Universidad del Valle de Guatemala
Electronica Digital 2

José Ignacio Ramírez Soto
Carné: 17787

Laboratorio 1: Juego de Carrera

Pseudocódigo

Configuración

```
ANSELH = 0
PORTB = 0
TRISB = 0b00000111
WPUB = 0b00000111
IOCB = 0b00000111
TRISA = 0
ANSEL = 0
PORTA = 0
TRISC = 0
PORTC = 0
TRISD = 0
PORTD = 0
```

Principal

```
Tabla_Display(3)
mientras(1)
while (inicio = 0)
si(boton = 1){
inicio = 1
si(inicio= 1){
Esperar 0.75 seg
Encender LED Rojo
```

```
            Tabla_Display(2)
            Esperar 0.75 seg
            Encender LED Amarrillo
            Apagar LED Rojo
            Tabla_Display(1)
            Esperar 0.75 seg
            Encender LED Verde
            Apagar LED Amarillo
            Tabla_Display(0)
            Inicio = 2
Si inicio = 2
        SI(boton2 y btn1 = 0)
            btn1 = 1
        Si (Contadorbtn1 <= 100)
            counterbtn1 = counterbtn1 + btn1
         Si(Contadorbtn1 > 100)
           Si(boton2 = 1 y btn1 = 1 y counted1 = 0)
            Si(PORTA = 0)
               PORTA++
            DeLoContrario
               PORTA = PORTA*2
            Si(PORTA = 0b10000000)
             LedGanador1 = 1
             Tabla_Display(1)
             Inicio = 0
            counted1 = 1
           DeLoContrario Si(boton2 = 0)
            counterbtn1 = 0
            btn1 = 0
```

```
        counted1 = 0
    SI(boton3 y btn2 = 0)
        Btn2 = 1
    Si (Contadorbtn2 <= 100)
        Counterbtn2 = counterbtn2 + btn2
     Si(Contadorbtn2 > 100)
       Si(boton3 = 1 y btn2 = 1 y counted2 = 0)
        Si(PORTD = 0)
           PORTD++
        DeLoContrario
           PORTD = PORTD*2
        Si(PORTD = 0b10000000)
         LedGanador2 = 1
         Tabla_Display(2)
         Inicio = 0
        Counted2 = 1
       DeLoContrario Si(boton3 = 0)
        Counterbtn2 = 0
        Btn2 = 0
        Counted2 = 0


Tabla_Display
        Si 0
                PORTC = 0b10001000;
        Si 1
                PORTC = 0b10111110;
        Si 2
                PORTC = 0b11000100;
```

Si 3

PORTC = 0b10010100;

