONTADOR ESTA EN 1, SE ENCIENDE LA PRIMER LED
  }

    else if (CONT1 > 1 && CONT1 < 8){ //SI EL CONTADOR SE ENCUENTRA ENTRE 1 Y 8
      PORTD = PORTD<<1;           //SE HACE UN SHIFT AL UNICO BIT ACTIVO
    }

  else if(CONT1 == 8){
    CONT1 = 0;     //SI EL CONTADOR ES = 8 SE RESETEA
    PORTDbits.RD6 = 0; //SE APAGA LA PENULTIMA LED
    PORTDbits.RD7 = 1; //SE ENCIENDE LA ULTIMA LED
    PORTEbits.RE1 = 1; //SE ACTIVA EL INDICADOR DE VICTORIA
```

```
        FLAG = 0; //SE APAGA LA BANDERA QUE HABILITA LOS BOTONES
    }
}
```