UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

ICO9

SISTEMAS DIGITALES

PING_PONG_GAME

EJERCICIO PRÁCTICO DE PROGRAMACIÓN DEL PIC16F84A

EN ENSAMBLADOR

RAMIREZ MACEDA SINHUE 1872803

GRANADOS AYALA ISAAC ADYAEL 1824741

RODOLFO ZOLA GARCIA LOZANO

17/11/2022

Juego Ping Pong (código)

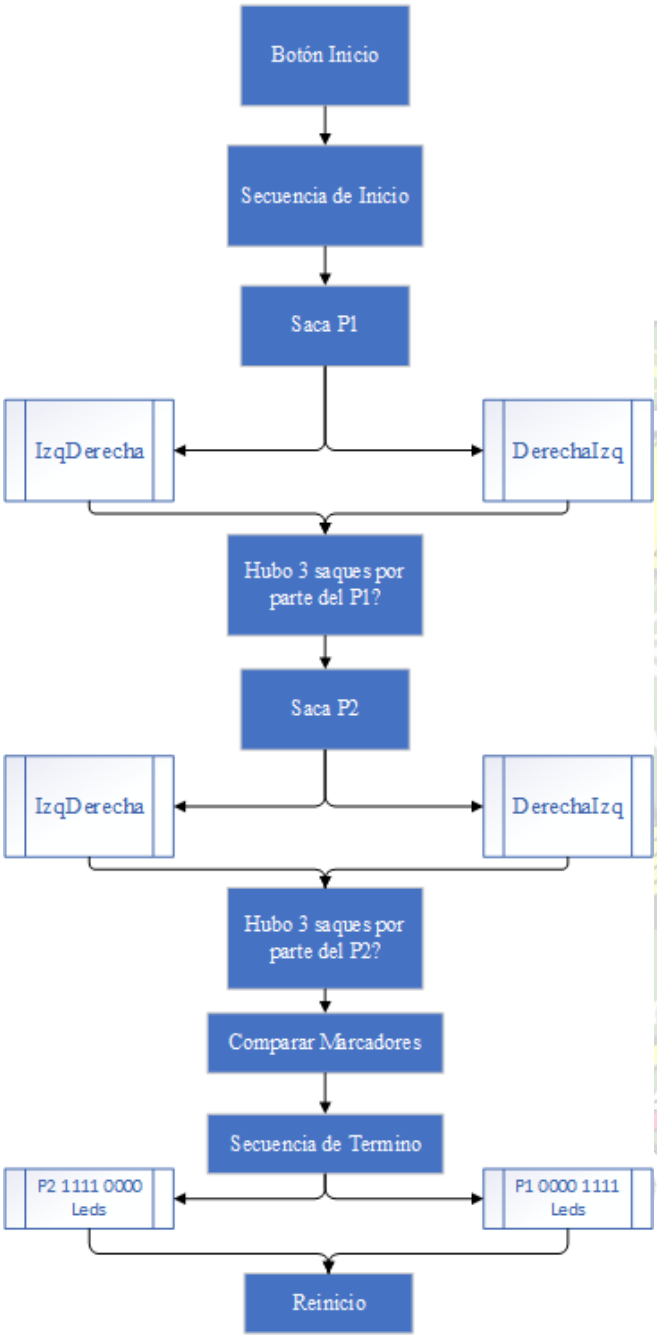
Se desea implementar un juego de ping-pong. El juego debe de tener las siguientes características:

1. El movimiento de la pelota se emulará con el encendido de ocho leds, los cuales irán prendiendo de forma secuencial, de un lado al otro, simulando el movimiento de la pelota. En cada lado de la tira de LEDs se conectará un push-button que servirá de “raqueta” para “golpear” la pelota.
2. El juego inicia cuando se presiona el botón de Start. Cuando se presiona el botón encenderá un LED del lado del Player 1. El juego iniciará en el momento que el Player 1 presione su botón.
3. Si el usuario no “golpea” la pelota dentro del intervalo de 200 ms, el juego considerará que perdió el turno y le dará un punto al otro jugador.
4. Cada que un usuario golpee la pelota se escuchará un bip.
5. El juego dará tres saques consecutivos al Player 1 y tres saques al Player 2. El que gane más saques será el ganador.
6. Cuando termina el juego, se encenderá la mitad de los LEDs que estén del lado del ganador y ahí se quedará el programa hasta que se presione la tecla start.

La conexión del circuito debe cumplir con las siguientes condiciones:

- Los LEDs deben de estar conectados al puerto B.
- El jugador 1 estará del lado del bit menos significativo.
- El push-button del Player1 estará conectado a RA0.
- El jugador 2 estará del lado del bit más significativo.
- El push-button del Player2 estará conectado a RA4.
- El botón Start estará conectado a RA2
- El zumbador estará conectado a la terminal RA1.

Diagrama de bloques



Botón de Inicio: Para el inicio del programa, los ocho leds estarán encendidos hasta que se presione el botón START para así pasar a la secuencia de inicio.

Secuencia de Inicio: La animación empezará encendiendo leds al modo de semáforo con una secuencia de sonido que al acabar dará paso al saque del P1.

Saque P1: En el ping pong el jugador uno ubicado a la derecha de la mesa, siempre saca por lo que la animación será un led de su lado encendido, que avanzara hasta que el push botón envié una señal activando el recorrido de derecha a izq.

