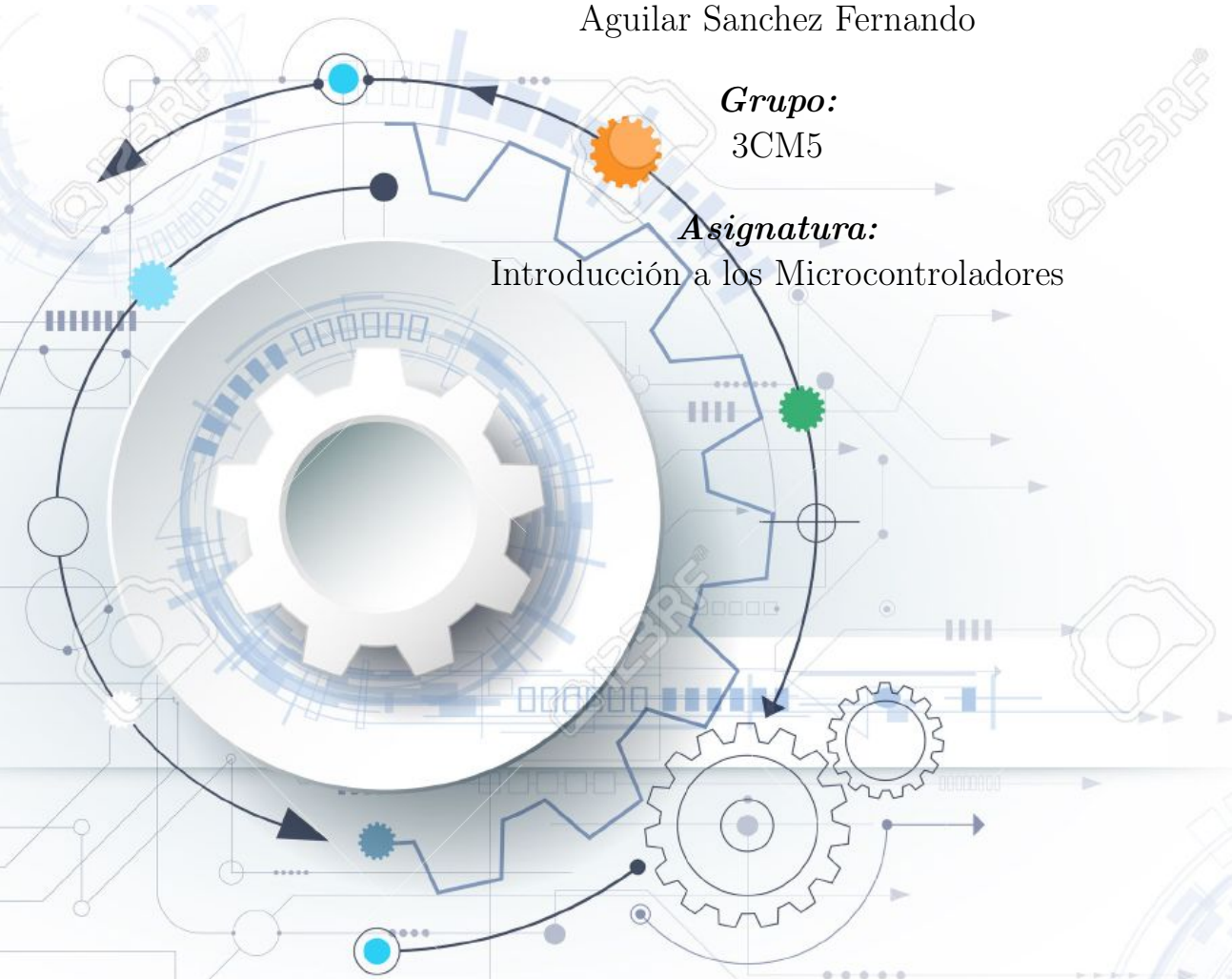

Instituto Politécnico Nacional
Escuela Superior de Cómputo

Práctica 3
Convertidor BCD a 7 Segmentos

Alumno:
Amador Nava Miguel Ángel

Profesor:
Aguilar Sanchez Fernando

Grupo:
3CM5

Asignatura:
Introducción a los Microcontroladores

Índice general

Introducción

Los codificadores tienen como función detectar la presencia de una determinada combinación de bits en sus entradas y señalar la presencia de este código mediante cierto nivel de salida.

