Práctica #3 BCD 7 Segmentos
Introducción a los Microcontroladores

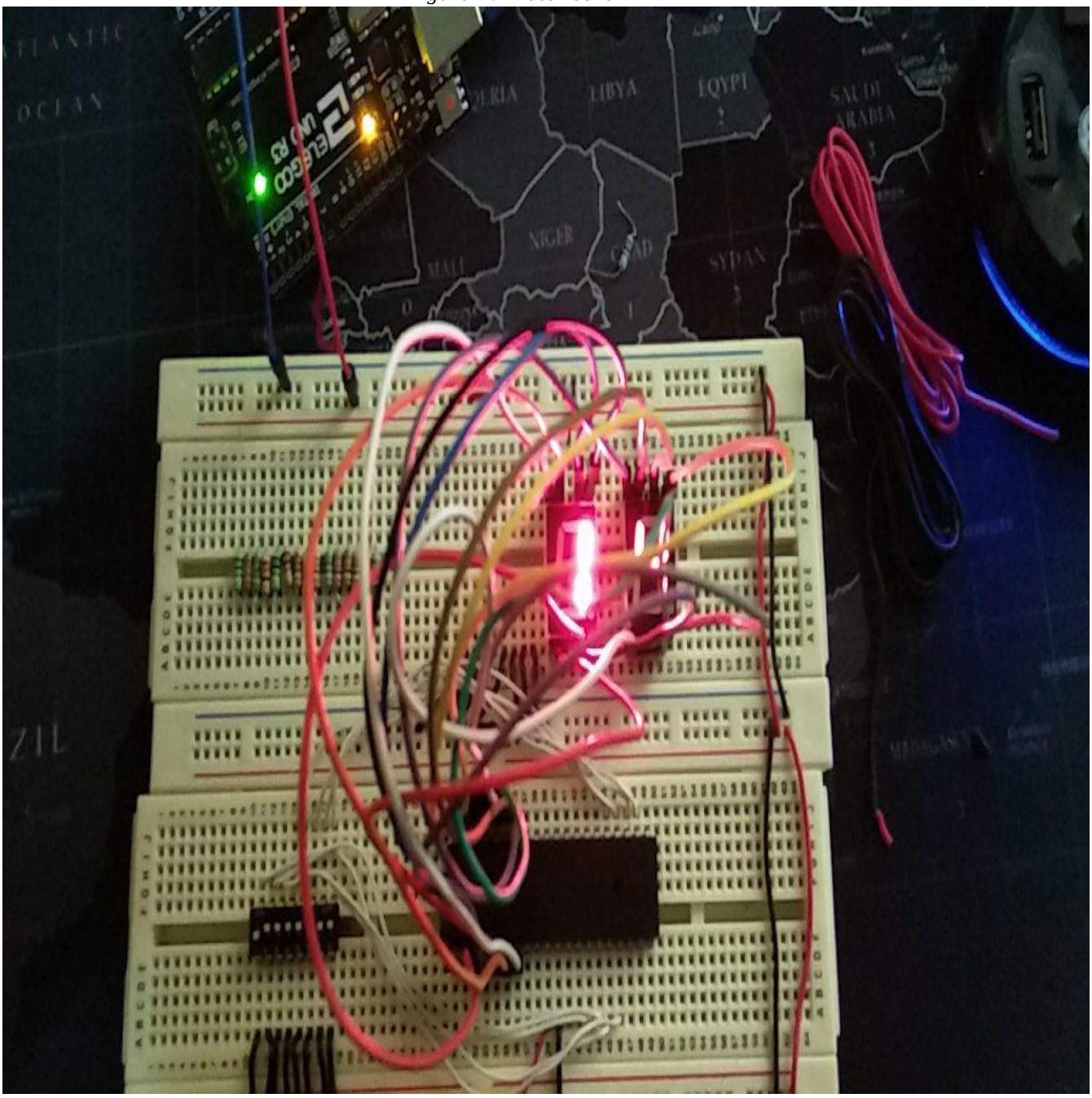
Fernando Aguilar Sánchez

Jorge Gómez Reus

Índice

1. Protoboard

Figura 1: ProtoBoard



2. Código

```
#include <mega8535.h>


void main(void)
{
unsigned char variable;
unsigned char variableA;
const char tabla7segmentos [10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x6f};

// 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=T Bit2=T Bit1=T Bit0=P
PORTC=(1<<PORTC7) | (1<<PORTC6) | (1<<PORTC5) | (1<<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=T Bit2=T Bit1=T Bit0=P
PORTD=(1<<PORTD7) | (1<<PORTD6) | (1<<PORTD5) | (1<<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
```

```
56 TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
57 TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
58 TCNT1H=0x00;
59 TCNT1L=0x00;
60 ICR1H=0x00;
61 ICR1L=0x00;
62 OCR1AH=0x00;
63 OCR1AL=0x00;
64 OCR1BH=0x00;
65 OCR1BL=0x00;
66
67 // Timer/Counter 2 initialization
68 // Clock source: System Clock
69 // Clock value: Timer2 Stopped
70 // Mode: Normal top=0xFF
71 // OC2 output: Disconnected
72 ASSR=0<<AS2;
73 TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) | (0<<CS20);
74 TCNT2=0x00;
75 OCR2=0x00;
76
77 // Timer(s)/Counter(s) Interrupt(s) initialization
78 TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<
       TOIE0);
79
80 // External Interrupt(s) initialization
81 // INT0: Off
82 // INT1: Off
83 // INT2: Off
84 MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
85 MCUCSR=(0<<ISC2);
86
87 // USART initialization
88 // USART disabled
89 UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
90
91 // Analog Comparator initialization
92 // Analog Comparator: Off
93 // The Analog Comparator's positive input is
94 // connected to the AIN0 pin
95 // The Analog Comparator's negative input is
96 // connected to the AIN1 pin
97 ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
98 SFIOR=(0<<ACME);
99
100 // ADC initialization
101 // ADC disabled
102 ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);
103
104 // SPI initialization
105 // SPI disabled
106 SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
107
108 // TWI initialization
109 // TWI disabled
110 TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
111
112 while (1)
113       {
114        variable=PIND&0x0f; //Enmascaramos los 4 bits menos significativos
115         variableA=PINC&0x0f;
116         //del puerto A ya que los demas no interesan.
```

```
117
118         if (variable<10)
119
120         PORTB=tabla7segmentos[variable];
121
122         if (variable>=10) //Si lo que leemos es mayor o igual de 10 que dibuje en el display una E de ERROR
123
124         PORTB=0x79;
125
126         if (variableA<10)
127
128         PORTA=~tabla7segmentos[variableA];
129
130         if (variableA>=10) //Si lo que leemos es mayor o igual de 10 que dibuje en el display una E de ERROR
131
132         PORTA=~0x79;
133
134       }
135 }
```

3. Circuito

Figura 2: Simulación