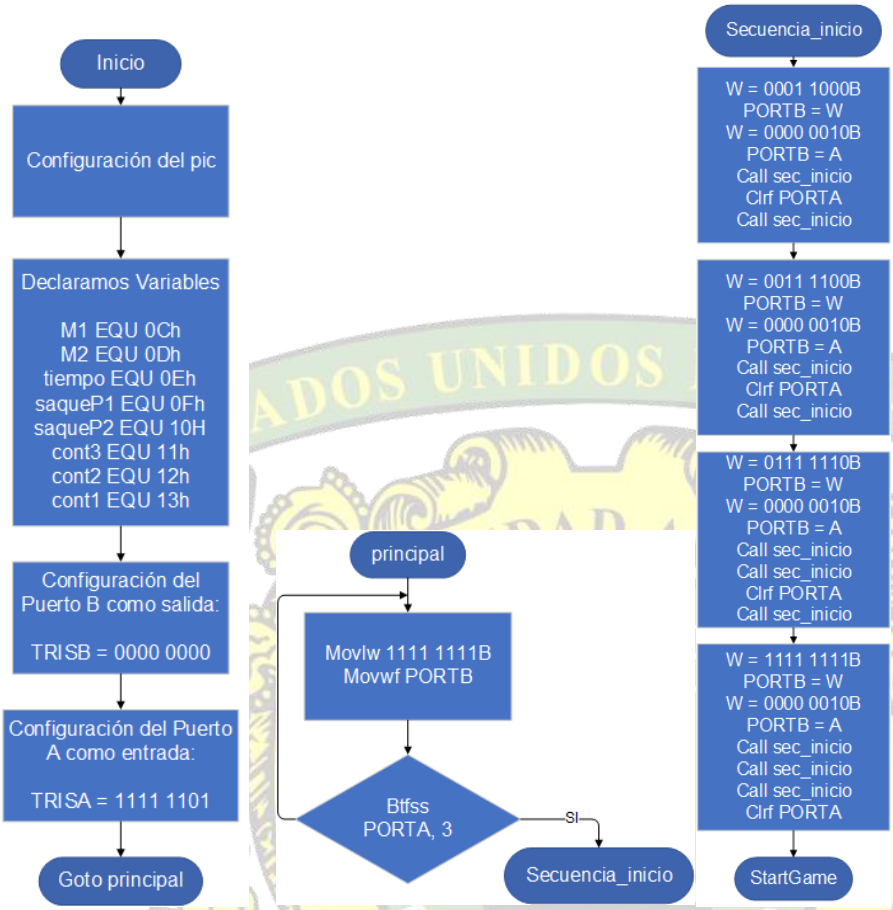
IzqDerecha/DerechaIzq: Esta función cuneta con la animación cuando cualquier jugador le atina a la pelota haciendo que viaje el led de un lado a otro hasta que alguno de los 2 no le dé.

Saques: Se checa cuantos saque lleva el jugador uno para así darle chance al otro jugador de realizar sus 3 saques.

Marcadores: en este punto ya los respectivos marcadores de los jugadores han sido incrementados con respecto a las rondas que ganaron, por lo que comparan dichos marcadores con la ayuda del código **SUBWF**, este para restar los valores y así poder identificar si el ganador es el P1, P2 o se queda en empate.

Secuencia de Termino: Este apartado va de la mano con la función anterior que cuando hay empate, se encienden los leds de la siguiente manera 0011 1100, para indicar que el ganador fue el P2 los leds se presentan así 1111 0000 y para el P1 de la siguiente manera 0000 1111.

Diagramas de Flujo



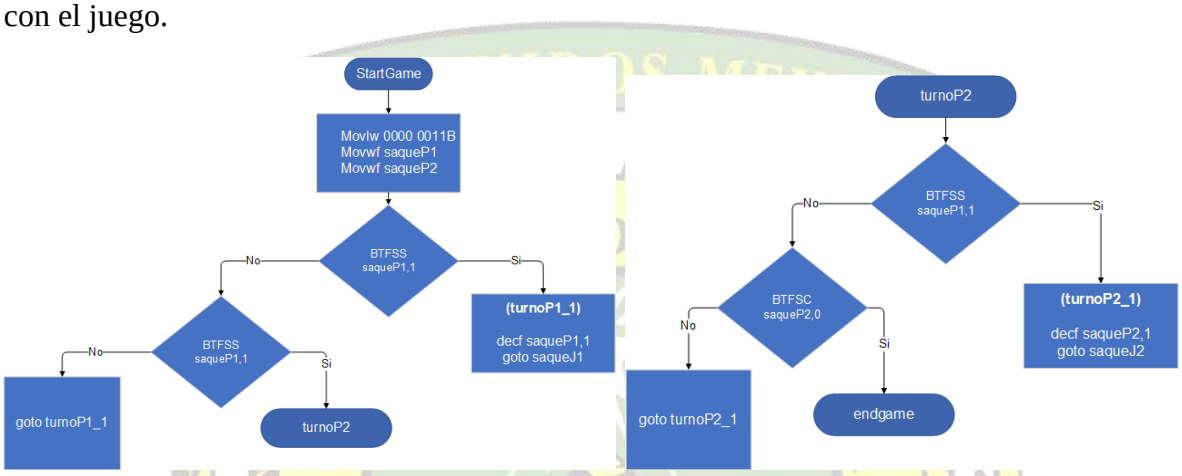
Empezamos analizando los primeros 3 diagramas de flujo, el de la izquierda nos describe el orden en el que configuraremos el PIC, agregando un total de 8 variables, por consiguiente, la configuración del Puerto B como salida colocando TRISB = 0000 0000. Después en la parte del puerto A se configura como entrada de la siguiente manera TRISA = 1111 1101, como podemos ver la posición 1 (RA1) lo dejamos en 0, esto para declarar que lo usaremos como salida y ahí mismo conectaremos el buzzer para las simulaciones de sonido.