Un decodificador posee N líneas de entrada para gestionar N bits y en una de las $\frac{2}{N}$ líneas de salida indica la presencia de una o mas combinaciones de n bits.[1]

El decodificador BCD a 7 segmentos permite representar un número BCD en un display de 7 segmentos. Un display de 7 segmentos es una matriz de LEDs (diodos emisores de luz) o también de elementos de cristal líquido, ordenados de tal manera que permiten "dibujarün número decimal. En la Figura 1 se muestra la forma de un display

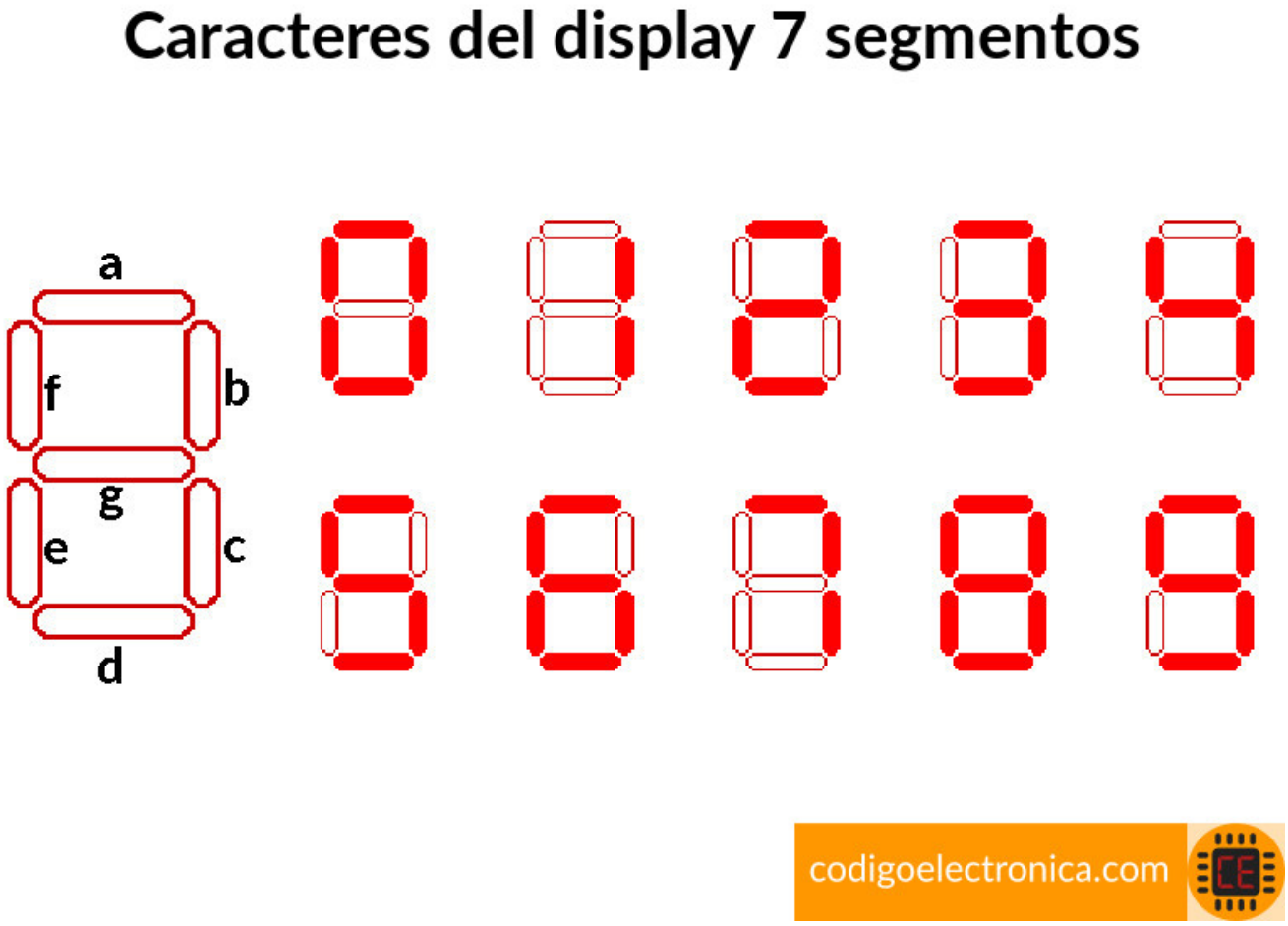

Figura 1: Caracteres del display de 7 segmentos

Cada segmento se identifica con una letra de la A a la F y justamente estas son las salidas del decodificador.

Con respecto a los displays de LEDs, los hay en dos configuraciones: cátodo común y ánodo

común. Esto se refiere a cómo se encuentran conectados internamente los leds que conforman el display. En el caso del display cátodo común, todos los cátodos de los leds están unidos y conectados a un pin del display. Análogamente, para los displays de ánodo común, todos los leds tienen unidos sus ánodos y conectados a un pin del display.[2]

Es necesario saber que entre cada segmente debe conectarse una resistencia para limitar la corriente consumida, pues corre el riesgo de dañarse el display el decodificador.

