INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

PRÁCTICA 20: PROYECTO FINAL

JUEGO DE PING-PONG

ALUMNOS: MALAGON BAEZA ALAN ADRIAN
MARTINEZ CHAVEZ JORGE ALEXIS

GRUPO: 6CM3

U.A: SISTEMAS EN CHIP

PROFESOR: FERNANDO AGUILAR SÁNCHEZ

FECHA DE ENTREGA: 18 DE JUNIO DE 2023

OBJETIVO

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

INTRODUCCIÓN TEÓRICA

Un microcontrolador es un computador completo (microprocesador + E/S + memoria + otros periféricos), aunque de limitadas prestaciones, que está contenido en el chip de un circuito integrado programable y se destina a gobernar una sola tarea con el programa que reside en su memoria. Sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar.

El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen las líneas que gobiernan los periféricos.

Cuando necesitamos que transcurra un determinado tiempo de espera antes de que ocurra un evento como por ejemplo el encendido de una luz, LED, activación de una bobina de un relé o lectura de una determinada entrada, se suele recurrir a los retardos. Prácticamente casi todos los programas de microcontroladores usan en algún momento una rutina de retardo.

Las extensas áreas de aplicación de los microcontroladores, que se pueden considerar ilimitadas, como pueden ser juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de arranque de nuestro coche, etc.

Los microcontroladores están diseñados para reducir el costo económico y el consumo de energía de un sistema en particular. Por eso el tamaño de la unidad central de procesamiento, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación.

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 Ia 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). lnhrrupt - Una vez que Ia ISR termina, el CPU retoma su actividad anterior desde el punto en que Ia dejó. —Antes de atender la interrupción, el CPU guarda el estado de sus registros (incluyendo el Program Counter), para poder continuar luego con Ia 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.

Función de retardo:

La generación de retardo de tiempo o “delay” es el concepto más importante en los sistemas integrados. La mayoría de las veces, necesitamos generar un retardo de tiempo preciso entre dos acciones en cualquier aplicación de microcontrolador. Podemos generar el retardo de tiempo usando técnicas como LOOP o usando funciones de retardo integradas.

La mayoría de los controladores tienen temporizadores incorporados. Estos temporizadores no solo se utilizan para generar retardos de tiempo, sino que también se utilizan para contar. El valor del contador se incrementa en 1 cuando ocurre una acción o un evento.

MATERIALES Y EQUIPO EMPLEADO

- CodeVisionAVR
- Simulador Proteus
    - Microcontrolador ATmega8435
    - 2 Displays de ánodo común
    - 4 resistores de 1k
    - 7 resistores de 330
    - 1 matriz de leds de 7x5
    - 4 push buttons

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.