Variables	Descripción
M1	Marcador del jugador 1
M2	Marcador del jugador 2
Tiempo	Guarda los tics del programa
saqueP1	Contiene el número de turnos que debe sacar el jugador 1.
saqueP2	Contiene el número de turnos que debe sacar el jugador 2.
Cont3	Contiene el valor para las secuencias de tiempo y así poder llevar a cabo los 200 ms
Cont2	Contiene el valor para las secuencias de tiempo y así poder llevar a cabo los 200 ms

Cont1	Contiene el valor para las secuencias de tiempo y así poder llevar a cabo los 200 ms

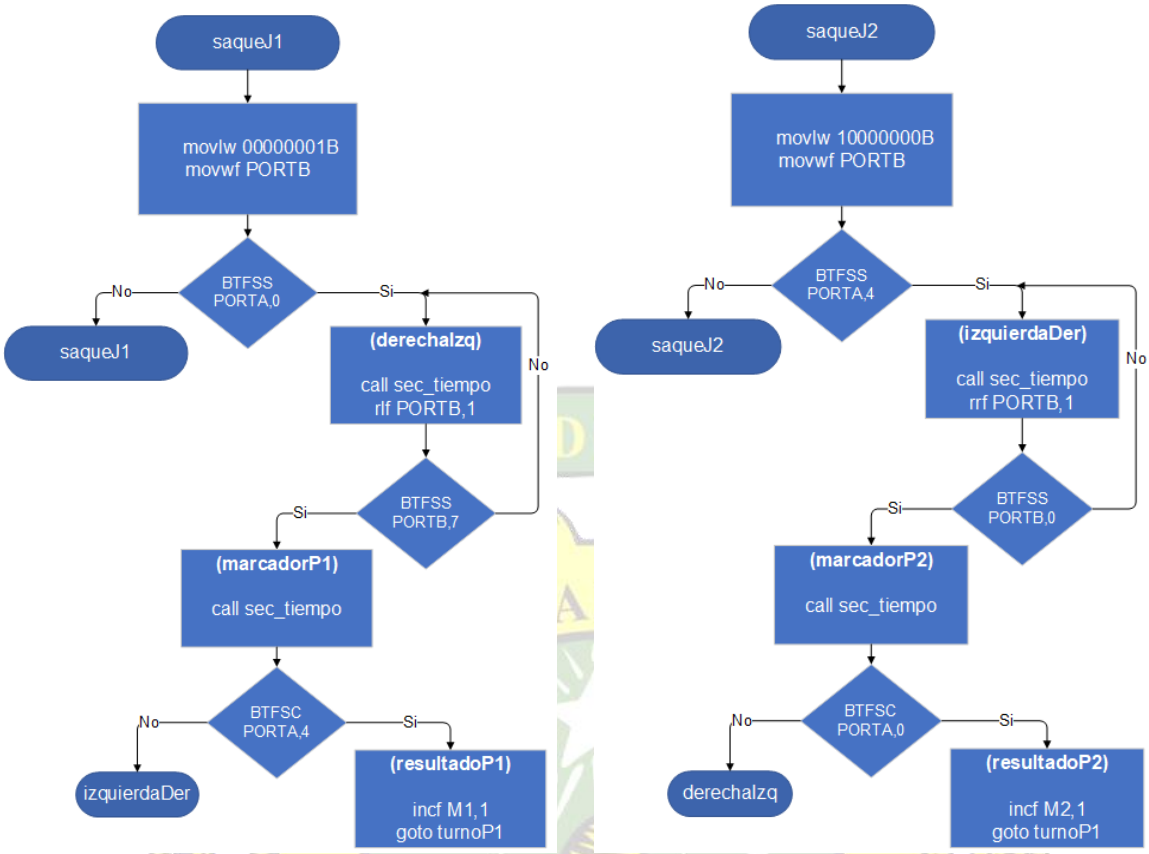
Continuando con el diagrama de en medio, estará enviando información constante al puerto B, esto simulando el encendido de los 8 leds que seguirán así hasta que el jugador presione el botón de START.

El diagrama de secuencia de Inicio lo que hace es una simulación del encendido de un semáforo de carreras, encendiendo de 2 en dos cada cierto tiempo para así después iniciar con el juego.