Objetivo

Al término de la sesión, los integrantes del equipo contaran con la habilidad de realizar un contador BCD empleando arreglos.

Material y equipo

- CodeVision AVR
- AVR Studio 4
- Microcontrolador ATmega 8535
- 1 Display ánodo común
- 1 Display cátodo común
- 14 Resistores de 330 Ω a $\frac{1}{4}$ W

Planteamiento del problema

Diseñar un convertidor BCD a 7 Segmentos para un Display Cátodo y Ánodo, donde cuente de 0 a 9 y valores superiores indique un error ó de 0 a 15 en hexadecimal.

Análisis teórico

En la Tabla 1 se muestras los valores en hexadecimal para display de 7 segmentos de Cátodo común. El cual se mostrará al generar una combinación en binario en un dip switch.

Número display	**.**	**g**	**f**	**e**	**d**	**c**	**b**	**a**	**Valor Hexadecimal**
0	0	1	1	1	1	1	1	1	0x3F
1	0	0	0	0	0	1	1	0	0x06
2	0	1	0	1	1	0	1	1	0x5B
3	0	1	0	0	1	1	1	1	0x4F
4	0	1	1	0	0	1	1	0	0x66
5	0	1	1	0	1	1	0	1	0x6D
6	0	1	1	1	1	1	0	1	0x7C
7	0	0	0	0	0	1	1	1	0x07
8	0	1	1	1	1	1	1	1	0x7F
9	0	1	1	0	1	1	1	1	0x6F
E	0	1	1	0	1	0	0	1	0x79

Cuadro 1: Valores para display de 7 segmentos Cátodo común

Para el display de Ánodo común podemos generar una tabla y obtener los valores negando cada bit ó en tiempo de ejecución usar una mascara para el complemento de cada número.

Para obtener el complemento se puede usar el mismo número haciendo XOR con 0xff.

