INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

PROGRAMA ACADÉMICO
INGENIERÍA EN SISTEMAS COMPUTACIONALES

INTRODUCCIÓN A LOS MICROCONTROLADORES

“Proyecto FINAL Juego de Ping-Pong”

Alumnos:

Rojas Alvarado Luis Enrique

Grupo:
3CM6

Link video explicativo:
https://youtu.be/m1J4vdR0EFQ

Profesor:
Fernando Aguilar Sánchez

OBJETIVO

Al término de este semestre los alumnos tendrán la capacidad para diseñar y elaborar un proyecto final.

Desarrollo Experimental

1.- Diseñe un Juego de Ping-Pong con las siguientes características armando su circuito final en “PLACA”:

a) Use una Matriz de leds de 7x5 o de 8x8.

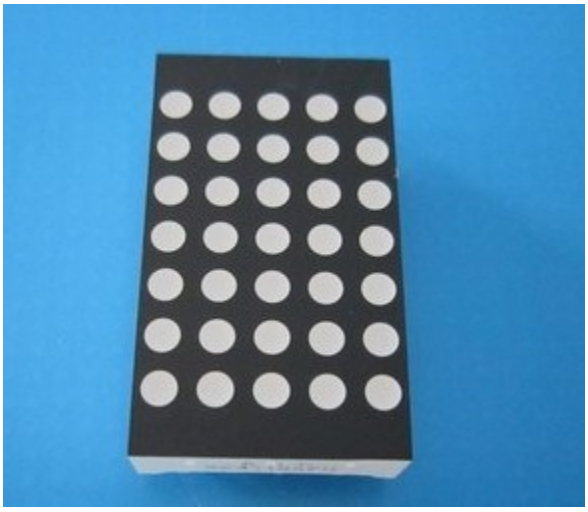
b) En la matriz de leds la pelota rebotará en las orillas y la raqueta estará formada por 2 puntos en la base de la matriz.

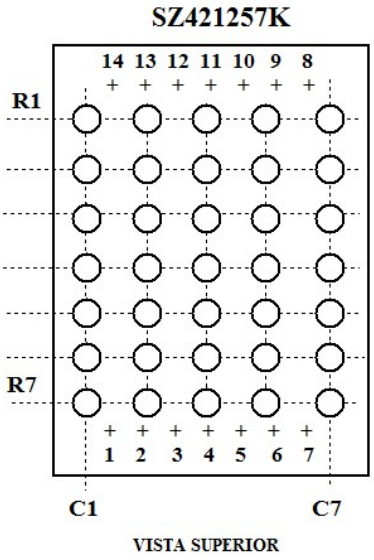
c) Se marcará un punto en el display de 7 segmentos por cada pelota que el jugador no alcance con la raqueta.

d) Los push button sirven para mover la raqueta de derecha a izquierda y viceversa.

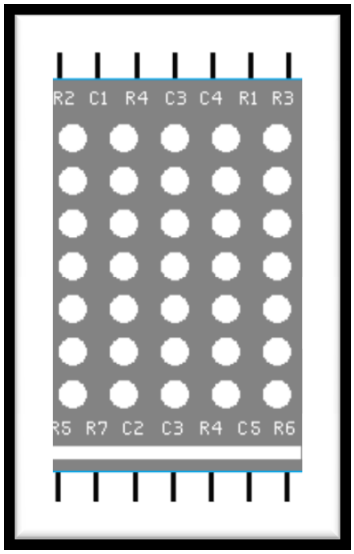
e) Para la entrega del Proyecto se debe anexar un informe en el que debe incluir su diseño, diagramas eléctricos y código del programa aplicado.

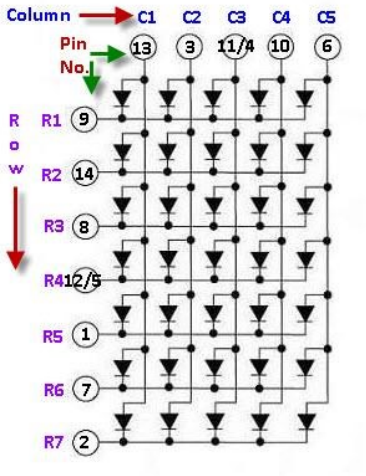
La matriz de LEDs no es más que un arreglo de LEDs agrupados dentro de un encapsulado, los cuales se encuentran agrupados en forma de matriz. Este acomodo nos ayuda para poder generar cualquier cosa que nosotros queramos siempre y cuando se pueda representar dentro de la matriz.