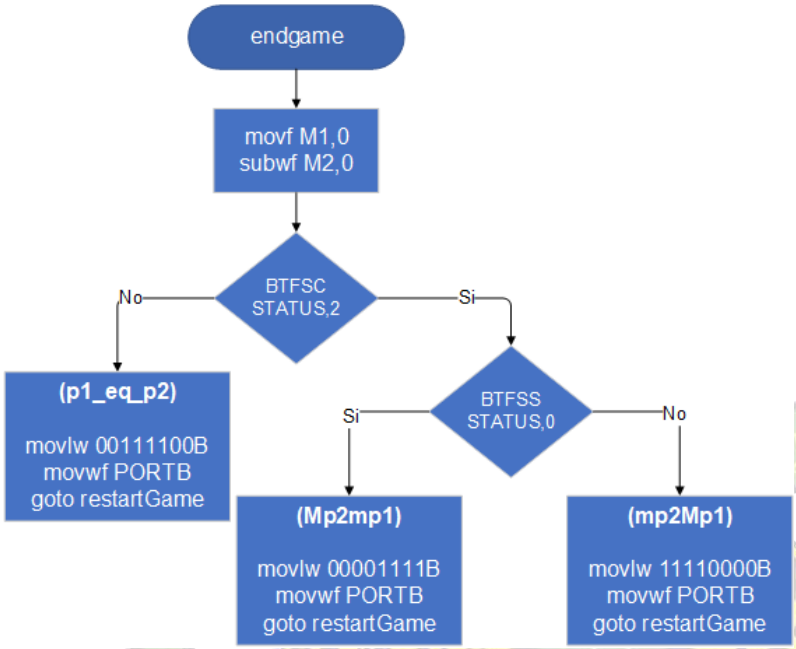


Continuando con la estructura, el siguiente diagrama de flujo sirve para añadir el numero de saques correspondientes para cada jugador que serán 3. Una vez agregado los valores a las respectivas variables, analizaremos bit por bit buscando si siguen encendido los valores, esto porque cuando se inicia con un 3, en binario se representa con 0011 entonces cuando le quitamos un turno a la variable de saqueJ1 este pasa a ser 2 pero de la siguiente manera 0010, entonces sigue sacando el jugador 1, después cuando se vuelva a restar, 0001 solo quedara encendido un bit por lo que saltara de línea y ahora perguntyara si el bit que queda sigue encendido por lo que cuando llegue a 0, el programa sabrá que el J1 ya agoto sus 3 saques y le dará chance al jugador 2.

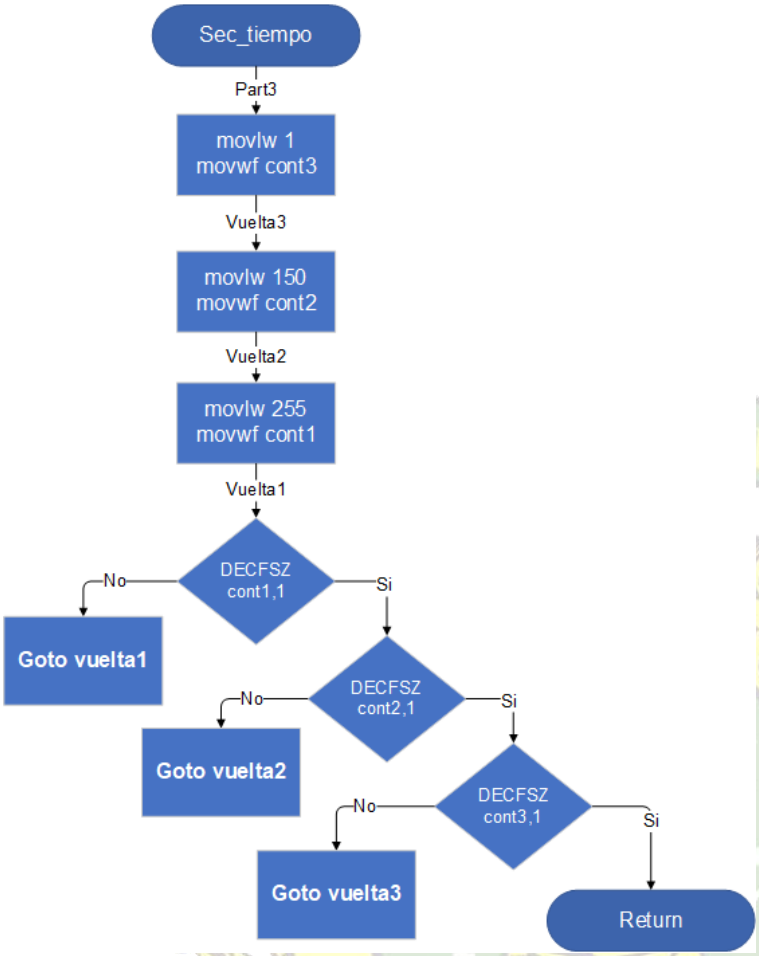
El diagrama es un paralelo al del jugador 1 solo que ahora se concentrara con las variables del J2 una vez haya acabado los 3 turnos del J1.



Llegamos al punto en el que cuando ya va a sacar un jugador, sea el 1 o el 2, se encenderá en el caso del P1 el RB0 y en el caso del P2 el RB7, esto para simular que tiene la pelota en su turno dicho jugador, que cuando el programa detecte que el jugador con la pelota la haya golpeado, este se ira al apartado de mover DERIZQ O IZQDER para dar una animación que la pelota esta viajando de un lado a otro. Hasta el final de los dos diagramas vemos que tenemos una condición, esta es que cuando la pelota llegue al otro lado, y el jugador que iba a recibir el disparo no logra conectar con la pelota, se le incrementara un punto al marcador del jugador que si conecto con la pelota, en el caso que estén conecte y conecte la pelota ambos jugadores, el led (pelota) estará recorriendo de un lado a otro esperando a que alguno de los 2 jugadores falle.



Por último, llegamos al ENDGAME, para llegar a este apartado, el programa ya debió de haber pasado por los 3 saques respectivos de cada jugador. Entonces moveremos uno de los marcadores a W y después con la ayuda del subwf, restaremos el marcador que mandamos a W menos el otro Marcador, por lo que preguntamos si la posición 2 del STATUS (Z) se enciende, significaría que el resultado de la operación dio 0, por lo que es un empate en toda regla, si no se prendió, preguntamos si el Carry en la posición 0 esta encendido, si lo esta significa que el P1 gano ósea que salió positiva la operación, si esta apagada significa que la operación salió negativa dándole la victoria al P2.



Esta subrutina es simple y sencillamente para que el programa se tome pautas al momento de estar ejecutando línea por línea, esta subrutina se ocupa principalmente para el tiempo que tarda la pelota en viajar de extremo a extremo, como también para la secuencia de inicio y que el tiempo en que se tarda en sonar el buzzer y encender los leds sean lentos para así dar una animación de semáforo de carreras.

