Instituto Politécnico Nacional

Escuela Superior de Computo

Materia:

Introducción a los microcontroladores.

Profesor:

Sanchez Aguilar Fernando

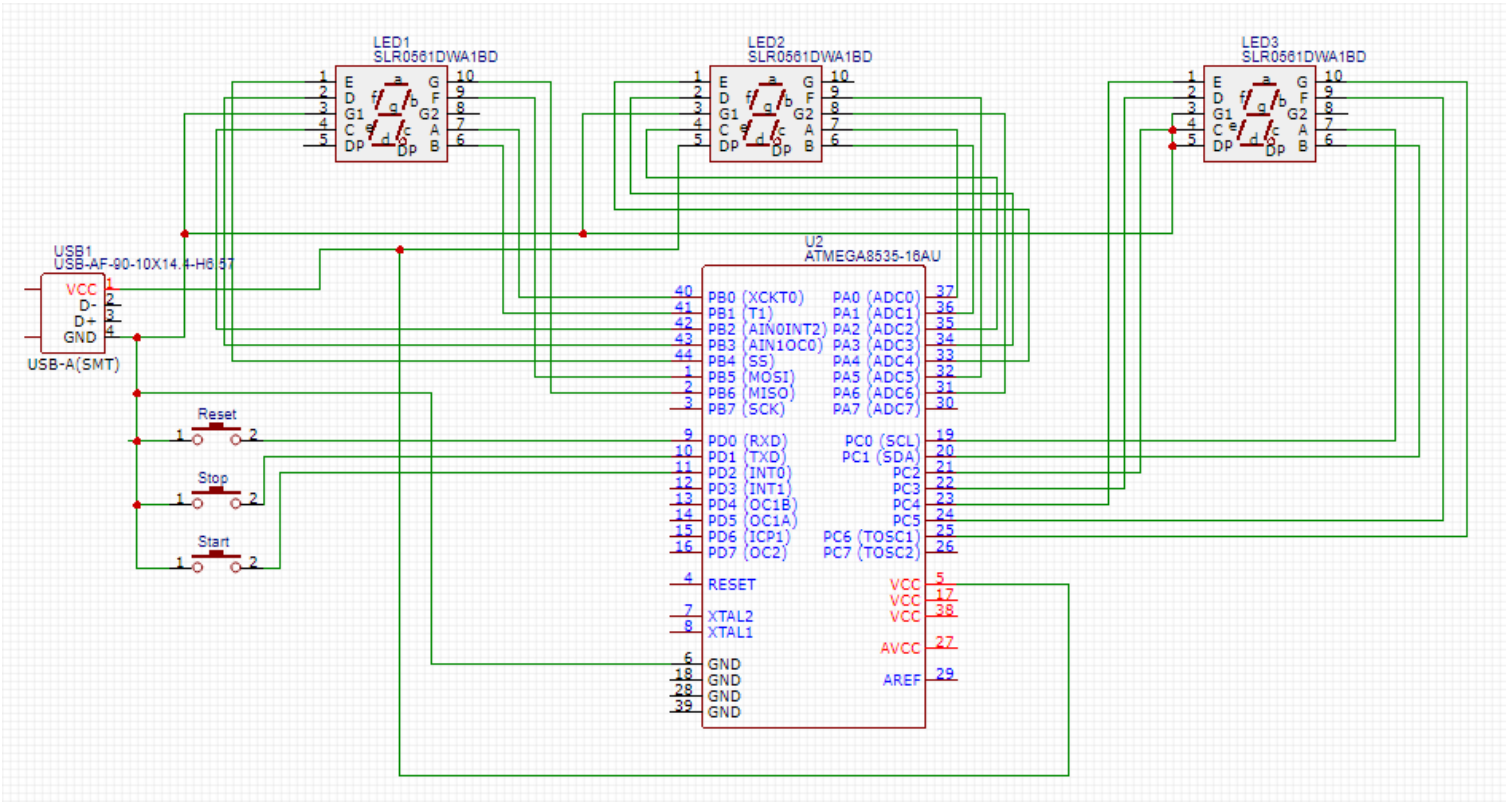
Alumnos:

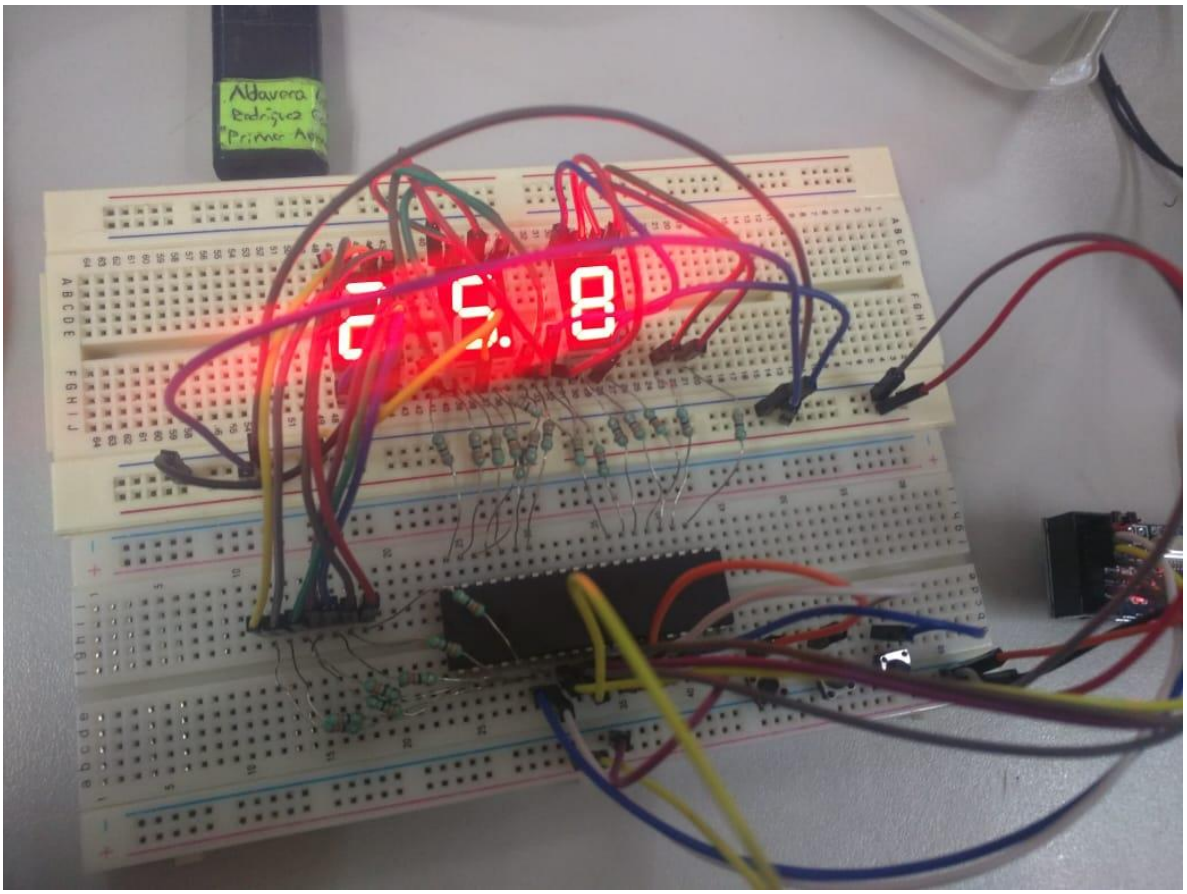
Aldavera Gallaga Iván

Lara Soto Rubén Jair

Morales Castellanos Adolfo Erik

Practica N°8


LED1
SLR0561DWA1BD
LED2
SLR0561DWA1BD
LED3
SLR0561DWA1BD
USB1
USB-AF-90-10X14.4-H6.57
USB-A(SMT)
VCC
D-
D+
GND
Reset
Stop
Start
U2
ATMEGA8535-16AU
PB0 (XCKT0)
PB1 (T1)
PB2 (AIN0INT2)
PB3 (AIN1OC0)
PB4 (SS)
PB5 (MOSI)
PB6 (MISO)
PB7 (SCK)
PD0 (RXD)
PD1 (TXD)
PD2 (INT0)
PD3 (INT1)
PD4 (OC1B)
PD5 (OC1A)
PD6 (ICP1)
PD7 (OC2)
RESET
XTAL2
XTAL1
GND
GND
GND
GND
PA0 (ADC0)
PA1 (ADC1)
PA2 (ADC2)
PA3 (ADC3)
PA4 (ADC4)
PA5 (ADC5)
PA6 (ADC6)
PA7 (ADC7)
PC0 (SCL)
PC1 (SDA)
PC2
PC3
PC4
PC5
PC6 (TOSC1)
PC7 (TOSC2)
VCC
VCC
VCC
AVCC
AREF



```
/*****************************************************
This program was created by the
CodeWizardAVR V2.60 Evaluation
Automatic Program Generator
© Copyright 1998-2012 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 02/02/2019
Author  :
Company :
Comments:


Chip type               : ATmega8535L
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 start PIND.2
#define stop PIND.1
#define reset PIND.0
int i;
unsigned char var=0, var1=0, var2=0;
const char tabla7segmentos [10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,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=O
ut
DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2) | (1<<D
DA1) | (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<<PO
RTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=O
ut
DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<D
DB1) | (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<<PO
RTB2) | (0<<PORTB1) | (0<<PORTB0);

// Port C initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=O
ut
```

```
54. DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) | (1<<DDC2) | (1<<D
    DC1) | (1<<DDC0);
55. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
56. PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PO
    RTC2) | (0<<PORTC1) | (0<<PORTC0);
57.
58. // Port D initialization
59. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
60. DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<D
    DD1) | (0<<DDD0);
61. // State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
62. PORTD=(1<<PORTD7) | (1<<PORTD6) | (1<<PORTD5) | (1<<PORTD4) | (1<<PORTD3) | (1<<PO
    RTD2) | (1<<PORTD1) | (1<<PORTD0);
63.
64. // Timer/Counter 0 initialization
65. // Clock source: System Clock
66. // Clock value: Timer 0 Stopped
67. // Mode: Normal top=0xFF
68. // OC0 output: Disconnected
69. TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) |
    (0<<CS00);
70. TCNT0=0x00;
71. OCR0=0x00;
72.
73. // Timer/Counter 1 initialization
74. // Clock source: System Clock
75. // Clock value: Timer1 Stopped
76. // Mode: Normal top=0xFFFF
77. // OC1A output: Disconnected
78. // OC1B output: Disconnected
79. // Noise Canceler: Off
80. // Input Capture on Falling Edge
81. // Timer1 Overflow Interrupt: Off
82. // Input Capture Interrupt: Off
83. // Compare A Match Interrupt: Off
84. // Compare B Match Interrupt: Off
85. TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WG
    M10);
86. TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) |
     (0<<CS10);
87. TCNT1H=0x00;
