



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE COMPUTO



PRÁCTICA 3

CONVERTIDOR BCD A 7 SEGMENTOS" Y "CONTADOR HEXADECIMAL

SISTEMAS EN CHIP
6CM3

PROFESOR:
AGUILAR SÁNCHEZ FERNANDO

INTEGRANTES:
MIGUEL VÁSQUEZ MARIANO JOSUÉ
FRANCO OLVERA ODER
GONZALEZ RAMOS ANGEL DAVID

Fecha de Elaboración
06/04/2024

INTRODUCCIÓN

Convertidor

Es un elemento digital que funciona a base de estados lógicos, con los cuales indica una salida determinada basándose en un dato de entrada característico, su función operacional se basa en la introducción a sus entradas de un número en código binario correspondiente a su equivalente en decimal para mostrar en los siete pines de salida establecidos para el integrado, una serie de estados lógicos que están diseñados para conectarse a un elemento alfanumérico en el que se visualizará el número introducido en las entradas del decodificador. El elemento alfanumérico que se conecta a las siete salidas del decodificador también está diseñado para trabajar con estados lógicos, es un dispositivo elaborado con un arreglo de LED de tal manera que muestre los números decimales desde el cero hasta el nueve dependiendo del dato recibido desde el decodificador, a este elemento se le conoce con el nombre de display ó dispositivo alfanumérico de 7 segmentos.

Display de 7 segmentos

Es un dispositivo alfanumérico formado por diodos emisores de luz (LED), posicionados para formar un número ocho, a cada uno se les llama segmentos. Encendiendo algunos y apagando otros se pueden formar diferentes números mediante combinaciones entre ellos.

Cada segmento esta designado con una letra. El punto decimal se denomina dp, pt ó simplemente P. El display se encuentra en una representación de encapsulado con los pines para conectarlo a un circuito. A cada pin o pata del encapsulado se le asigna la letra correspondiente del segmento. Esto significa que, por ejemplo, con el pin "a" podemos controlar el estado del segmento "a"(encenderlo o apagarlo).

Materiales y Equipo empleado

- CodeVision AVR
- AVR Studio 4
- Microcontrolador ATmega 8535
- 1 display ánodo común
- 1 display cátodo común
- 14 resistores de 330 ohm a un cuarto de wat

CÓDIGO

```
/*****
```

This program was created by the CodeWizardAVR V4.01

Automatic Program Generator

© Copyright 1998-2024 Pavel Haiduc, HP InfoTech S.R.L.

<http://www.hpinfotech.ro>

Project :

Version :

Date : 24/02/2024

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

```
*****/
```

```
// I/O Registers definitions
```