La matriz de LEDs que se usara en esta práctica es una como la de la foto superior, esta es de 5 columnas por 7 filas, las columnas son representadas por una C y las filas por una R, en la imagen inferior podemos ver como se encuentran distribuidos los pines de la matriz a usar.



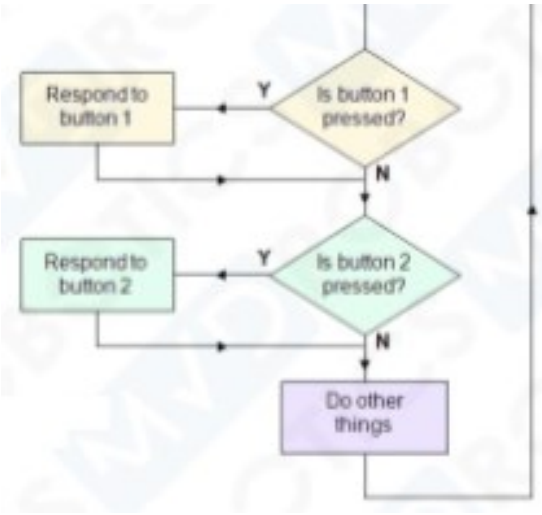


Interrupciones: Básicamente, la interrupción es un mecanismo mediante el cual el CPU puede, ante cierto evento, suspender lo que está haciendo en ese momento y pasar a atender una rutina de alta prioridad. Una vez finalizada ésta, el CPU vuelve a su actividad anterior. - La interrupción es disparada por un evento externo al CPU. Puede ser el cambio de estado de un pin (interrupción externa) o cierta señal de un dispositivo interno del uC, por ejemplo el desborde de un timer, el estado del ADC, etc

