

# INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Cómputo

## **Introducción a los microcontroladores**

**Profesor:** Fernando Aguilar Sánchez

**Grupo:** 3CM17

**Proyecto 3:** Ping pong

Olmedo Ramírez Esteban Raymundo – 2019630134

**Fecha:** 15/12/2021

# 1. Objetivo

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

## 2. Introducción

### 2.1. Microcontrolador

Un microcontrolador es un circuito integrado compacto diseñado para manejar una operación específica en un sistema embebido. Un microcontrolador típico incluye un procesador, memoria y periféricos de entrada / salida (E / S) en un solo chip. A veces denominados controlador integrado o unidad de microcontrolador, los microcontroladores se encuentran en vehículos, robots, máquinas de oficina, dispositivos médicos, transceptores de radio móviles, máquinas expendedoras y electrodomésticos, entre otros dispositivos. Son esencialmente computadoras personales en miniatura simples diseñadas para controlar pequeñas funciones de un componente más grande, sin un complejo sistema operativo (SO) de interfaz de usuario.

## 3. Circuito

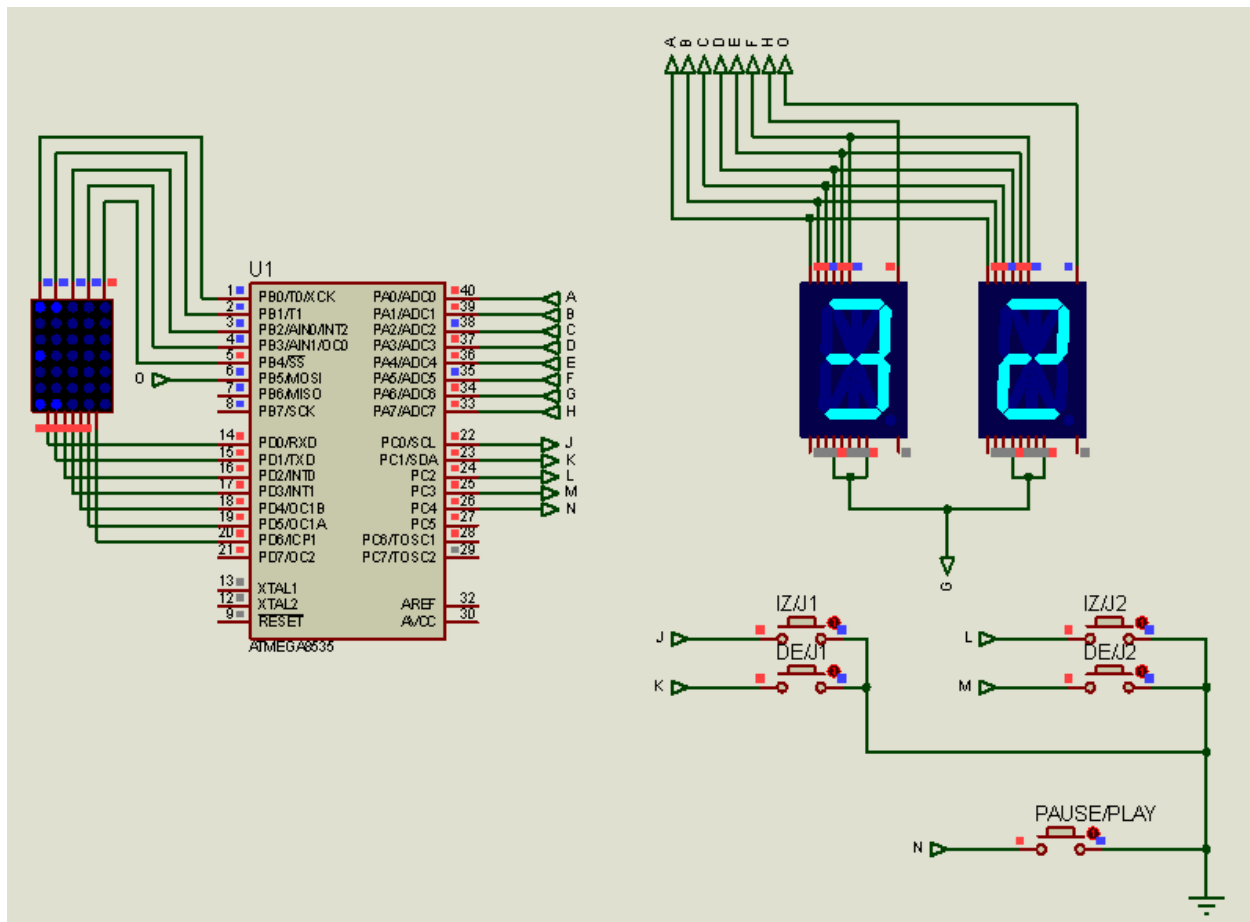


Figura 1: Circuito realizado

## 4. Código

```
/******  
This program was created by the CodeWizardAVR V3.45  
Automatic Program Generator  
Copyright 1998-2021 Pavel Haiduc, HP InfoTech S.R.L.  
http://www.hpinfotech.ro  
  
Project :  
Version :  
Date : 14/12/2021  
Author :  
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>  
#define H 7  
#define W 5  
  
struct Pelota {  
    int pos_x, pos_y;  
    int vel_x, vel_y;  
};  
  
struct Jugador {  
    int pos_x;  
};  
  
struct Jugador jugador1, jugador2;  
struct Pelota pelota;  
  
int estado = 2; // 0 no a comenzado el juego, 1 jugando, 2 termino  
const char display[10] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x6f};  
bit ul_izj2, au_izj2, ul_dej2, au_dej2, ul_izj1, au_izj1, ul_dej1, au_dej1, au_pa, ul_pa;  
int score_j1 = 0, score_j2 = 0;  
  
void mover_derecha(int jugador) {  
    if(jugador == 1) {  
