

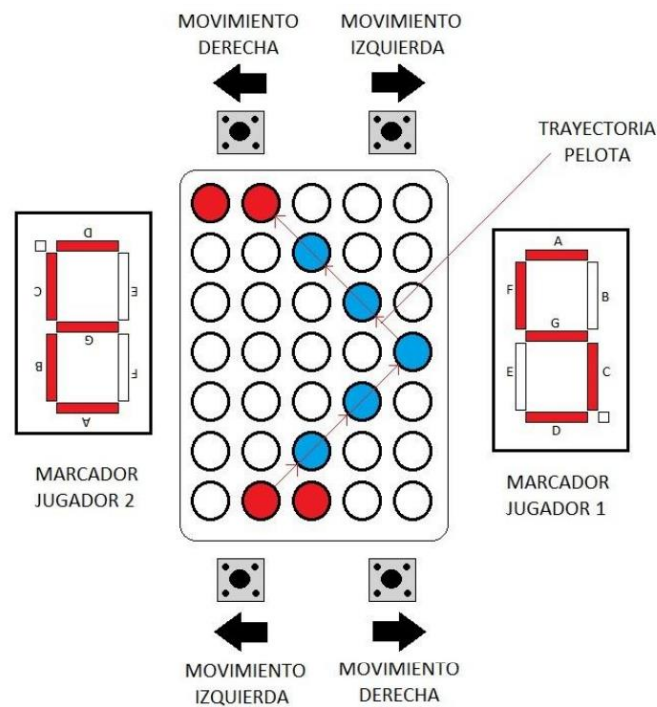


INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE CÓMPUTO

SISTEMAS EN CHIP

PROYECTO No. 3

"Juego de Ping-Pong"



ALUMNO: RENTERIA ARRIAGA JOSUE.

GRUPO: 6SCM1

PROFESOR: FERNANDO AGILAR SÁNCHEZ.

13/ENERO /2023

I. OBJETIVO GENERAL:

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

II. MATERIAL Y EQUIPO EMPLEADO

- ✓ CodeVision AVR.
- ✓ AVR Studio 4.
- ✓ Microcontrolador ATmega 8535.
- ✓ 3 Display Cátodo común.
- ✓ 8 Resistores de $330\ \Omega$ a $\frac{1}{4}\ W$.
- ✓ 1 Matriz de leds de 7x5.

III. INTRODUCCIÓN TEÓRICA

En este proyecto podremos saber utilizar correctamente y aplicar la implementación del código en el lenguaje de programación C de los conceptos de hacer un matriz de Leds de 7x5 que se pueda observar un juego de Ping Pon que con unos botones podamos mover correctamente las barras.

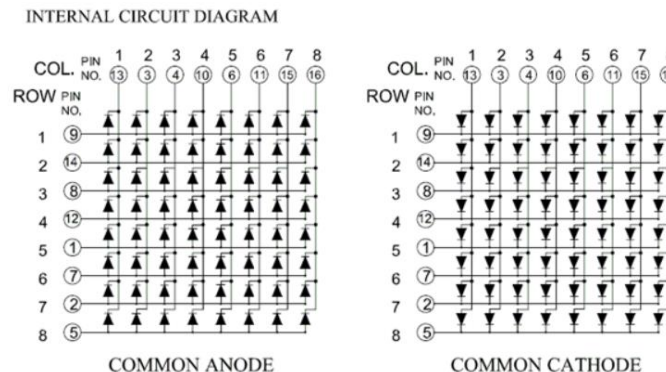
En este proyecto de estudiar la implementación de un Programa y Circuito para programar una Matriz de Leds y Display de 7 Segmentos como salidas del programa y como entradas del programa se implementarán un par de botones (derecha e izquierda). La practica se trata de la implementación del Juego de Ping Pong, donde en la Matriz de leds se vera una pelota y 2 plataformas que evitaran que dicha pelota entre cada jugador, los botones servirán para mover la plataforma de cada jugador de derecha a izquierda y el los Display se mostrara el contador de cada jugador.

En este caso del proyecto se ocuparán algunos conceptos obtenidos anteriormente en el curso como lo es la implementación de código en C para poder programar nuestro microcontrolador, algunos conceptos de Fundamentos de Diseño Digital (Display de 7 Segmentos), implementación de una Matriz de Leds, concepto de un convertidor analógico a Digital que se vio en Arquitectura y de Electrónica Analógica para poder realizar correctamente el armado de nuestro circuito y que este funcione correctamente.

Ocuparemos algunos elementos como lo dice el formato del proyecto ya que ocuparemos el simulador Code Vision para poder implementar el código para después programar nuestro ATmega8535, en mi caso se ocupará el programa Khazama AVR con su programador para poder programar nuestro ATmega con el archivo .hex que se generó anteriormente. Para la parte del armado del circuito será algo sencillo ya que solo se implementará parte de los conceptos de las anteriores practicas y solo se unirán para hacer este proyecto.