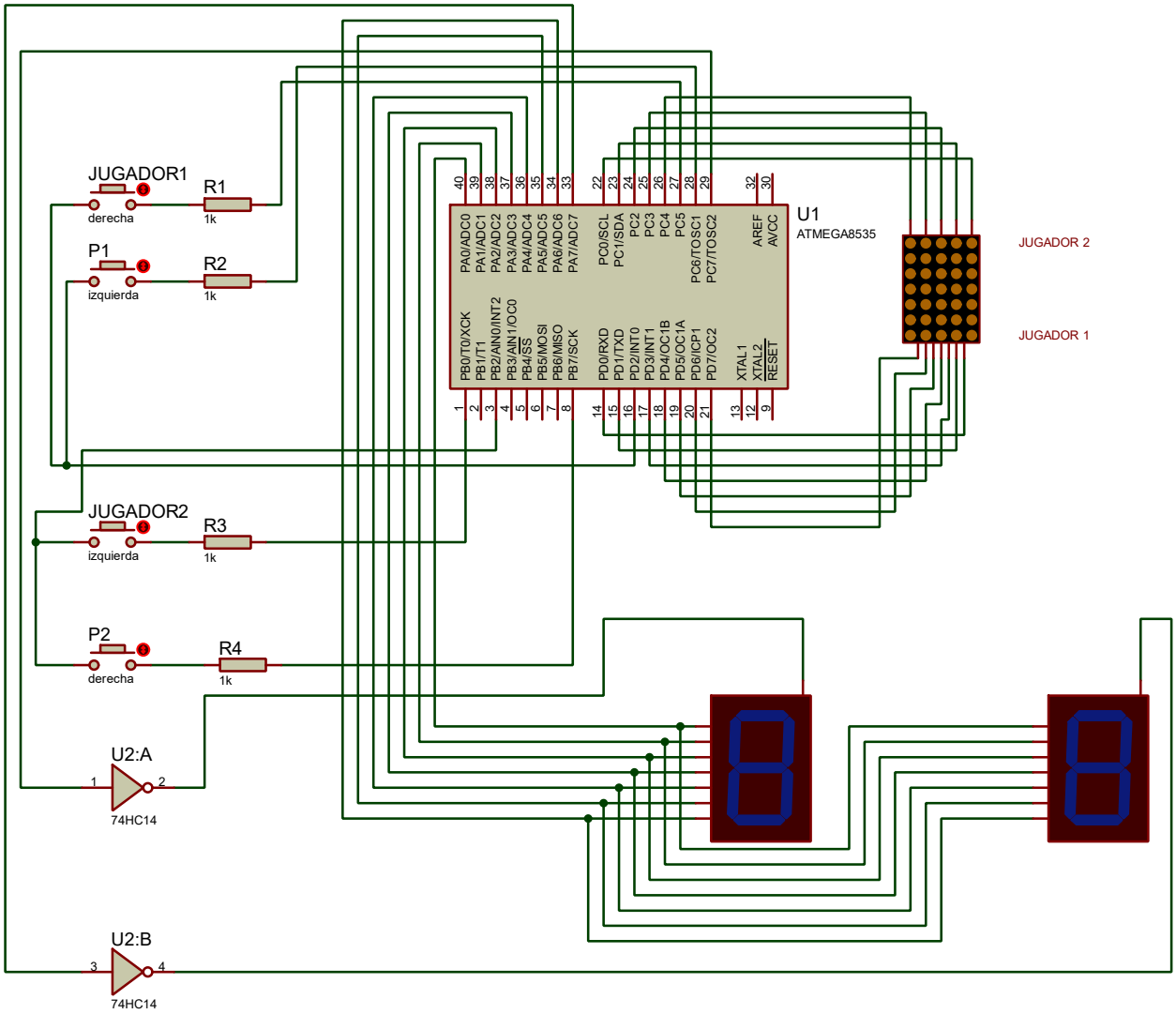
En algunos procesadores también existen las interrupciones generadas por software (lo cual tiene sentido en el contexto de un 5.0.).

Cuando el CPU recibe la señal de interrupción, abandona (interrumpe) inmediatamente lo que está haciendo y salta a una rutina especial. Dicha rutina suele llamarse interrupt Handler o interrupt Service Routine (ISR). Inhrrupt - Una vez que la ISR termina, el CPU retoma su actividad anterior desde el punto en que la dejó. —Antes de atender la interrupción, el CPU guarda el estado de sus registros (incluyendo el Program Counter), para poder continuar luego con la ejecución de las instrucciones del programa principal. Ese ”estado” del CPU se llama el contexto, y la operación de salvarlo y restituirlo se llama cambio de contexto. - Las interrupciones son señales aleatorias, externas al CPU, que pueden ocurrir en cualquier momento, fuera del flujo predecible de un programa. Es un proceso asincrono; funciona como si fuera otro hilo de ejecución (thread).

El microcontrolador dispone de mecanismos para activar, desactivar, priorizar, inhibir, etc. , distintas fuentes de interrupciones, y para asociar cada una de ellas con rutinas ISR programadas por el usuario.



Circuito



Código

```
/*****************************************************
This program was produced by the
CodeWizardAVR V2.05.0 Professional
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 24/10/2019
Author  : NeVaDa
Company :
Comments:


Chip type               : ATmega8535
Program type            : Application
AVR Core Clock frequency: 1.000000 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 128
*****************************************************/

