Práctica #8 Cronómetro
Introducción a los Microcontroladores

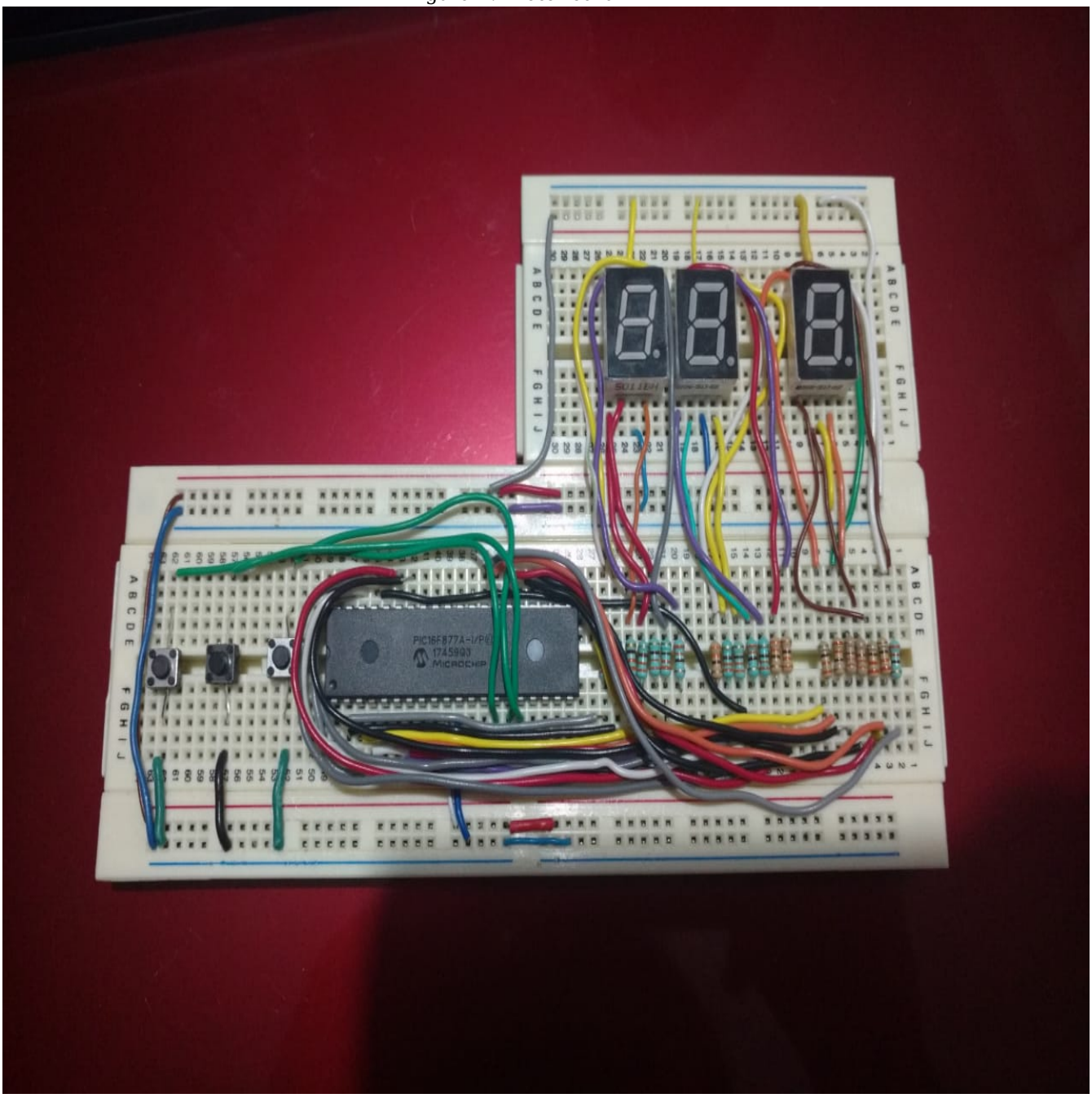
Fernando Aguilar Sánchez

Nelly Arlet Bautista Hernández
Jorge Gómez Reus

Índice

1. Protoboard

Figura 1: ProtoBoard



2. Código

```
#include <mega8535.h>
#include <delay.h>
#define reset PIND.0
#define stop PIND.1
#define start PIND.2

bit botonrp;    //reset
bit botonra;    //reset
bit botonsp;    //stop
bit botonsa;    //stop
bit botontp;    //start
bit botonta;    //start
unsigned char decimas;
unsigned char unidades;
unsigned char decenas;
unsigned char bandera;
const char tabla7segmentos [10]={0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f};


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