Código

```
/*
 * File:   main.c
 * Author: JIRS0129
 *
 * Created on January 24, 2020, 12:44 PM
 */


#include <xc.h>
#define _XTAL_FREQ 4000000

//Global variables
unsigned int start = 0;
unsigned int btn1 = 0;
unsigned int btn2 = 0;
unsigned int counterbtn1 = 0;
unsigned int counterbtn2 = 0;
unsigned int counted1 = 0;
unsigned int counted2 = 0;

//Function prototypes
void setup (void);
void reset (void);
void disp_table(unsigned int number);
unsigned int playercounter(unsigned int prior);

void main(void) {
```

```
setup();
disp_table(3);  //Setup and place the 3 on display

while(1){
  while (start == 0) {     //infinite loop until start button is pressed.
    if(PORTBbits.RB0 == 1){
      start = 1;
      reset();            //Reset ports in case not first time running.
    }
  }
  if(start == 1){    //If start button pressed
    //Race countdown routine
    __delay_ms(750);
    PORTBbits.RB7 = 0;
    disp_table(2);
    __delay_ms(750);
    PORTBbits.RB7 = 1;
    PORTBbits.RB6 = 0;
    disp_table(1);
    __delay_ms(750);
    PORTBbits.RB6  = 1;
    PORTBbits.RB5 = 0;
    disp_table(0);
    __delay_ms(5);
    start = 2;  //Allow players to press buttons
  }
  else {
     if(PORTBbits.RB1 && btn1 == 0){ //If button is pressed and hasn't been pressed before
        btn1 = 1;
     }

     if (counterbtn1 <= 100) {       //count 100 times for the anti-bounce
       counterbtn1 = counterbtn1 + btn1;
     }
```

```
else {                      //when done counting

 if(counterbtn1 > 100){
  if(PORTBbits.RB1 == 1 && btn1 == 1 && counted1 == 0){   //check if button is still pressed.

    if(PORTA == 0){      //if so, increase PORTA first time
       PORTA++;
    }
    else{              //or multiply by 2 after the first time
       PORTA = PORTA*2;
    }
    if(PORTA == 0b10000000){     //if J1 reached end
     PORTBbits.RB3 = 1;         //Activate led
     disp_table(1);             //Show 1 in 7 seg
     start = 0;                 //reset in case of new game
    }
    counted1 = 1;               //variable for code to only run once.
  }
  else if(PORTBbits.RB1 == 0){                // If it was released befor the 100'th count or after the increment in PORTA
    counterbtn1 = 0;            //reset variables for next push
    btn1 = 0;
    counted1 = 0;
  }
 }
}

if(PORTBbits.RB2 && btn2 == 0){ //If button is pressed and hasn't been pressed before
   btn2 = 1;
}
if (counterbtn2 <= 100) {      //count 100 times for the anti-bounce
 counterbtn2 = counterbtn2 + btn2;
}else{                      //when done counting

 if(counterbtn2 > 100){
```

```
        if(PORTBbits.RB2 == 1 && btn2 == 1 && counted2 == 0){   //check if button is still pressed.


          if(PORTD == 0){      //if so, increase PORTAD first time
             PORTD++;
          }else{               //or multiply by 2 after the first time
             PORTD = PORTD*2;
          }
          if(PORTD == 0b10000000){     //if J2 reached end
            PORTBbits.RB4 = 1;         //Activate led
            disp_table(2);             //Show 2 in 7 seg
            start = 0;                 //reset in case of new game
          }
          counted2 = 1;                //variable for code to only run once.
        }else if(PORTBbits.RB2 == 0){                // If it was released befor the 100'th count or after the increment in PORTA
          counterbtn2 = 0;             //reset variables for next push
          btn2 = 0;
          counted2 = 0;
         }
       }
      }


   }
 }
}


void reset(void){   //Clears all ports and resets game
   PORTA = 0;
   PORTD = 0;
   disp_table(3);
   PORTBbits.RB7 = 0;
   PORTBbits.RB6 = 1;
   PORTBbits.RB5 = 1;
   PORTBbits.RB4 = 0;
   PORTBbits.RB3 = 0;
}
```

```
void disp_table(unsigned int number){ //Table to display numbers on 7 segment


   //dot C D E G F A B
 PORTD = number*0;
 switch (number) {
   case 0b00000000:
     PORTC = 0b10001000;
     break;
   case 0b00000001:
     PORTC = 0b10111110;
     break;
   case 0b00000010:
     PORTC = 0b11000100;
     break;
   case 0b00000011:
     PORTC = 0b10010100;
     break;
 }
 return;
}


void setup(void){
   // PIC16F887 Configuration Bit Settings


   // 'C' source line config statements


   // CONFIG1
   #pragma config FOSC = INTRC_NOCLKOUT// Oscillator Selection bits (INTOSC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, I/O
function 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 enabled)
```

```
#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.

//PORTB interrupt on change enable.
INTCONbits.GIE = 1;
INTCONbits.RBIE = 1;
INTCONbits.INTE = 1;
INTCONbits.INTF = 0;
INTCONbits.RBIF = 0;
//PORTB (0:2 buttons, 3:5 RGY, 6:7 winners' LEDs) initialization
ANSELH = 0;
PORTB = 0;
TRISB = 0b00000111;
WPUB = 0b00000111;
IOCB = 0b00000111;

//PORTA (J1) initialization
TRISA = 0;
ANSEL = 0;
PORTA = 0;

//PORTC (display) initialization
TRISC = 0;
PORTC = 0;

//PORTD (J2) initialization
TRISD = 0;
```

```
    PORTD = 0;


    return;


}//*/
```

Link a GitHub

https://github.com/JIRS0129/Digital2