```

```
#include <mega8535.h>
#include <delay.h>

eeprom short random;
eeprom short barra1;
eeprom short barra2;
int rand;
//
int cont1,cont2;
int puntos1,puntos2;
int i,j;
int x,y;
int direccion;
int dsplz;
int stay;
int cols;
int inst;
//
int win;
int rapidez;
int rapidez1;
float rapidez2;
int gan1;
int gan2;
int gan3;
int gan4;
const char tabla7segmentos[2][10]={
{0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x10},
{0xc0,0xcf,0xa4,0x86,0x8b,0x92,0x90,0xc7,0x80,0x82}
};

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    // Place your code here
    if(cont1>20){
    cont1=0;
    if(inst){
        //Derecha
        barra1++;
        if(barra1>3){
            barra1=3;
        }
    }else{
        //Izquierda
        barra1--;
        if(barra1<0){
            barra1=0;
        }
    }
    }else{
    cont1++;
    }
}

// External Interrupt 2 service routine
interrupt [EXT_INT2] void ext_int2_isr(void)
{
// Place your code here
    if(cont2>20){
    cont2=0;
    if(inst){
        //Izquierda
        barra2--;
        if(barra2<0){
            barra2=0;
        }
    }else{
```

```
92.          //Derecha
93.          barra2++;
94.          if(barra2>3){
95.              barra2=3;
96.          }
97.
98.     }
99.     }else{
100.           cont2++;
101.           }
102.
103.       }
104.
105.       // Declare your global variables here
106.
107.       void main(void)
108.       {
109.       // Declare your local variables here
110.
111.       // Input/Output Ports initialization
112.       // Port A initialization
113.       // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
114.       // State7=0 State6=1 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
115.       PORTA=0x40;
116.       DDRA=0xFF;
117.
118.       // Port B initialization
119.       // Func7=Out Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=Out
120.       // State7=1 State6=P State5=P State4=P State3=P State2=P State1=P State0=1
121.       PORTB=0xFF;
122.       DDRB=0x81;
123.
124.       // Port C initialization
125.       // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
126.       // State7=0 State6=1 State5=1 State4=0 State3=0 State2=0 State1=0 State0=0
127.       PORTC=0x60;
128.       DDRC=0xFF;
129.
130.       // Port D initialization
131.       // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=In Func1=Out Func0=Out
132.       // State7=1 State6=1 State5=1 State4=1 State3=1 State2=P State1=1 State0=1
133.       PORTD=0xFF;
134.       DDRD=0xFB;
135.
136.       // Timer/Counter 0 initialization
137.       // Clock source: System Clock
138.       // Clock value: Timer 0 Stopped
139.       // Mode: Normal top=0xFF
140.       // OC0 output: Disconnected
141.       TCCR0=0x00;
142.       TCNT0=0x00;
143.       OCR0=0x00;
144.
145.       // Timer/Counter 1 initialization
146.       // Clock source: System Clock
147.       // Clock value: Timer1 Stopped
148.       // Mode: Normal top=0xFFFF
149.       // OC1A output: Discon.
150.       // OC1B output: Discon.
151.       // Noise Canceler: Off
152.       // Input Capture on Falling Edge
153.       // Timer1 Overflow Interrupt: Off
154.       // Input Capture Interrupt: Off
155.       // Compare A Match Interrupt: Off
156.       // Compare B Match Interrupt: Off
157.       TCCR1A=0x00;
158.       TCCR1B=0x00;
159.       TCNT1H=0x00;
```

```
160.        TCNT1L=0x00;
161.        ICR1H=0x00;
162.        ICR1L=0x00;
163.        OCR1AH=0x00;
164.        OCR1AL=0x00;
165.        OCR1BH=0x00;
166.        OCR1BL=0x00;
167.
168.        // Timer/Counter 2 initialization
169.        // Clock source: System Clock
170.        // Clock value: Timer2 Stopped
171.        // Mode: Normal top=0xFF
172.        // OC2 output: Disconnected
173.        ASSR=0x00;
174.        TCCR2=0x00;
175.        TCNT2=0x00;
176.        OCR2=0x00;
177.
178.        // External Interrupt(s) initialization
179.        // INT0: On
180.        // INT0 Mode: Falling Edge
181.        // INT1: Off
182.        // INT2: On
183.        // INT2 Mode: Falling Edge
184.        GICR|=0x60;
185.        MCUCR=0x02;
186.        MCUCSR=0x00;
187.        GIFR=0x60;
188.
189.        // Timer(s)/Counter(s) Interrupt(s) initialization
190.        TIMSK=0x00;
191.
192.        // USART initialization
193.        // USART disabled
194.        UCSRB=0x00;
195.
196.        // Analog Comparator initialization
197.        // Analog Comparator: Off
198.        // Analog Comparator Input Capture by Timer/Counter 1: Off
199.        ACSR=0x80;
200.        SFIOR=0x00;
201.
202.        // ADC initialization
203.        // ADC disabled
204.        ADCSRA=0x00;
205.
206.        // SPI initialization
207.        // SPI disabled
208.        SPCR=0x00;
209.
210.        // TWI initialization
211.        // TWI disabled
212.        TWCR=0x00;
213.
214.        // Global enable interrupts
215.        #asm("sei")
216.        if(random<1){
217.            random=1;
218.        }
219.        //multiplicar por numero primo
220.        random=(random*7)+2;
221.        random=random%17;
222.        //definir direccion
223.        rand=(int)random;
224.        rand*=7;
225.        direccion=rand%4;
226.        //definir x
227.        x=2;
```

```
228.        //definir y
229.        y=3;
230.        //
231.        if(!(barra1>=0 && barra1<=3)){