A	**B**	**A XOR B**
0	0	0
0	1	1
1	0	1
1	1	0

Cuadro 2: Puerta lógica XOR

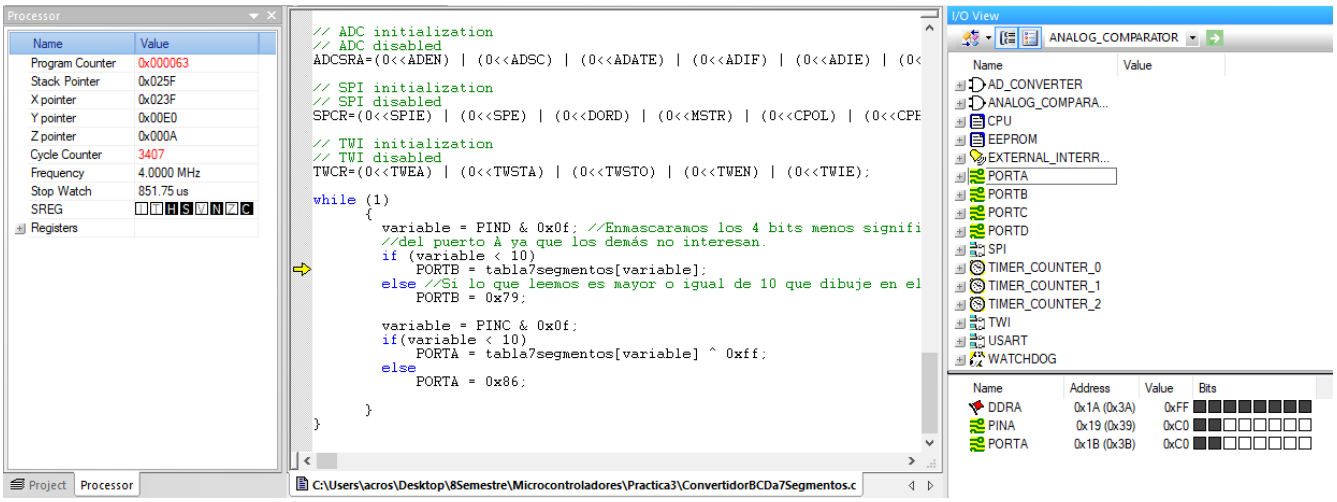

Figura 2: Valor 0 de inicialización para puerto A

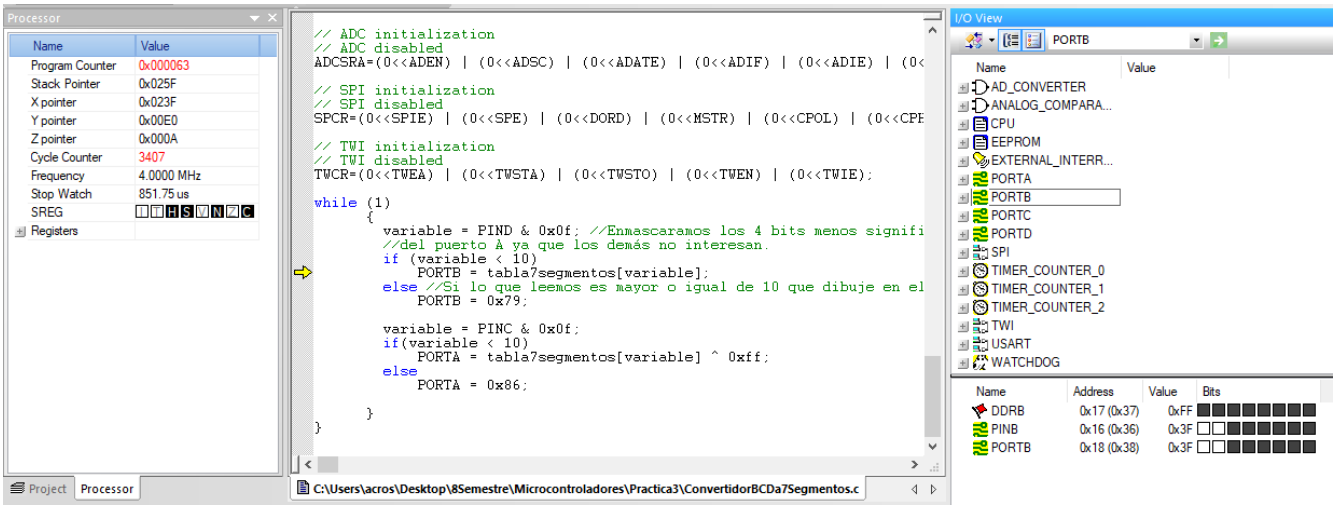