        if(jugador1.pos_x != 4) {  
            jugador1.pos_x++;  
        }  
    }  
    else {  
        if(jugador2.pos_x != 4) {  
            jugador2.pos_x++;  
        }  
    }  
}  
  
void mover_izquierda(int jugador) {  
    if(jugador == 1) {  
        if(jugador1.pos_x != -1) {  
            jugador1.pos_x--;  
        }  
    }  
    else {  
        if(jugador2.pos_x != -1) {  
            jugador2.pos_x--;  
        }  
    }  
}
```

```

}
void inicializar() {
    score_j1 = 0;
    score_j2 = 0;
    pelota.pos_x = 2;
    pelota.pos_y = 3;
    jugador1.pos_x = 0;
    jugador2.pos_x = 0;
    pelota.vel_x = 1;
    pelota.vel_y = 1;
}

void actualizar_pelota() {
    pelota.pos_x += pelota.vel_x;
    pelota.pos_y += pelota.vel_y;
    if(pelota.pos_y == 0 || pelota.pos_y == 6) {
        pelota.vel_y *= -1;
    }
    if(pelota.pos_x == 0 || pelota.pos_x == 4) {
        pelota.vel_x *= -1;
    }
}

void actualizar_score() {
    if(pelota.pos_y == 0) {
        if(jugador1.pos_x != pelota.pos_x && jugador1.pos_x + 1 != pelota.pos_x) {
            score_j2 = score_j2 + 1;
            if(score_j2 == 9)
                estado = 2;
        }
    }
    if(pelota.pos_y == 6) {
        if(jugador2.pos_x != pelota.pos_x && jugador2.pos_x + 1 != pelota.pos_x) {
            score_j1 = score_j1 + 1;
            if(score_j1 == 9)
                estado = 2;
        }
    }
}

void imprimir_score(int score, int num_display) {
    if(num_display == 2) {
        PORTB = PORTB & (~0x20);
        PORTA = display[score] | 0x80;
        // PORTA = display[7];
        delay_ms(5);
        PORTB = PORTB | 0x20;
    }
    else {
        PORTA = display[score];
        // PORTA = display[4];
        PORTB = PORTB | 0x20;
        delay_ms(5);
    }
}

void checar_movimiento_jugador1() {
    au_izj1 = PINC.0;
    au_dej1 = PINC.1;
    if(au_izj1 == 0 && ul_izj1 == 1) {
        mover_izquierda(1);
    }
    if(au_dej1 == 0 && ul_dej1 == 1) {
        mover_derecha(1);
    }
    ul_izj1 = au_izj1;
    ul_dej1 = au_dej1;
}

void checar_movimiento_jugador2() {