232.            barra1=0;
233.        }
234.        if(!(barra2>=0 && barra2<=3)){
235.            barra2=3;
236.        }
237.        //------------
238.        cont1=0;
239.        cont2=0;
240.        i=j=0;
241.        rapidez1=99;
242.        rapidez=rapidez1;
243.        rapidez2=(float)(rapidez1);
244.        dsplz=0;
245.        stay=0;
246.        cols=0;
247.        puntos1=0;
248.        puntos2=0;
249.        inst=0;
250.        gan1=0;
251.        gan2=0;
252.        gan3=0;
253.        gan4=0;
254.        win=0;
255.
256.        while (1)
257.              {
258.              // Place your code here
259.              if(!win){
260.                delay_ms(1);
261.                //
262.                //
263.                cols++;
264.                if(cols>4){
265.                    cols=0;
266.                }
267.                //
268.                //
269.                PORTA &=0x7f;
270.                PORTC &=0x7f;
271.                PORTB |=0x81;
272.                PORTC |=0x60;
273.                //
274.                inst++;
275.                if(inst>1){
276.                    inst=0;
277.                }
278.                //
279.                if(inst){
280.                    PORTA =tabla7segmentos[inst][puntos2];
281.                    //PORTA |=0x80;
282.                    PORTC &=0xbf;
283.                    PORTB &=0x7f;
284.                }else{
285.                    PORTA =tabla7segmentos[inst][puntos1];
286.                    PORTC |=0x80;
287.                    PORTC &=0xdf;
288.                    PORTB &=0xfe;
289.                }
290.              //Matriz de LEDs
291.              if(dsplz>rapidez){
292.                dsplz=0;
293.              if(!stay){
294.                stay=1;
295.              }else{
```

```
296.                stay=0;
297.                //Validaciones
298.                switch(direccion){
299.                    case 0: //noroeste
300.                        if((x-1)<0){
301.                            if(y==1){
302.                                if(barra2==0 || barra2==1){
303.                                    direccion=2;
304.
305.                                }else{
306.                                    direccion=1;
307.                                }
308.                            }else if(y>=2){
309.                                direccion=1;
310.                            }
311.                        }else{
312.                            if(y==1){
313.                                if(barra2==0){
314.                                        if(x==1){
315.                                            direccion=3;
316.                                        }
317.                                        if(x==2){
318.                                            direccion=2;
319.                                        }
320.                                }else if(barra2==1){
321.                                        if(x==2){
322.                                            direccion=3;
323.                                        }
324.                                        if(x==3){
325.                                            direccion=2;
326.                                        }
327.                                }else if(barra2==2){
328.                                        if(x==3){
329.                                            direccion=3;
330.                                        }
331.                                }
332.                            }else if(y<=2){
333.                                //direccion=0;
334.                            }
335.                        }
336.                        break;
337.                    case 1: //noreste
338.                        if((x+1)>4){
339.                            if(y==1){
340.                                if(barra2==2 || barra2==3){
341.                                    direccion=3;
342.
343.                                }else{
344.                                    direccion=0;
345.                                }
346.                            }else if(y>=2){
347.                                direccion=0;
348.                            }
349.                        }else{
350.                            if(y==1){
351.                                if(barra2==1){
352.                                        if(x==1){
353.                                            direccion=2;
354.                                        }
355.                                }else if(barra2==2){
356.                                        if(x==2){
357.                                            direccion=2;
358.                                        }
359.                                        if(x==1){
360.                                            direccion=3;
361.                                        }
362.                                }else if (barra2==3){
363.                                        if(x==3){
```

```
                                        direccion=2;
                                    }
                                    if(x==2){
                                        direccion=3;
                                    }
                                }