Código del programa.

```
#include <xc.inc>

; ZONA DE CONFIGURACION GENERAL
    CONFIG FOSC = XT
    CONFIG WDTE = OFF
    CONFIG PWRTE = OFF
    CONFIG CP = OFF

; DEFINICION DE LA SECCION DEL CODIGO
    PSECT Programa, class=CODE,delta=2,abs
    ORG 00h
    M1 EQU 0Ch
    M2 EQU 0Dh
    tiempo EQU 0Eh
    saqueP1 EQU 0Fh
    saqueP2 EQU 10H
    cont3 EQU 11h
    cont2 EQU 12h
    cont1 EQU 13h

    Programa:
        GOTO INICIO

; ZONA DE CONFIGURACION DE LOS PUERTOS
        INICIO:
;CONFIGURAMOS TODO EL PUERTO B COMO SALIDA
            bsf STATUS,5
            clrf TRISB
;CONFIGURAMOS EL PUERTO A COMO ENTRADA
            MOVLW 11111001B
            MOVWF TRISA
            bcf STATUS,5

            principal:
                movlw 11111111B
                movwf PORTB
                btfss PORTA,3
                goto principal
                goto Secuencia_inicio


            Secuencia_inicio:
                clrf PORTB
        ;Primer encendido
                movlw 00011000B
                movwf PORTB
                movlw 00000010B
                movwf PORTA
                call sec_inicio
                clrf PORTA
                call sec_inicio

        ;Segundo encendido
                movlw 00111100B
                movwf PORTB
                movlw 00000010B
                movwf PORTA
                call sec_inicio
                clrf PORTA
                call sec_inicio
```

```
;Tercer encendido
        movlw 01111110B
        movwf PORTB
        movlw 00000010B
        movwf PORTA
        call sec_inicio
        call sec_inicio
        clrf PORTA
        call sec_inicio

;Cuarto encendido
        movlw 11111111B
        movwf PORTB
        movlw 00000010B
        movwf PORTA
        call sec_inicio
        call sec_inicio
        call sec_inicio
        clrf PORTA


    startGame:
        clrf PORTB
        clrf M1
        clrf M2
        movlw 00000011B
        movwf saqueP1
        movwf saqueP2
```

```
    turnoP1:
        BTFSS saqueP1,1
        goto turnoValP1
        goto turnoP1_1
        turnoP1_1:
            decf saqueP1,1
            ;BTFSC saqueP1,0
            goto saqueJ1
            ;goto turnoP2
        turnoValP1:
            BTFSC saqueP1,0
            goto turnoP1_1
            goto turnoP2

    turnoP2:
        BTFSS saqueP2,1
        goto turnoValP2
        goto turnoP2_1
        turnoP2_1:
            decf saqueP2,1
            goto saqueJ2
            ;goto endGame
        turnoValP2:
            BTFSC saqueP2,0
            goto turnoP2_1
            goto endGame
saqueJ1:
    movlw 00000001B
    movwf PORTB
    btfss PORTA,0
    goto saqueJ1
```

```
saqueJ2:
    movlw 10000000B
    movwf PORTB
    btfss PORTA,4
    goto saqueJ2
    goto Sonido2

Sonido1:
    movlw 00000010B
    movwf PORTA
    call sec_inicio
    clrf PORTA
    goto derechaIzq

Sonido2:
    movlw 00000010B
    movwf PORTA
    call sec_inicio
    clrf PORTA
    goto izquierdaDer

derechaIzq:
    call sec_tiempo
    rlf PORTB,1
    btfss PORTB,7
    goto derechaIzq
    goto marcadorP1
```