```
56 TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
57 TCNT0=0x00;
58 OCR0=0x00;
59
60 // Timer/Counter 1 initialization
61 // Clock source: System Clock
62 // Clock value: Timer1 Stopped
63 // Mode: Normal top=0xFFFF
64 // OC1A output: Disconnected
65 // OC1B output: Disconnected
66 // Noise Canceler: Off
67 // Input Capture on Falling Edge
68 // Timer1 Overflow Interrupt: Off
69 // Input Capture Interrupt: Off
70 // Compare A Match Interrupt: Off
71 // Compare B Match Interrupt: Off
72 TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
73 TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
74 TCNT1H=0x00;
75 TCNT1L=0x00;
76 ICR1H=0x00;
77 ICR1L=0x00;
78 OCR1AH=0x00;
79 OCR1AL=0x00;
80 OCR1BH=0x00;
81 OCR1BL=0x00;
82
83 // Timer/Counter 2 initialization
84 // Clock source: System Clock
85 // Clock value: Timer2 Stopped
86 // Mode: Normal top=0xFF
87 // OC2 output: Disconnected
88 ASSR=0<<AS2;
89 TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) | (0<<CS20);
90 TCNT2=0x00;
91 OCR2=0x00;
92
93 // Timer(s)/Counter(s) Interrupt(s) initialization
94 TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<
       TOIE0);
95
96 // External Interrupt(s) initialization
97 // INT0: Off
98 // INT1: Off
99 // INT2: Off
100 MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
101 MCUCSR=(0<<ISC2);
102
103 // USART initialization
104 // USART disabled
105 UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
106
107 // Analog Comparator initialization
108 // Analog Comparator: Off
109 // The Analog Comparator's positive input is
110 // connected to the AIN0 pin
111 // The Analog Comparator's negative input is
112 // connected to the AIN1 pin
113 ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
114 SFIOR=(0<<ACME);
115
116 // 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
         // R E S E T
         if(reset == 0)
             botonra=0;
          else
            botonra=1;
          //Cambio de flanco de 1 a 0
          if((botonrp==1)&&(botonra==0))
          {
              decimas=0;
              unidades=0;
              decenas=0;
              bandera = 0;
              PORTA=tabla7segmentos [0];
              PORTB=tabla7segmentos [0];
              PORTC=tabla7segmentos [0];
              delay_ms(40);
          }
          if((botonrp==0)&&(botonra==1))
         {
              delay_ms(40);
         }
         botonrp=botonra;

         // S T A R T
          if(start==0)
            botonsa=0;
          else
            botonsa=1;
          if((botonsp==1)&&(botonsa==0))
          {
            bandera = 2;
            delay_ms(40);
          }
          if((botonsp==0)&&(botonsa==1))
          {
            delay_ms(40);
          }
          botonsp=botonsa;

          // S T O P
           if(stop==0)
             botonta=0;
          else
             botonta=1;
          if((botontp==1)&&(botonta==0))
          {
            bandera = 0;
            delay_ms(40);
```

```
179             }
180             if((botontp==0)&&(botonta==1))
181             {
182                delay_ms(40);
183             }
184             botontp=botonta;
185 
186             if(bandera == 2)  //El cronometro avanza
187             {
188               decimas++;
189               //Autoincremento de las decimas de segundo
190               if (decimas==10){
191                   decimas=0;
192                   unidades++;
193               }
194               if(unidades==10){
195                   unidades=0;
196                   decenas++;
197               }
198               if(decenas == 6)
199               {
200                  decenas=0;
201                  decimas=0;
202                  unidades=0;
203               }
204               PORTA=tabla7segmentos [unidades];
205               PORTB=tabla7segmentos [decenas];
206               PORTC=tabla7segmentos [decimas];
207               delay_ms(100);
208             }
209             else    //El cronometro esta detenido
210                {
211                   PORTA=tabla7segmentos [unidades];
212                   PORTB=tabla7segmentos [decenas];
213                   PORTC=tabla7segmentos [decimas];
214                }
215       }
216 }
```

3. Circuito

Figura 2: Simulación