Figura 3: Valor 0 de inicialización para puerto B

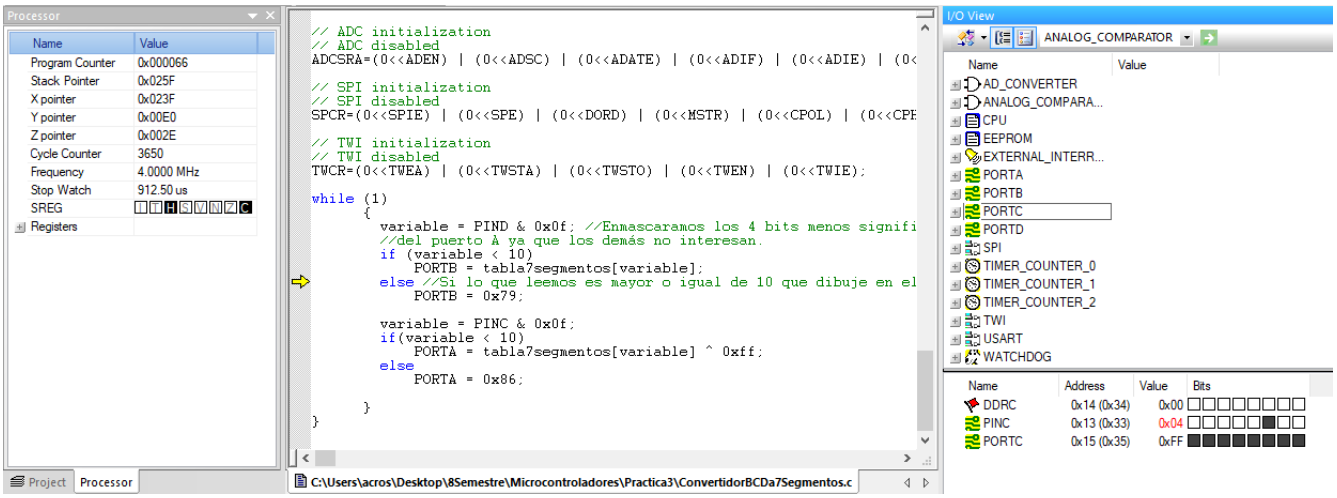

Figura 4: Ingreso de valor 4 al PINC

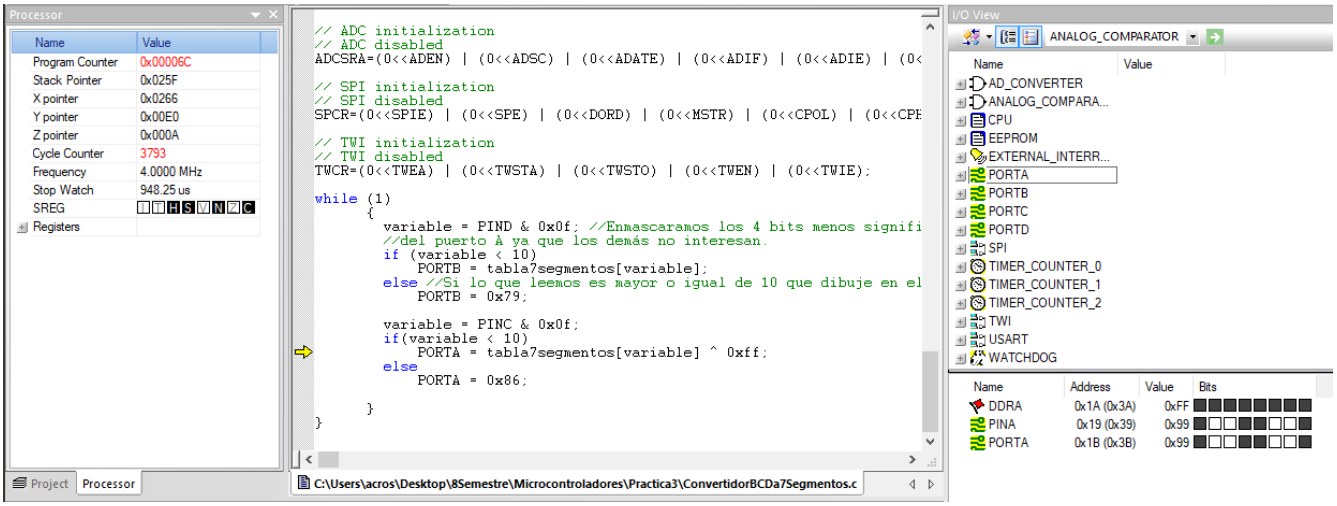

Figura 5: Salida del valor 4 al puerto A

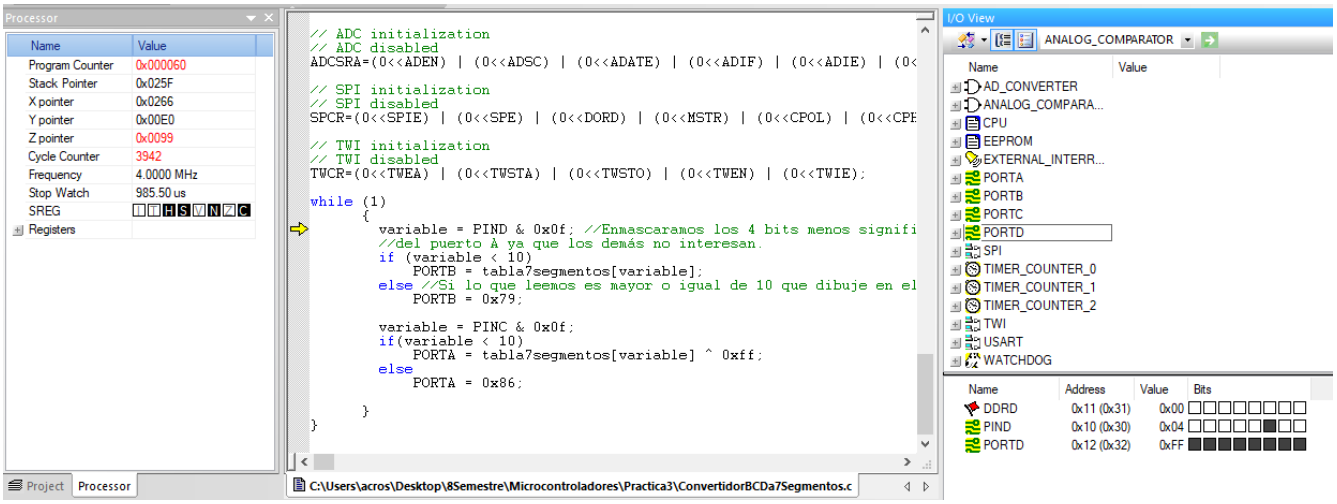