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

```
unsigned char variable1, variable2;
```

```
const char tabla7segmentos[16] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07,  
0x7F, 0x6F, 0x77, 0x7C, 0x39, 0x5E, 0x79, 0x71};
```

```
// Declare your global variables here
```

```
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=T Bit5=T Bit4=T Bit3=P Bit2=P Bit1=P Bit0=P
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (1<<PORTC3) |
(1<<PORTC2) | (1<<PORTC1) | (1<<PORTC0);

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

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=(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
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) |
(0<<WGM10);
TCCR1B=(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;
TCCR2=(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
```

```
// INT0: 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 AIN0 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<<ACIS0);
```

```
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

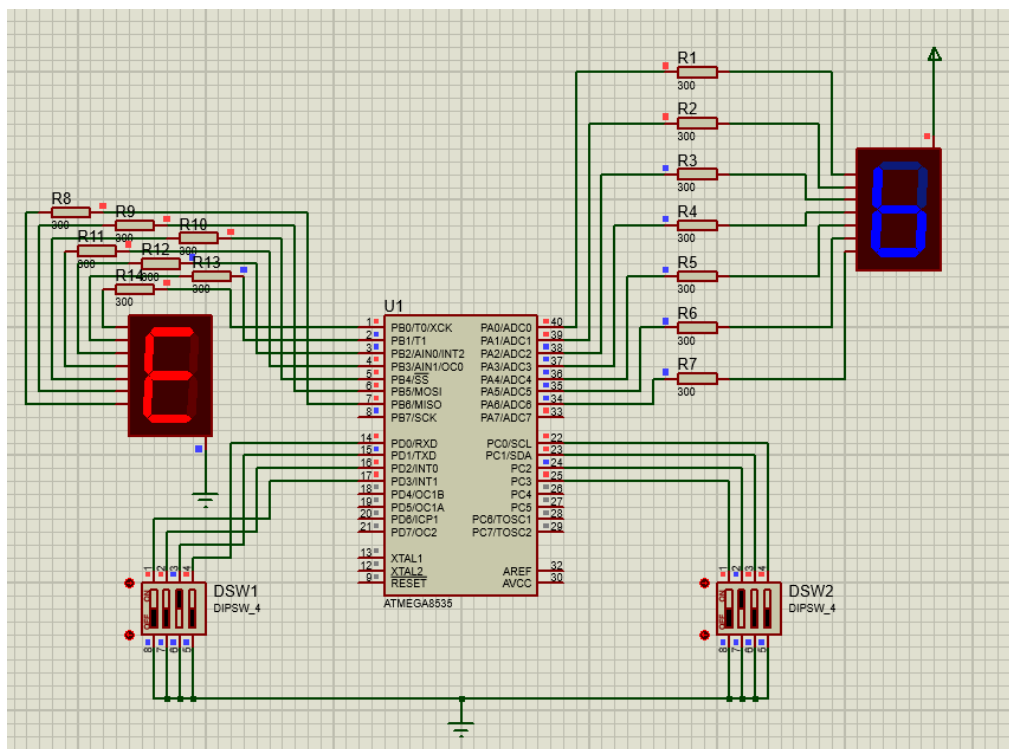
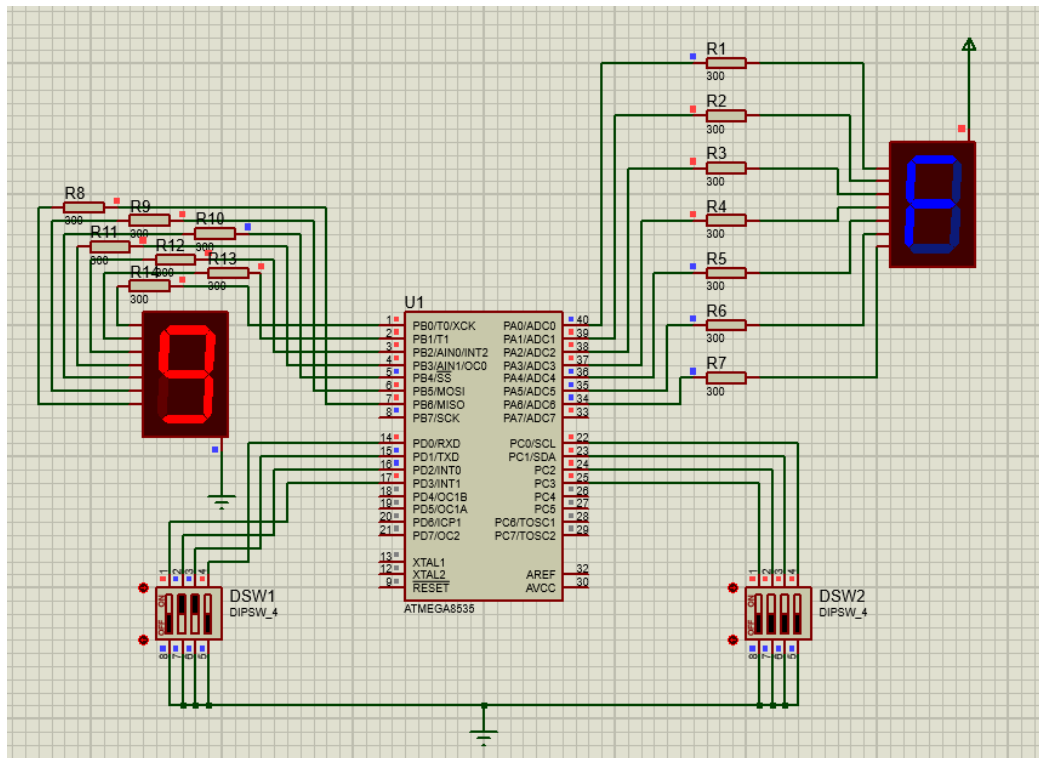
    variable1 = PIND&0x0F; // Se enmascaran los 4 bits menos significativos
para el puerto D
    variable2 = PINC&0x0F; // Se enmascaran los 4 bits menos significativos
para el puerto C

    // Se muestran los valores en decimal
    if(variable1 < 10) {
        PORTB = tabla7segmentos[variable1];
    }
    else {
        PORTB = 0x79;
    }

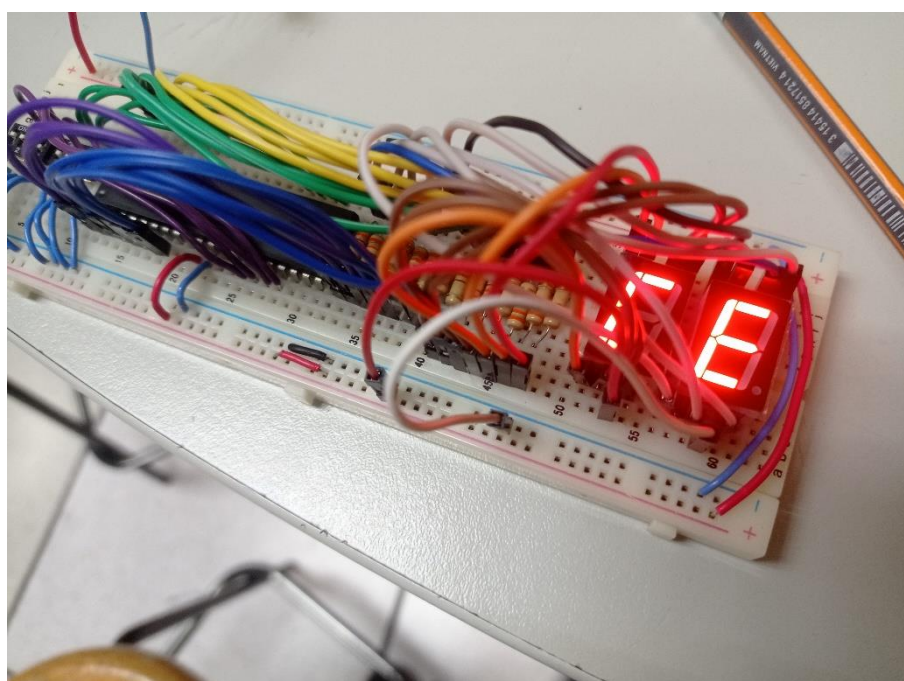
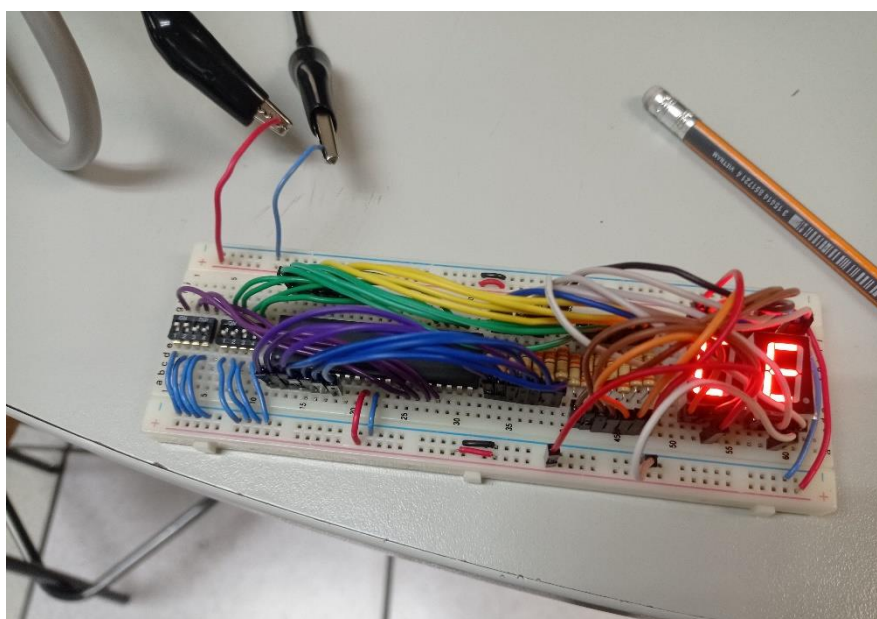
    // Se muestran los valores en hexadecimal
    if(variable2 < 16) {
        PORTA = ~tabla7segmentos[variable2];
    }
}
}

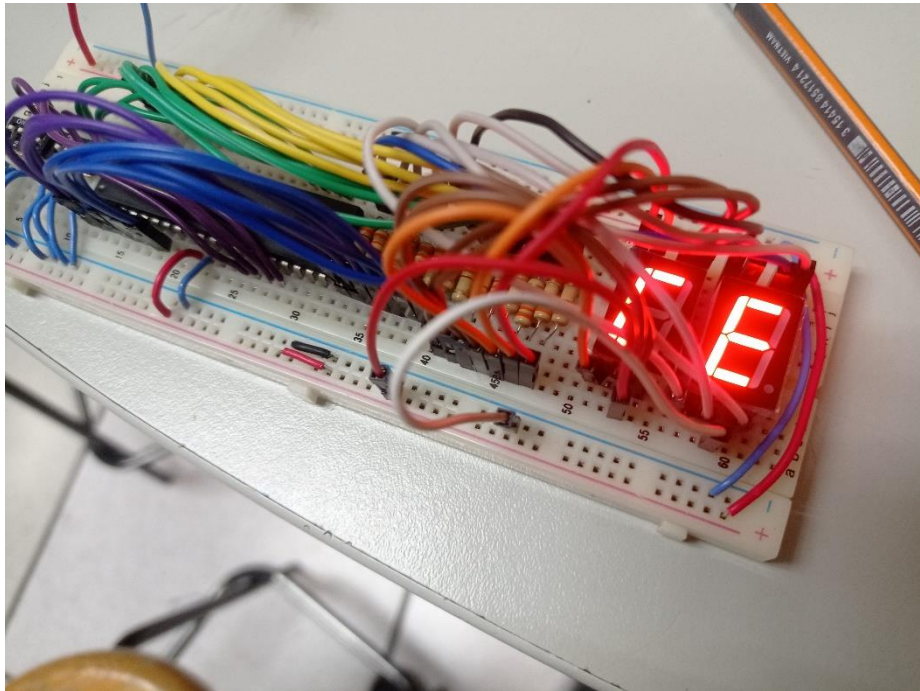
```