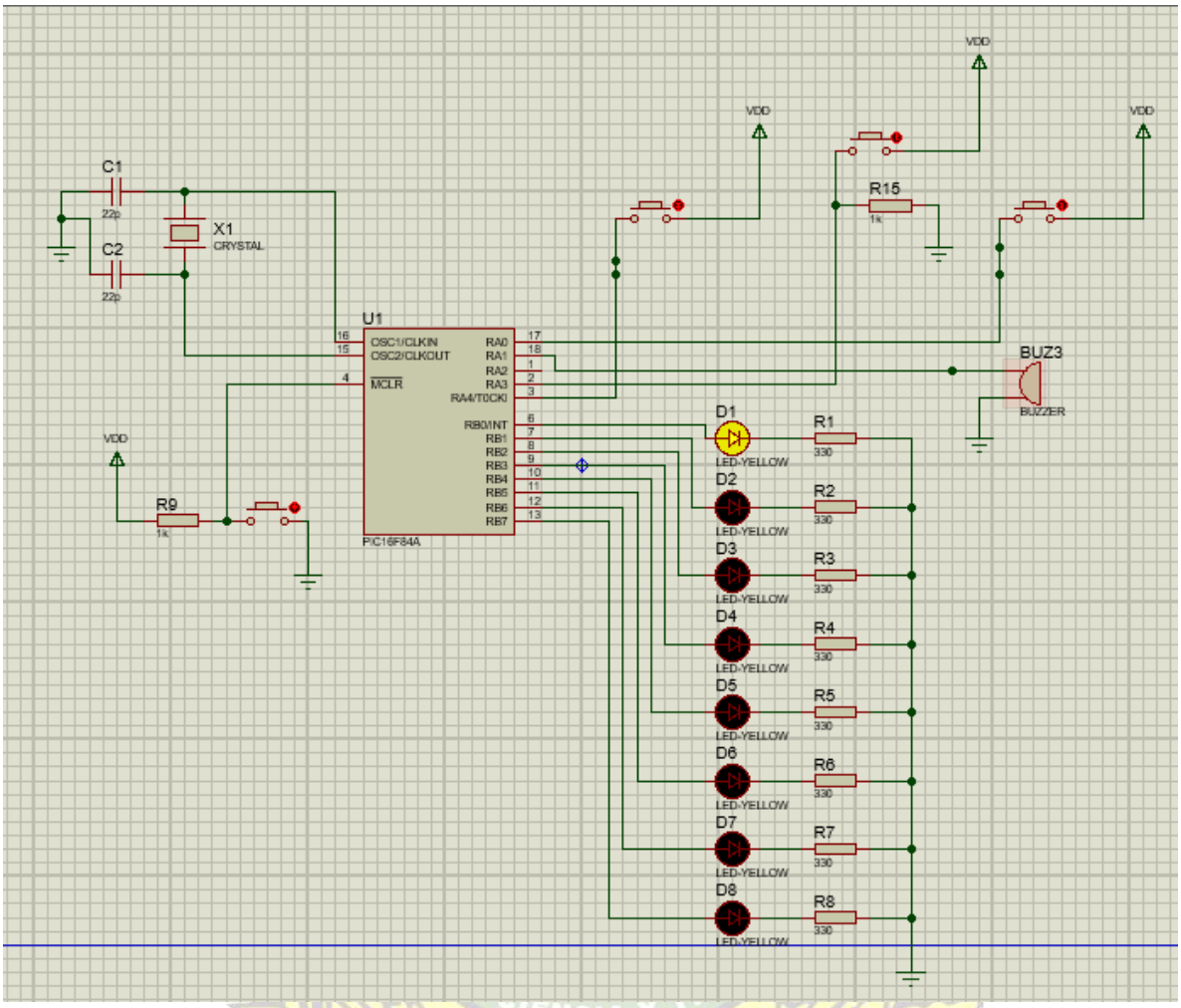
```
izquierdaDer:
    call sec_tiempo
    rrf PORTB,1
    btfss PORTB,0
    goto izquierdaDer
    goto marcadorP2

marcadorP1:
    call sec_tiempo
    BTFSC PORTA,4
    goto Sonido2
    goto resultadoP1

resultadoP1:
    incf M1,1
    goto turnoP1

marcadorP2:
    call sec_tiempo
    BTFSC PORTA,0
    goto Sonido1
    goto resultadoP2

resultadoP2:
    incf M2,1
    goto turnoP1
```

```
endGame:
    movf M1,0
    subwf M2,0
    btfsc STATUS,2
    goto p1_eq_p2
    btfss STATUS,0
    goto Mp2mp1
    goto mp2Mp1
p1_eq_p2:
    movlw 00111100B
    movwf PORTB
    goto restartGame
Mp2mp1:
    movlw 00001111B
    movwf PORTB
    goto restartGame
mp2Mp1:
    movlw 11110000B
    movwf PORTB
    goto restartGame
```

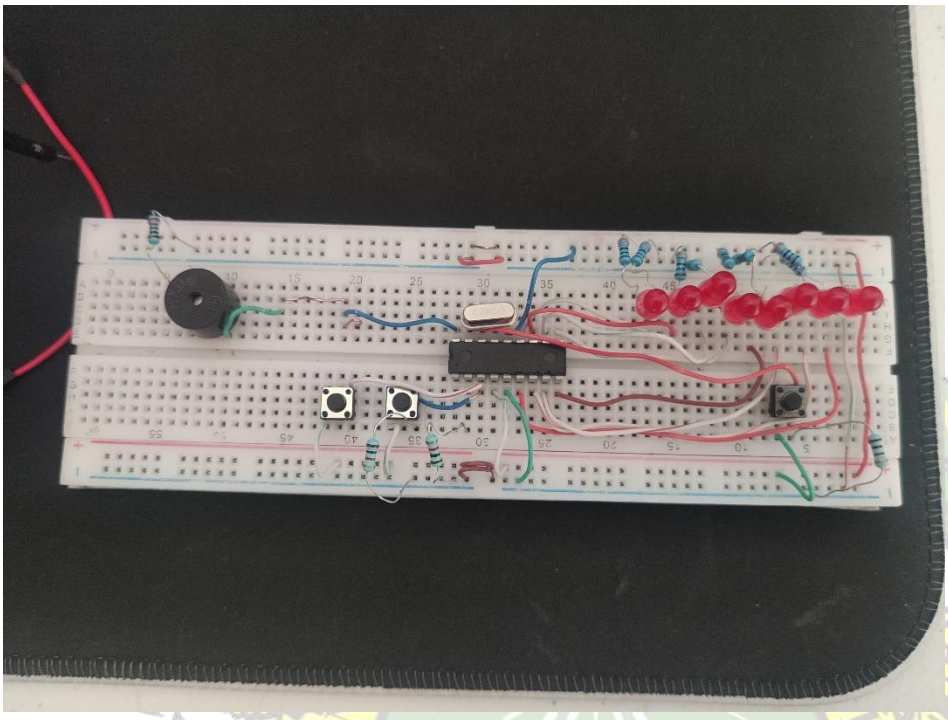
```
sec_tiempo:
    Part3:
        movlw 1
        movwf cont3
    vuelta3:
        movlw 150
        movwf cont2
    vuelta2:
        movlw 255
        movwf cont1
    vuelta1:
        decfsz cont1,1
        goto vuelta1

        decfsz cont2,1
        goto vuelta2

        decfsz cont3,1
        goto vuelta3
    return
```