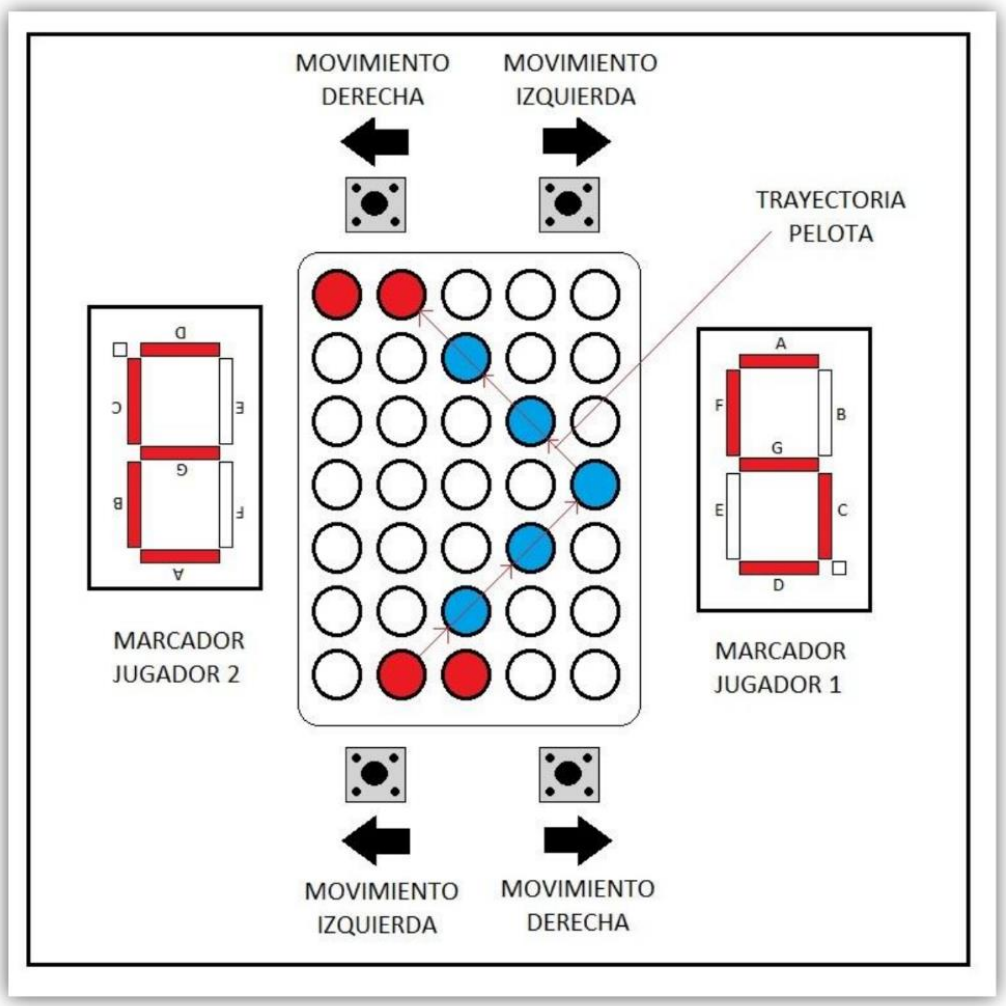

Figura 1. Circuito para el ping-pong.