SIMULACIÓN EN PROTEUS



FOTOS





CONCLUSIONES INDIVIDUALES

Franco Olvera Demian Oder

Este fue el primer acercamiento que se tuvo con los Display de siete segmentos utilizando al CI ATMEGA en conjunto. Se trata de un circuito insidiosamente simple que combina la lógica necesaria para manejar los Display, duplicándola para manejar valores decimales y hexadecimales por una y otra parte, respectivamente. Un aspecto interesante de esta práctica fue el libre albedrío que se dio para poder codificar el funcionamiento de ambos Displays. Más allá de este pequeño reto, la complejidad de la práctica no fue complicada más allá de lidiar con fallos físicos del ATMEGA que se estaba trabajando.

Gonzalez Ramos Angel David

La práctica de diseñar un convertidor BCD a 7 segmentos y un contador hexadecimal en un microcontrolador como el ATmega8535 es esencial para comprender los procesos de conversión de datos y la representación de números en pantallas de siete segmentos. Esto implica el dominio de la lógica de codificación BCD y la configuración de los pines de salida para controlar los segmentos del display. Además, el desarrollo de un contador hexadecimal permite explorar la manipulación de números en base 16, así como la sincronización con pulsos de reloj internos. Estas prácticas no solo fortalecen la comprensión de conceptos fundamentales de electrónica digital, sino que

también proporcionan las habilidades necesarias para implementar soluciones prácticas en sistemas embebidos y proyectos de automatización.

Miguel Vásquez Mariano Josué

Con la práctica desarrollada implementamos un convertidor BCD a 7 segmentos haciendo uso de un arreglo en el cual se almacenan los valores necesarios para que en el display se muestre el valor equivalente. Estos valores se guardan de tal manera que el valor leído coincida con el valor del índice donde se encuentra almacenado el valor del display, de esa forma solamente utilizamos el valor leído para obtener el valor que se encuentra en esa posición y mandarlo a la salida. Así mismo, obtuvimos un código reutilizable que podrá ser usado en futuras prácticas para mostrar valores de salida en el display y así poder leerlas más fácilmente. Realizarlo fue bastante fácil gracias a la versatilidad del lenguaje C para el manejo de arreglos y variables a diferencias de lenguajes como el de VHDL.

REFERENCIAS

- Electronica Digital Circuitos - Decodificador BCD a 7 segmentos. (s.f.). <https://sites.google.com/site/electronicadigitalmegatec/introducci%C3%B3n-a-sistemas-digitales/decodificador-bcd-a-7-segmentos>