                            }else if(y>=2){
                                //direccion=1;
                            }
                        }
                        break;
                    case 2: //sureste
                        if((x+1)>4){
                            if(y==5){
                                if(barra1==2 || barra1==3){
                                    direccion=0;

                                }else{
                                    direccion=3;
                                }
                            }else if(y<=4){
                                direccion=3;
                            }
                        }else{
                            if(y==5){
                                if(barra1==1){
                                        if(x==1){
                                            direccion=1;
                                        }
                                }else if(barra1==2){
                                        if(x==2){
                                            direccion=1;
                                        }
                                        if(x==1){
                                            direccion=0;
                                        }
                                }else if (barra1==3){
                                        if(x==3){
                                            direccion=1;
                                        }
                                        if(x==2){
                                            direccion=0;
                                        }
                                }
                            }else if(y<=4){
                                //direccion=2;
                            }
                        }
                        break;
                    case 3: //suroeste
                        if((x-1)<0){
                            if(y==5){
                                if(barra1==0 || barra1==1){
                                    direccion=1;

                                }else{
                                    direccion=2;
                                }
                            }else if(y<=4){
                                direccion=2;
                            }
                        }else{
                            if(y==5){
                                if(barra1==0){
                                        if(x==1){
                                            direccion=0;
                                        }
                                        if(x==2){
```

```
                                        direccion=1;
                                    }
                                }else if(barra1==1){
                                        if(x==2){
                                            direccion=0;
                                        }
                                        if(x==3){
                                            direccion=1;
                                        }
                                }else if(barra1==2){
                                        if(x==3){
                                            direccion=0;
                                        }
                                }
                            }else if(y<=4){
                                //direccion=3;
                            }
                        }
                        break;
                    default:
                        break;
                }
                //
                switch(direccion){
                    case 0: //noroeste
                        x--;
                        y--;
                        break;
                    case 1: //noreste
                        x++;
                        y--;
                        break;
                    case 2: //sureste
                        x++;
                        y++;
                        break;
                    case 3: //suroeste
                        x--;
                        y++;
                        break;
                    default:
                        break;
                }
            }
            }else{
              dsplz++;
            }
            //
              PORTD |=0xfb;
              PORTC &=0xe0;
              if((cols%5)==x){
              switch(y){
                  case 0:
                      PORTD &=0x7f;
                      break;
                  case 1:
                      PORTD &=0xbf;
                      break;
                  case 2:
                      PORTD &=0xdf;
                      break;
                  case 3:
                      PORTD &=0xef;
                      break;
                  case 4:
                      PORTD &=0xf7;
                      break;
                  case 5:
```

```
                        PORTD &=0xfd;
                        break;
                    case 6:
                        PORTD &=0xfe;
                        break;
                    default:
                        break;
                }
                }
                //
                switch((cols%5)){
                    case 0:
                        if(barra1==0){
                            PORTD &=0xfe;
                        }