CÓDIGO GENERADO POR CODEVISION

```
/*****************************************************
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    : 28/05/2023
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>

// Alphanumeric LCD functions
#include <alcd.h>
const char tabla[10] = {48, 49, 50, 51, 52, 53, 54, 55, 56, 57};


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{
        //Derecha
        barra2++;
        if(barra2>3){
            barra2=3;
        }

    }
    }else{
    cont2++;
    }

}

// Declare your global variables here

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
//  Func7=Out  Func6=Out  Func5=Out  Func4=Out  Func3=Out  Func2=Out  Func1=Out
Func0=Out
// State7=0 State6=1 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTA=0x40;
DDRA=0xFF;

// Port B initialization
// Func7=Out Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=Out
// State7=1 State6=P State5=P State4=P State3=P State2=P State1=P State0=1
PORTB=0xFF;
DDRB=0x81;

// Port C initialization
//  Func7=Out  Func6=Out  Func5=Out  Func4=Out  Func3=Out  Func2=Out  Func1=Out
Func0=Out
// State7=0 State6=1 State5=1 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x60;
DDRC=0xFF;

// Port D initialization
//  Func7=Out  Func6=Out  Func5=Out  Func4=Out  Func3=Out  Func2=In  Func1=Out
Func0=Out
// State7=1 State6=1 State5=1 State4=1 State3=1 State2=P State1=1 State0=1
PORTD=0xFF;
```

```
DDRD=0xFB;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Falling Edge
// INT1: Off
// INT2: On
```

```
// INT2 Mode: Falling Edge
GICR|=0x60;
MCUCR=0x02;
MCUCSR=0x00;
GIFR=0x60;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// USART disabled
UCSRB=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Global enable interrupts
#asm("sei")
if(random<1){
    random=1;
}
//multiplicar por numero primo
random=(random*7)+2;
random=random%17;
//definir direccion
rand=(int)random;
rand*=7;
direccion=rand%4;
//definir x
x=2;
//definir y
y=3;
//
if(!(barra1>=0 && barra1<=3)){
```

```
    barra1=0;
}
if(!(barra2>=0 && barra2<=3)){
    barra2=3;
}
//------------
cont1=0;
cont2=0;
i=j=0;
rapidez1=99;
rapidez=rapidez1;
rapidez2=(float)(rapidez1);
dsplz=0;
stay=0;
cols=0;
puntos1=0;
puntos2=0;
inst=0;
gan1=0;
gan2=0;
gan3=0;
gan4=0;
win=0;

while (1)
      {

      // Place your code here
      if(!win){


        delay_ms(1);

        //
        //
        cols++;
        if(cols>4){
            cols=0;
        }
        //
        //
        PORTA &=0x7f;
        PORTC &=0x7f;
        PORTB |=0x81;
        PORTC |=0x60;
        //
        inst++;
        if(inst>1){
            inst=0;
```

```
  }
  //
  if(inst){
      PORTA =tabla7segmentos[inst][puntos2];
      //PORTA |=0x80;
      PORTC &=0xbf;
      PORTB &=0x7f;
  }else{
      PORTA =tabla7segmentos[inst][puntos1];
      PORTC |=0x80;
      PORTC &=0xdf;
      PORTB &=0xfe;
  }
//Matriz de LEDs
if(dsplz>rapidez){
  dsplz=0;
if(!stay){
  stay=1;
}else{
  stay=0;
  //Validaciones
      switch(direccion){
            case 0: //noroeste
                 if((x-1)<0){
              if(y==1){
                  if(barra2==0 || barra2==1){
                      direccion=2;

                  }else{
                      direccion=1;
                  }
              }else if(y>=2){
                  direccion=1;
              }
          }else{
              if(y==1){
                  if(barra2==0){
                          if(x==1){
                              direccion=3;
                          }
                          if(x==2){
                              direccion=2;
                          }
                  }else if(barra2==1){
                          if(x==2){
                              direccion=3;
                          }
                          if(x==3){
                              direccion=2;
```

```
                }
        }else if(barra2==2){
                if(x==3){
                    direccion=3;
                }
        }
    }else if(y<=2){
        //direccion=0;
    }
}
         break;
  case 1: //noreste
         if((x+1)>4){
    if(y==1){
        if(barra2==2 || barra2==3){
            direccion=3;

        }else{
            direccion=0;
        }
    }else if(y>=2){
        direccion=0;
    }
}else{
    if(y==1){
        if(barra2==1){
                if(x==1){
                    direccion=2;
                }
        }else if(barra2==2){
                if(x==2){
                    direccion=2;
                }
                if(x==1){
                    direccion=3;
                }
        }else if (barra2==3){
                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);
          }
          //
        //
        }
        }
}
```