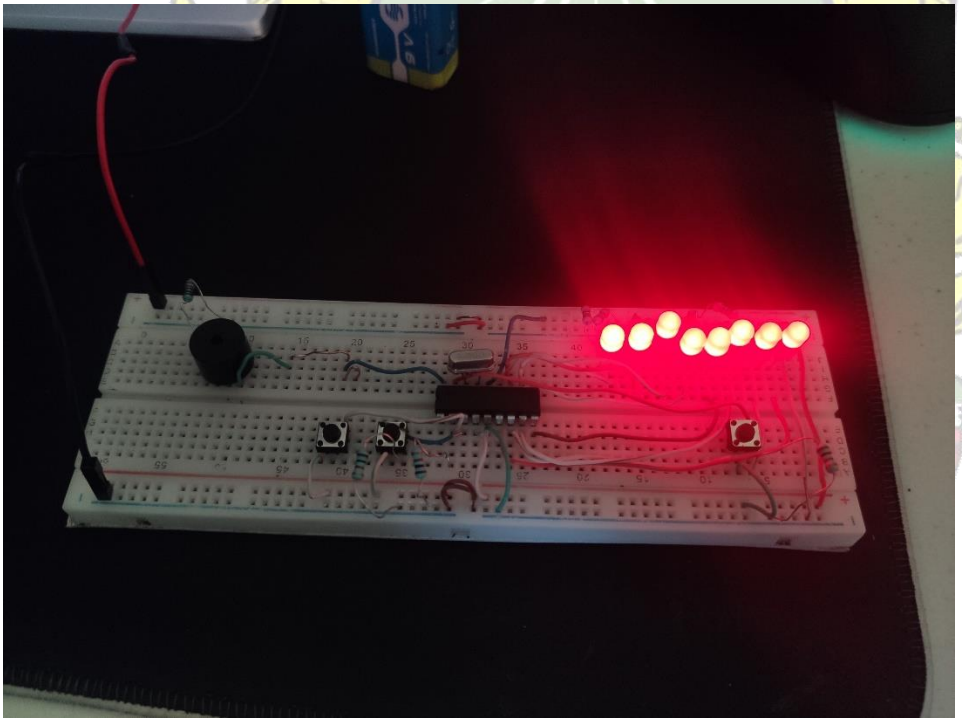
```
sec_inicio:
    P3:
        movlw 2
        movwf cont3
    v3:
        movlw 150
        movwf cont2
    v2:
        movlw 255
        movwf cont1
    v1:
        decfsz cont1,1
        goto v1

        decfsz cont2,1
        goto v2

        decfsz cont3,1
        goto v3
return

restartGame:
    clrf PORTA
    btfss PORTA,3
    goto restartGame
    goto startGame

gameOver:
    END Programa
```

Diagrama eléctrico.

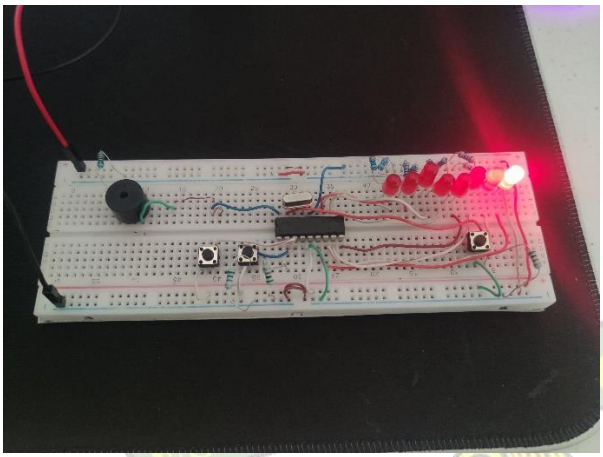


Circuito funcionando.