Figura 6: Ingreso de valor 4 al PIND

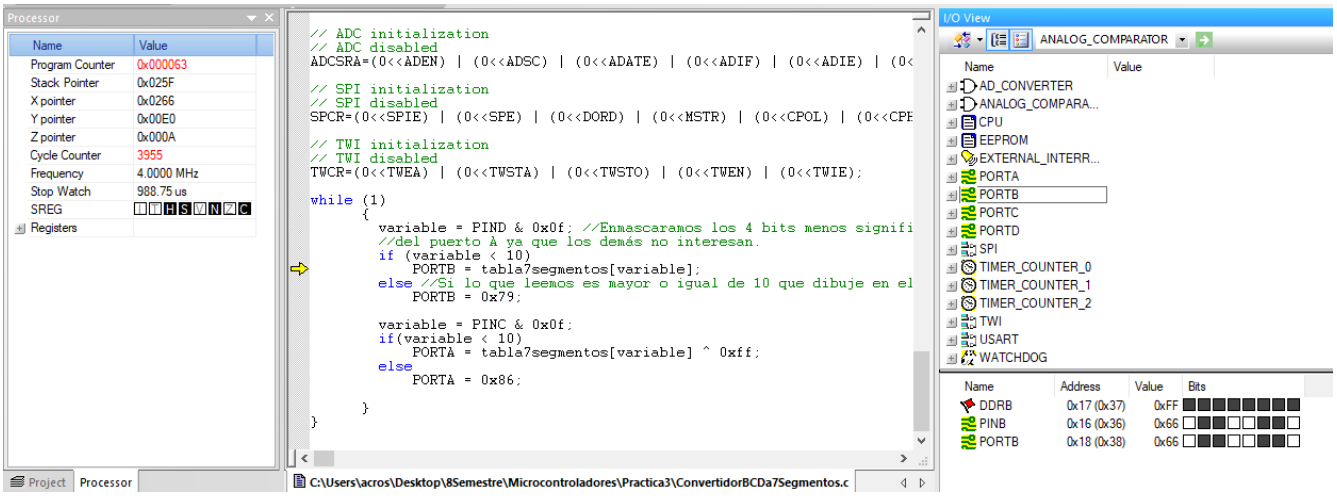

Figura 7: Salida del valor 4 al puerto B

Diseño del circuito

En la Figura 8 se muestra el circuito para la implementación de la práctica.