CIRCUITO ELECTRICO EN PROTEUS

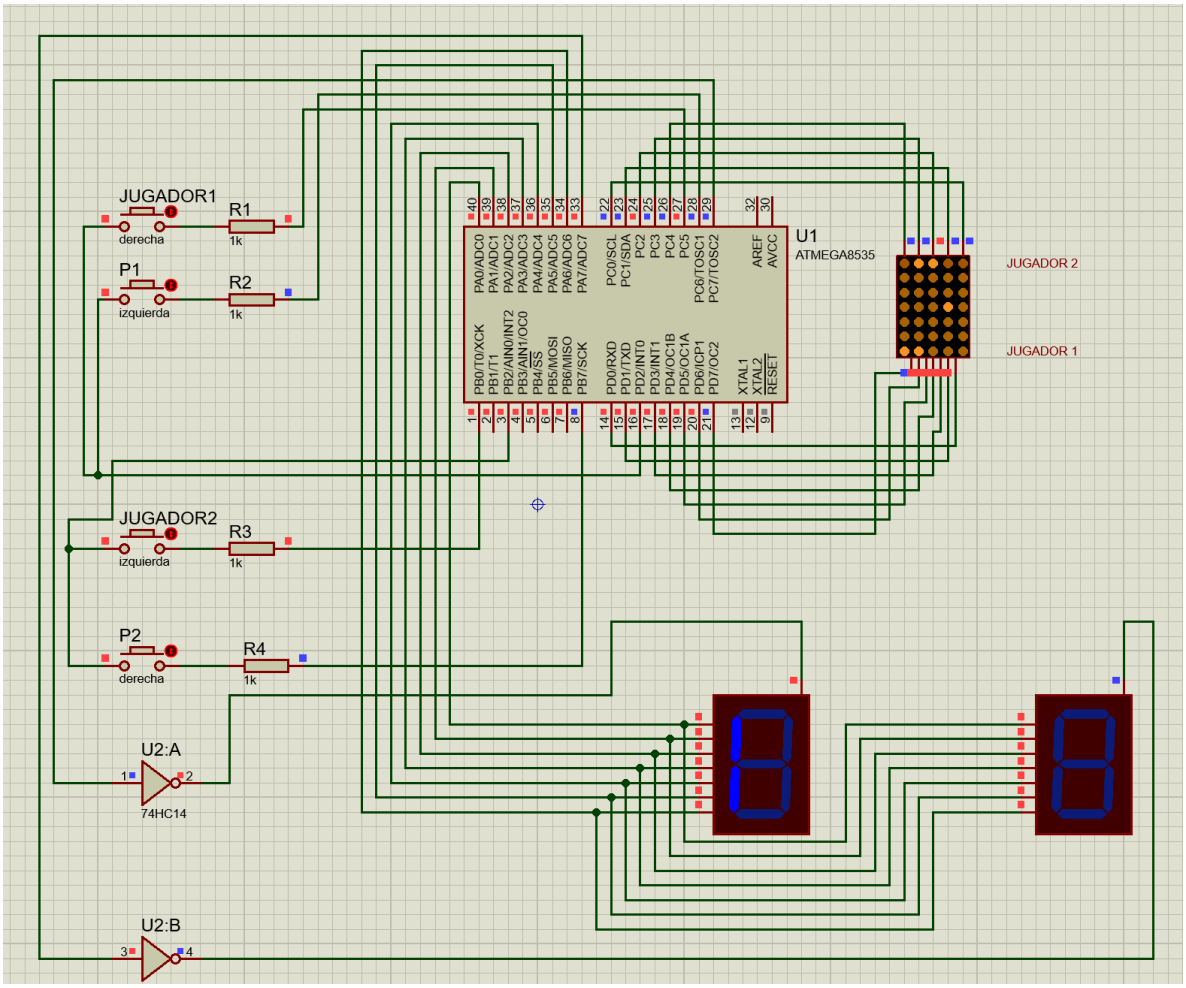

Figura 2. Circuito Completo

CIRCUITO EN EL PROTO ARMADO

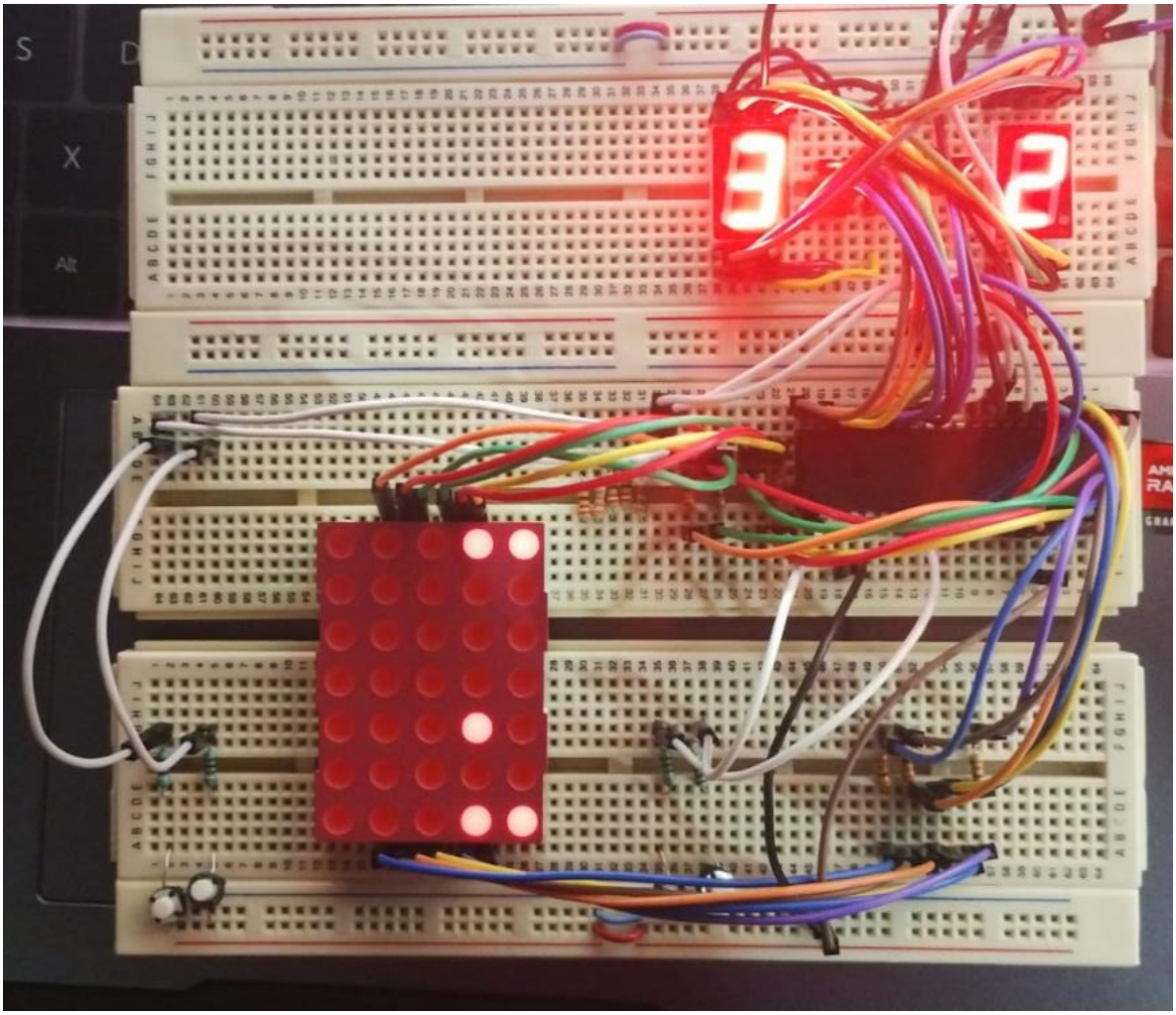

Figura 3. Circuito en protoboard

OBSERVACIONES Y CONCLUSION

Tras realizar la práctica fue posible aplicar los conocimientos de las prácticas anteriores y también se le dio uso a la matriz 7x5 para mostrar distintos patrones manipulando arreglos. Lo más complicado fue programar el movimiento de la pelota, pero fue de gran ayuda haber realizado las prácticas anteriores para entender mejor el manejo de arreglos y usar correctamente la matriz.

BIBLIOGRAFÍA

- Wikipedia, «Microcontrolador,» Wikipedia, 8 agosto 2021. [En línea]. Available: https://es.wikipedia.org/wiki/Microcontrolador. [Último acceso: 30 abril 2022].
- «Decodificador BCD a 7 segmentos,» Electronica Digital Circuitos, [En línea]. Available: https://sites.google.com/site/electronicadigitalmegatec/home/deccoder-bcd-a-7-segmentos.
- Vicente, «Retardos en los PIC,» Apuntes de Electrónica, [En línea]. Available: https://www.apuntesdeelectronica.com/microcontroladores/retardos-en-los-pic.htm.
- Wikipedia, «Conversor de señal analógica a digital,» Wikimedia, Inc., 31 julio 2021. [En línea]. Available: https://es.wikipedia.org/wiki/Conversor_de_señal_analógica_a_digital. [Último acceso: 30 abril 2022].
- Wikipedia, «Termómetro,» Wikimedia, 26 octubre 2021. [En línea]. Available: https://es.wikipedia.org/wiki/Termómetro#Tipos_de_termómetros. [Último acceso: 30 abril 2022].