Algunos conceptos que se necesitan para poder entender algunos conceptos de la practica son los siguientes:

Una matriz de LED es un conjunto de LED cuyos agrupados en filas (o renglones) y columnas, las cuales pueden ser tan grandes como un quiera o requiera. En la siguiente ilustración se muestra el esquema de conexión de una matriz de 8 x 8 de un solo color.



En la ilustración izquierda se muestra que en las filas todos los ánodos comparten un solo nodo mientras que la imagen de la derecha las filas de LED los cátodos están conectados a un solo nodo, de ahí que las matrices tengan la configuración de ánodo común o de cátodo común. Por consiguiente, en el caso de la matriz de ánodo común, los cátodos de los LED de cada columna están conectados a un sólo nodo y, en la matriz de cátodo común, los ánodos de los LED de cada columna están conectados a un nodo.

Para que funcione, además de la matriz de LED, es muy importante que se encuentren perfectamente acoplados las partes que hacen que el sistema funcione como una sola entidad: el hardware y las líneas de código del programa.

Este proyecto nos ayudara a entender cómo se pueden implementar los circuitos con un microcontrolador y todo el proceso que conlleva realizar la programación del microcontrolador.

Con esta introducción se podrá comprender y realizar el desarrollo del proyecto que se desarrolla más adelante.

IV. 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.

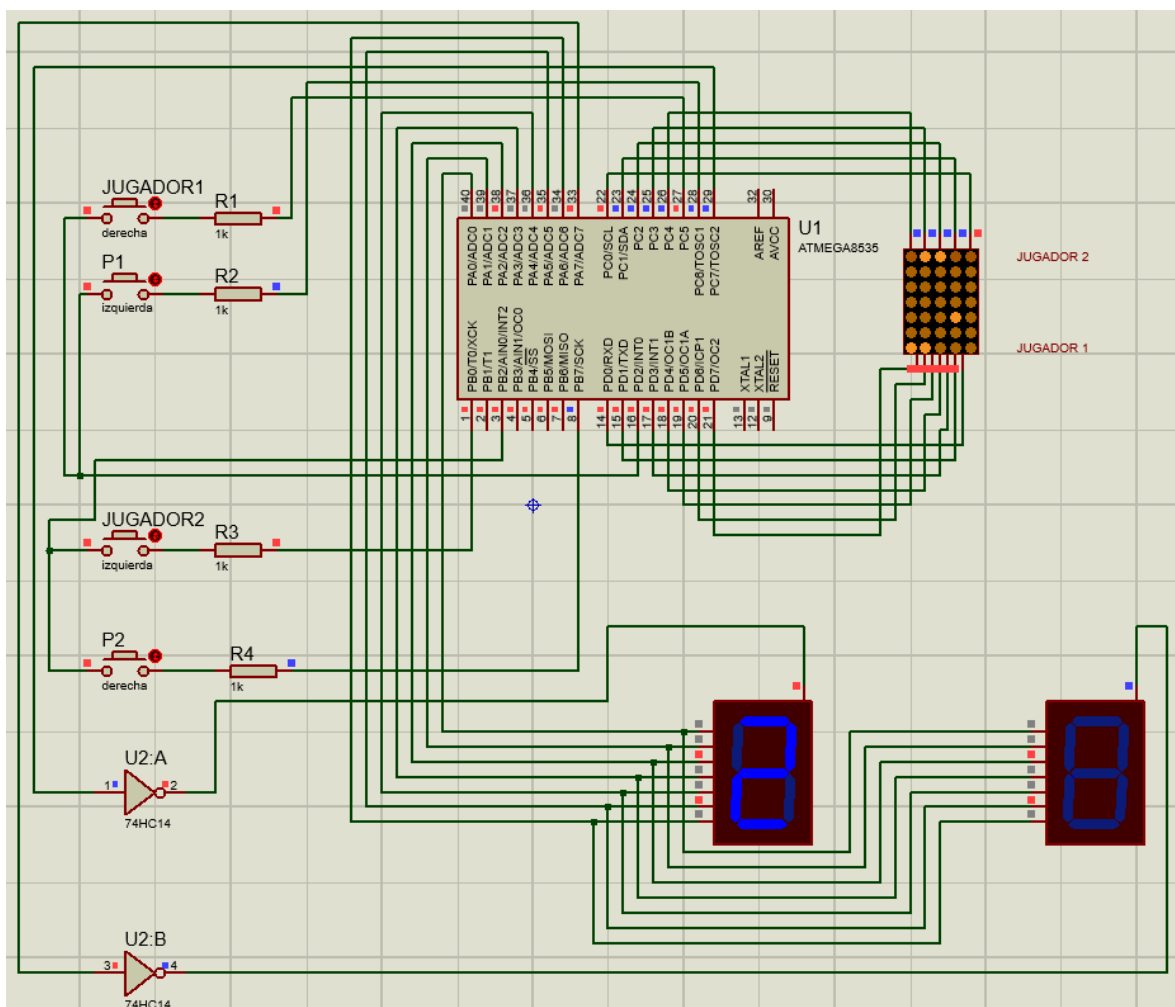


Fig.1_Circuito Armado y Simulado en Proteus.

CIRCUITO EN FISICO ARMADO.