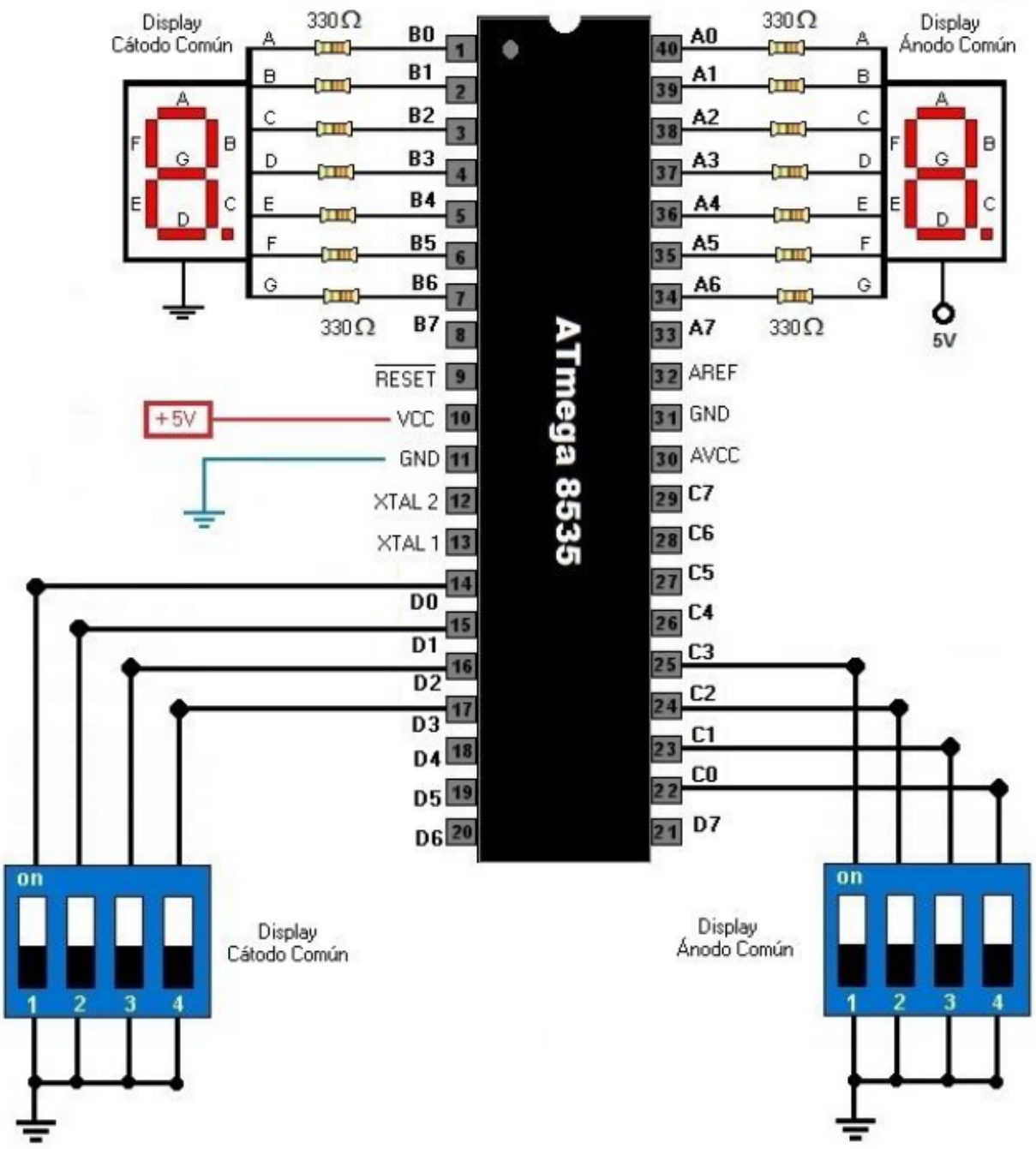

Figura 8: Circuito para el convertidor BCD a 7 Segmentos con los displays ánodo y cátodo común

Análisis práctico

En la Figura 9 se puede ver el circuito implementado, siendo alimentado con el grabador de AVR's, el display de Ánodo muestra la combiandción dle número 7 y el display de Cátodo la combinación del número 5.

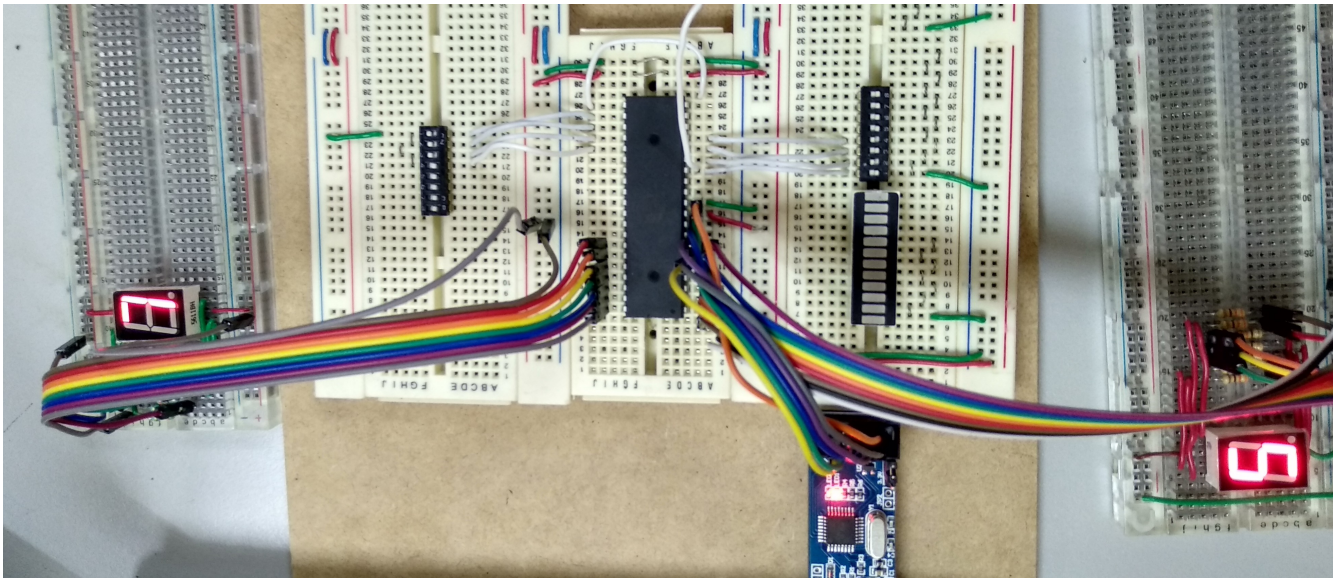

Figura 9: Circuito de displays de 7 segmentos

Comparaciones

No es un circuito complejo lo cual no permite variaciones de lo teórico a lo práctico.