                        if(barra2==0){
                            PORTD &=0x7f;
                        }
                        PORTC |=0x01;
                        break;
                    case 1:
                        if(barra1==0){
                            PORTD &=0xfe;
                        }
                        if(barra2==0){
                            PORTD &=0x7f;
                        }
                        if(barra1==1){
                            PORTD &=0xfe;
                        }
                        if(barra2==1){
                            PORTD &=0x7f;
                        }
                        PORTC |=0x02;
                        break;
                    case 2:
                        if(barra1==1){
                            PORTD &=0xfe;
                        }
                        if(barra2==1){
                            PORTD &=0x7f;
                        }
                        if(barra1==2){
                            PORTD &=0xfe;
                        }
                        if(barra2==2){
                            PORTD &=0x7f;
                        }
                        PORTC |=0x04;
                        break;
                    case 3:
                        if(barra1==2){
                            PORTD &=0xfe;
                        }
                        if(barra2==2){
                            PORTD &=0x7f;
                        }
                        if(barra1==3){
                            PORTD &=0xfe;
                        }
                        if(barra2==3){
                            PORTD &=0x7f;
                        }
                        PORTC |=0x08;
                        break;
                    case 4:
                        if(barra1==3){
                            PORTD &=0xfe;
```

```
                    }
                    if(barra2==3){
                        PORTD &=0x7f;
                    }
                    PORTC |=0x10;
                    break;
                default:
                    break;
            }
          //
            if(y<0 || y>6){
            PORTD |=0xfb;
            PORTC &=0xe0;
            if(y>6){
                puntos2++;
                if(puntos2>9){
                    win=1;
                }
                //redefinir direccion
                rand*=7;
                rand=rand%11;
                direccion=rand%2;

            }
            if(y<0){
                puntos1++;
                if(puntos1>9){
                    win=1;
                }
                //redefinir direccion
                rand*=7;
                rand=rand%11;
                direccion=(rand%2)+2;

            }
            //redefinir x
            rand*=19;
            rand+=7;
            x=1+(rand%3);
            //redefinir y
            y=3;
            //
            if(puntos1>8 && puntos2>8){
                rapidez2=(float)(rapidez1);
                rapidez2=rapidez2*33/100;
                rapidez=(int)(rapidez2);
            }else if(puntos1>7 && puntos2>7){
                rapidez2=(float)(rapidez1);
                rapidez2=rapidez2*39/100;
                rapidez=(int)(rapidez2);
            }else if(puntos1>6 && puntos2>6){
                rapidez2=(float)(rapidez1);
                rapidez2=rapidez2*41/100;
                rapidez=(int)(rapidez2);
            }else if(puntos1>5 && puntos2>5){
                rapidez2=(float)(rapidez1);
                rapidez2=rapidez2*53/100;
                rapidez=(int)(rapidez2);
            }else if(puntos1>4 && puntos2>4){
                rapidez2=(float)(rapidez1);
                rapidez2=rapidez2*63/100;
                rapidez=(int)(rapidez2);
            }else if(puntos1>3 && puntos2>3){
                rapidez2=(float)(rapidez1);
                rapidez2=rapidez2*73/100;
                rapidez=(int)(rapidez2);
            }else if(puntos1>2 && puntos2>2){
                rapidez2=(float)(rapidez1);
```

```
                    rapidez2=rapidez2*83/100;
                    rapidez=(int)(rapidez2);
                }else if(puntos1>1 && puntos2>1){
                    rapidez2=(float)(rapidez1);
                    rapidez2=rapidez2*93/100;
                    rapidez=(int)(rapidez2);
                }
                delay_ms(200);
                }
                //
              //
              }
              }
      }
```

Conclusiones

En la realización de esta práctica, la capacidad para diseñar y elaborar un proyecto final.

Aprendí la interacción del microprocesador con los periféricos asociados a esta práctica, reforzando así mis conocimientos de la arquitectura de este así como de los funcionamientos del periférico.

Referencias

[1] P. flotante, «Módulo matriz de leds rojos de 7x5, ánodo a renglón SZ411257N,» PuntoFotante.net, [En línea]. Available: https://www.puntoflotante.net/MATRIZ-DE-LEDS-DE-7X5-A.htm. [Último acceso: 21 Enero 2021].

[2] Pablo Gindel, «Microcontroladores 6 – interrupciones,» SlideShare, 30 Septiembre 2015. [En línea]. Available: https://es.slideshare.net/pablogindel/microcontroladores-6-interrupciones#:~:text=Básicamente%2C%20la%20interrupción%20es%20un,vuelve%20a%20su%20actividad%20anterior.. [Último acceso: 21 Enero 2021].