```

```

    au_izj2 = PINC.2;
    au_dej2 = PINC.3;
    if (au_izj2 == 0 && ul_izj2 == 1) {
        mover_izquierda(2);
    }
    if (au_dej2 == 0 && ul_dej2 == 1) {
        mover_derecha(2);
    }
    ul_izj2 = au_izj2;
    ul_dej2 = au_dej2;
}

void checar_play() {
    au_pa = PINC.4;
    if (au_pa == 0 && ul_pa == 1) {
        if (estado != 1) {
            if (estado == 2)
                inicializar();
            estado = 1;
        }
        else {
            estado = 0;
        }
    }
    ul_pa = au_pa;
}

void checar_botones() {
    checar_movimiento_jugador1();
    checar_movimiento_jugador2();
    checar_play();
}

void imprimir_pantalla() {
    int i = 0;
    char msk = 0;
    for (i = 0; i < 5; i++) {
        msk = 0xFF;
        if (pelota.pos_x == i) {
            msk &= ~(1 << pelota.pos_y);
        }
        if (jugador1.pos_x == i || jugador1.pos_x == (i-1)) {
            msk &= ~(1 << 0);
        }
        if (jugador2.pos_x == i || jugador2.pos_x == (i-1)) {
            msk &= ~(1 << 6);
        }
        PORTB = (1 << i) | 0x20;
        PORTD = msk;
        imprimir_score(score_j1, 1);
        imprimir_score(score_j2, 2);
    }
}

void imprimir_ganador(int ganador) {
    unsigned char val[3][5] = {
        {0x41, 0x2e, 0x36, 0x3a, 0x41},
        {0x3f, 0x3d, 0x00, 0x3f, 0x3f},
        {0x3d, 0x1e, 0x2e, 0x36, 0x39}
    };
    int i = 0;
    for (i = 0; i < 5; i++) {
        PORTB = (1 << i) | 0x20;
        PORTD = val[ganador][i];
        imprimir_score(score_j1, 1);
        imprimir_score(score_j2, 2);
    }
}

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

```

```

// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2) | (1<<DDA1) | (1<<DDA0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRB=(1<<ddb7) | (1<<ddb6) | (1<<ddb5) | (1<<ddb4) | (1<<ddb3) | (1<<ddb2) | (1<<ddb1) | (1<<ddb0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

// Port C initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
// State: Bit7=T Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
PORTC=(0<<PORTC7) | (1<<PORTC6) | (1<<PORTC5) | (1<<PORTC4) | (1<<PORTC3) | (1<<PORTC2) | (1<<PORTC1) | (1<<PORTC0);

// Port D initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRD=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (1<<DDD3) | (1<<DDD2) | (1<<DDD1) | (1<<DDD0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OCO output: Disconnected
TCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Disconnected
// OC1B output: Disconnected
// 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
TCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
TCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
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=0<<AS2;
TCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

```

```

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization
// INTO: Off
// INT1: Off
// INT2: Off
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
MCUCSR=(0<<ISC2);

// USART initialization
// USART disabled
UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AINO pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACISO);
SFIOR=(0<<ACME);

// ADC initialization
// ADC disabled
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

while (1)
{
    // Place your code here
    inicializar();
    while(1) {
        imprimir_score(score_j1, 1);
        imprimir_score(score_j2, 2);
        imprimir_pantalla();
        if(estado == 1) {
            actualizar_pelota();
            actualizar_score();
        }
        else if(estado == 2) {
            if(score_j1 == 9) {
                imprimir_ganador(1);
            }
            else if(score_j2 == 9) {
                imprimir_ganador(2);
            }
            else {
                imprimir_ganador(0);
            }
        }
        checar_botones();
    }
}

```

## 5. Conclusiones

Después de haber realizado este proyecto pude integrar los conocimientos adquiridos durante el semestre en la materia de microcontroladores, demostrando uno de los usos de los mismos para poder manejar distintos

dispositivos con un fin específico, en este caso crear un pequeño juego de ping pong.