Conclusiones

El lenguaje de maquina sólo son ceros y unos pero aún decir esto es de manera didactica pues realmente es la abstracción de si hay voltaje o no lo hay. El procesador solo interpreta tramas de voltaje, de manera analoga sólo devuelve voltaje el microcontrolador y es necesario que el desarrollador encapsule para que los usuarios pueden usar de manera simple e intuitiva lo implemnetado. En este caso recibe en binario el el microcontrollador el número y utiliza sus puertos como interfaz para mostrar ese número y pueda ser interpretado por cualquier persona.

Código en lenguaje C

```
/*****************************************************
This program was created by the CodeWizardAVR V3.34
Automatic Program Generator
 Copyright 1998-2018 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project : Introduccin a los Microcontroladores
Version : 1.0
Date    : 29/01/2019
Author  : Amador Nava Miguel ngel
Company : Escuela Superior de Cmputo
Comments: Prctica 3 Convertidor BCD a 7 Segmentos


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>

// Declare your global variables here
unsigned char variable;
const char tabla7segmentos [10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x

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
DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2)
// 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) |

// Port B initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out
```

```
42 DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2)
43 // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
44 PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) |
45
46 // Port C initialization
47 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=I
48 DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2)
49 // State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
50 PORTC=(1<<PORTC7) | (1<<PORTC6) | (1<<PORTC5) | (1<<PORTC4) | (1<<PORTC3) |
51
52 // Port D initialization
53 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=I
54 DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2)
55 // State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
56 PORTD=(1<<PORTD7) | (1<<PORTD6) | (1<<PORTD5) | (1<<PORTD4) | (1<<PORTD3) |
57
58 // Timer/Counter 0 initialization
59 // Clock source: System Clock
60 // Clock value: Timer 0 Stopped
61 // Mode: Normal top=0xFF
62 // OC0 output: Disconnected
63 TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<C
64 TCNT0=0x00;
65 OCR0=0x00;
66
67 // Timer/Counter 1 initialization
68 // Clock source: System Clock
69 // Clock value: Timer1 Stopped
70 // Mode: Normal top=0xFFFF
71 // OC1A output: Disconnected
72 // OC1B output: Disconnected
73 // Noise Canceler: Off
74 // Input Capture on Falling Edge
75 // Timer1 Overflow Interrupt: Off
76 // Input Capture Interrupt: Off
77 // Compare A Match Interrupt: Off
78 // Compare B Match Interrupt: Off
79 TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) |
80 TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<
81 TCNT1H=0x00;
82 TCNT1L=0x00;
83 ICR1H=0x00;
84 ICR1L=0x00;
85 OCR1AH=0x00;
```

```
86  OCR1AL=0x00;
87  OCR1BH=0x00;
88  OCR1BL=0x00;
89
90  // Timer/Counter 2 initialization
91  // Clock source: System Clock
92  // Clock value: Timer2 Stopped
93  // Mode: Normal top=0xFF
94  // OC2 output: Disconnected
95  ASSR=0<<AS2;
96  TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS
97  TCNT2=0x00;
98  OCR2=0x00;
99
100 // Timer(s)/Counter(s) Interrupt(s) initialization
101 TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0
102
103 // External Interrupt(s) initialization
104 // INT0: Off
105 // INT1: Off
106 // INT2: Off
107 MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
108 MCUCSR=(0<<ISC2);
109
110 // USART initialization
111 // USART disabled
112 UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCS
113
114 // Analog Comparator initialization
115 // Analog Comparator: Off
116 // The Analog Comparator's positive input is
117 // connected to the AIN0 pin
118 // The Analog Comparator's negative input is
119 // connected to the AIN1 pin
120 ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0
121 SFIOR=(0<<ACME);
122
123 // ADC initialization
124 // ADC disabled
125 ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS
126
127 // SPI initialization
128 // SPI disabled
129 SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) |
```

```
130
131 // TWI initialization
132 // TWI disabled
133 TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
134
135 while (1)
136       {
137         variable = PIND & 0x0f; //Enmascaramos los 4 bits menos significati
138         //del puerto A ya que los dems no interesan.
139         if (variable < 10)
140             PORTB = tabla7segmentos[variable];
141         else //Si lo que leemos es mayor o igual de 10 que dibuje en el dis
142             PORTB = 0x79;
143
144         variable = PINC & 0x0f;
145         if(variable < 10)
146             PORTA = tabla7segmentos[variable] ^ 0xff;
147         else
148             PORTA = 0x86;
149
150       }
151 }
```

Bibliografía

[1] Decodificadores. [Online]. Available: http://personales.unican.es/manzanom/Planantiguo/EDigitalI/DECG6.pdf

[2] Subsistemas digitales. UTN. [Online]. Available: http://frrq.cvg.utn.edu.ar/pluginfile.php/9323/mod_resource/content/1/04-%20Subsistemas%20digitales.pdf