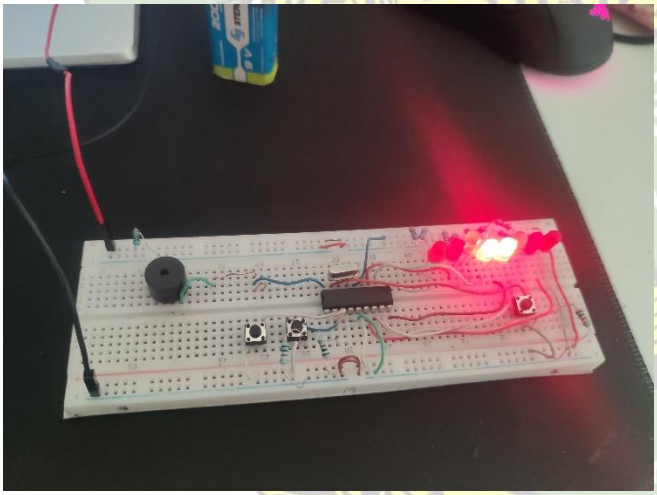




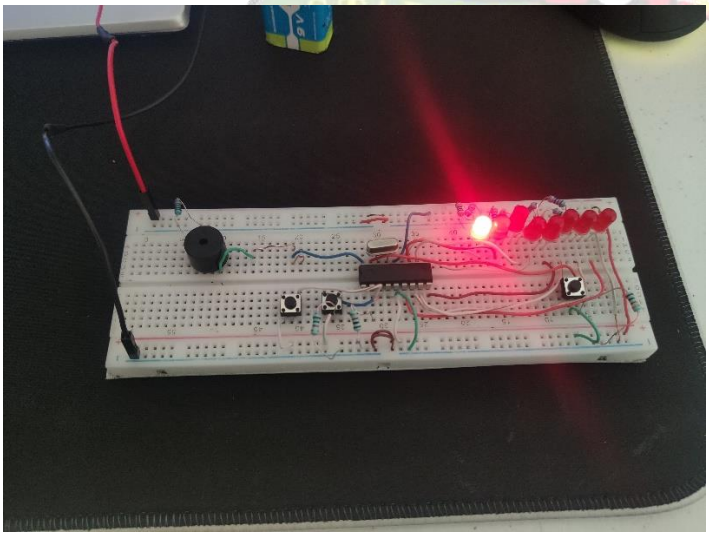
Saque del P1



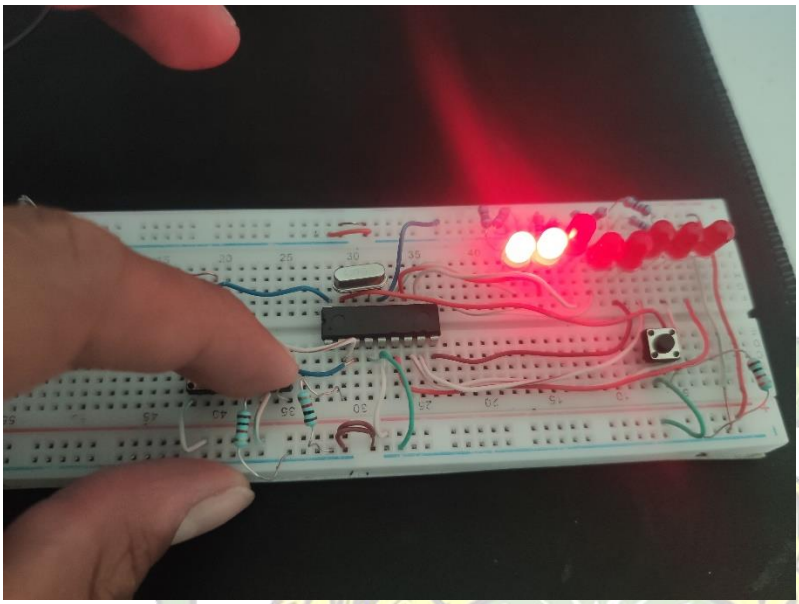
Animación del golpeo de P1, DerIzq



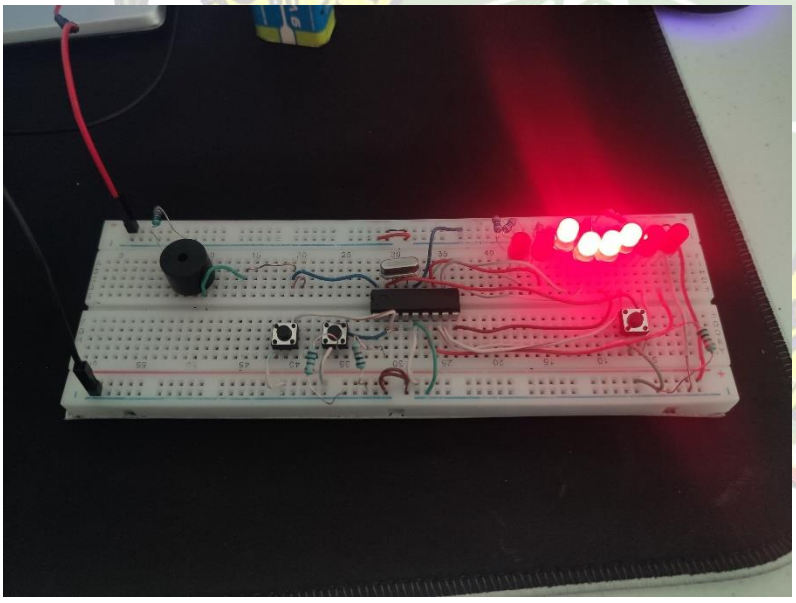
Saque del P2



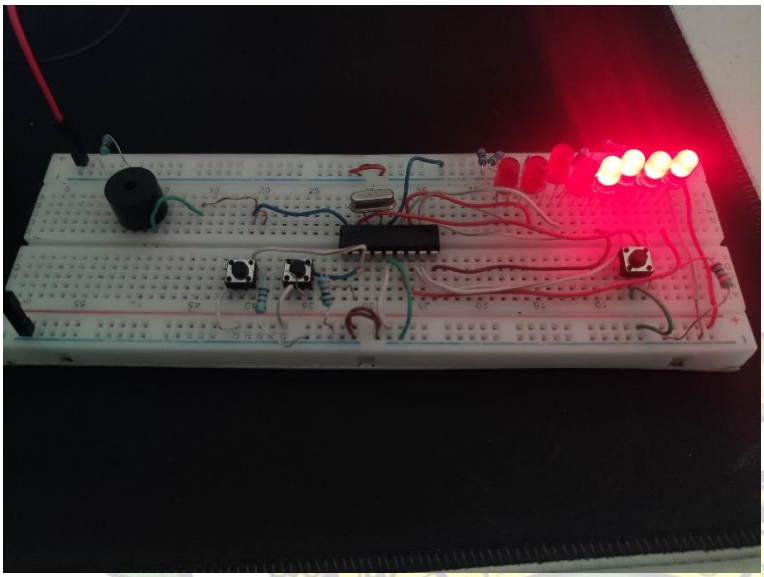
Animación del golpeo de P2, IzqDer



Simulación de EMPATE



Simulación de P1 VICTORY



Simulación de P2 VICTORY