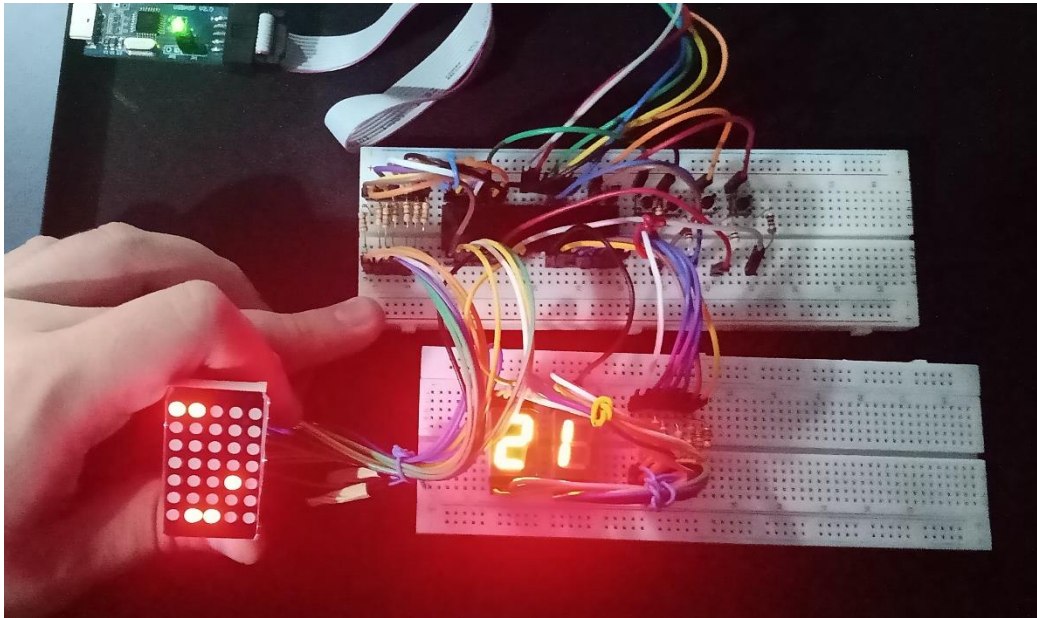


Fig.2_Circuito Armado en Físico (Juego Ping Pong).

V. CÓDIGO

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

eeprom short random;
eeprom short barral;
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(barral==2 || barral==3){
                    direccion=0;

                }else{
                    direccion=3;
                }
            }else if(y<=4){
                direccion=3;
            }
        }else{
            if(y==5) {
                if(barral==1) {
                    if(x==1) {
                        direccion=1;
                    }
                }else if(barral==2) {
                    if(x==2) {
                        direccion=1;
                    }
                    if(x==1) {
                        direccion=0;
                    }
                }else if (barral==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(barral==0 || barral==1){
                    direccion=1;

                }else{
                    direccion=2;
                }
            }else if(y<=4) {
                direccion=2;
            }
        }
    }
}

```

```

    }
}else{
    if(y==5) {
        if(barral==0) {
            if(x==1) {
                direccion=0;
            }
            if(x==2) {
                direccion=1;
            }
        }else if(barral==1) {
            if(x==2) {
                direccion=0;
            }
            if(x==3) {
                direccion=1;
            }
        }else if(barral==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);
}
//
//
}
}
```

VI. OBSERVACIONES Y CONCLUSIONES

Gracias a la realización del proyecto podemos observar cómo se utiliza e implementa un circuito con un microcontrolador (ATMega8535) y como gracias a su programador pudimos logra programar el integrado con nuestro código en lenguaje C. En este proyecto al momento de armar el circuito y simularlo pudimos observar que hay una forma de colocar correctamente la Matriz de Leds, cabe destacar también que en el caso de las Matrices de Leds también hay de Ánodo y Cátodo por lo cual al no saber esto me costo demasiado hacer al inicio mi práctica, cuando descubrí esto supe que tenia una Matriz diferente a lo que estaba programando.

Tras la investigación y explicación del proyecto previa me quedo un poco más claro como poder implementar una Matriz de Leds 7x5 y programar el ATMega8535. También al observar el data set de la Matriz se comprendido mejor como conectarlo en físico este componente.

Para concluir este proyecto podemos decir que fue una experiencia interesante ya que se tuvo que hacer la parte de la Matriz y si me costó mucho hacerlo y electrónica ya había estudiado eso, pero, nunca realizado en físico. Gracias a la anterior practica de la Matriz de Leds de 7x5 ya fue más fácil de hacerlo, ya que es casi lo mismo que esta.

VII. BIBLIOGRAFÍA

- ✓ CodeVisionAVR. (2018). CodeVisionAVR. 20/Septiembre/2022, de CodeVisionAVR Sitio web: <http://www.hpinfotech.ro/cvavr-download.html>
- ✓ Abraham Camarillo. (2017). ¿Cómo funcionan los teclados matriciales y matrices de LEDs?, 14/Enero/2023, de 330 Ohms, Sitio web: <https://blog.330ohms.com/2017/09/27/como-funcionan-los-teclados-matriciales-y-matrices-de-leds/>