88. TCNT1L=0x00;
89. ICR1H=0x00;
90. ICR1L=0x00;
91. OCR1AH=0x00;
92. OCR1AL=0x00;
93. OCR1BH=0x00;
94. OCR1BL=0x00;
95.
96. // Timer/Counter 2 initialization
97. // Clock source: System Clock
98. // Clock value: Timer2 Stopped
99. // Mode: Normal top=0xFF
100.        // OC2 output: Disconnected
101.        ASSR=0<<AS2;
102.        TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<
    CS21) | (0<<CS20);
103.        TCNT2=0x00;
104.        OCR2=0x00;
105.
106.        // Timer(s)/Counter(s) Interrupt(s) initialization
```

```
107.      TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) |
   (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
108.
109.      // External Interrupt(s) initialization
110.      // INT0: Off
111.      // INT1: Off
112.      // INT2: Off
113.      MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
114.      MCUCSR=(0<<ISC2);
115.
116.      // USART initialization
117.      // USART disabled
118.      UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<U
   CSZ2) | (0<<RXB8) | (0<<TXB8);
119.
120.      // Analog Comparator initialization
121.      // Analog Comparator: Off
122.      ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) |
   (0<<ACIS1) | (0<<ACIS0);
123.      SFIOR=(0<<ACME);
124.
125.      // ADC initialization
126.      // ADC disabled
127.      ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<AD
   PS2) | (0<<ADPS1) | (0<<ADPS0);
128.
129.      // SPI initialization
130.      // SPI disabled
131.      SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA)
   | (0<<SPR1) | (0<<SPR0);
132.
133.      // TWI initialization
134.      // TWI disabled
135.      TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
136.
137.      while (1)
138.      {
139.            if(reset==0){
140.            var=0;
141.            var1=0;
142.            var2=0;
143.            i=0;
144.            PORTC=tabla7segmentos[var];
145.            PORTA=tabla7segmentos[var1];
146.            PORTB=tabla7segmentos[var2];
147.            }
148.
149.            if(start==0){
150.            for(i=0; i<=600; i++){
151.            var++;
152.            if(var==10){
153.            var1++;
154.            var=0;
155.            }
156.            if(var1==10){
157.            var2++;
158.            var1=0;
159.            var=0;
160.            }
161.            if(var2==6){
162.            var1=0;
```

```
            var=0;
            var2=0;
            }


            if(stop==0){
            int j,k,l;
            j=var;
            k=var1;
            l=var2;
            PORTC=tabla7segmentos[j];
            PORTA=tabla7segmentos[k];
            PORTB=tabla7segmentos[l];
            break;
            }


            if(reset==0){
            var=0;
            var1=0;
            var2=0;
            i=0;
            PORTC=tabla7segmentos[var];
            PORTA=tabla7segmentos[var1];
            PORTB=tabla7segmentos[var2];
            break;
            }

            delay_ms(100);
            PORTC=tabla7segmentos[var];
            PORTA=tabla7segmentos[var1];
            PORTB=tabla7segmentos[var2];
            }
            }

            //continuacion por si la riego
      }
      }
```