Instituto Politécnico Nacional

Escuela Superior de Computo

Materia:

Introducción a los microcontroladores.

Profesor:

Sanchez Aguilar Fernando

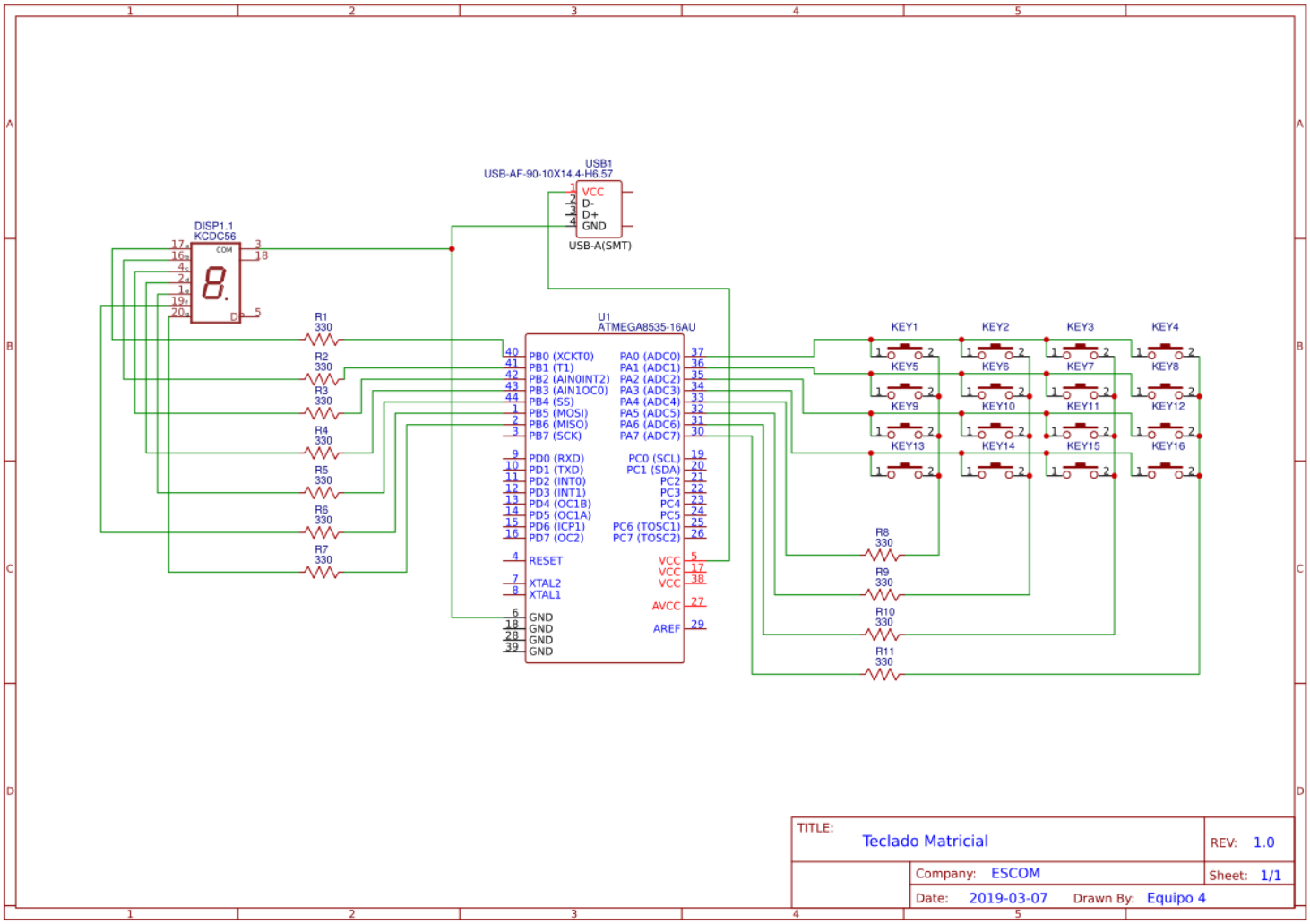
Alumnos:

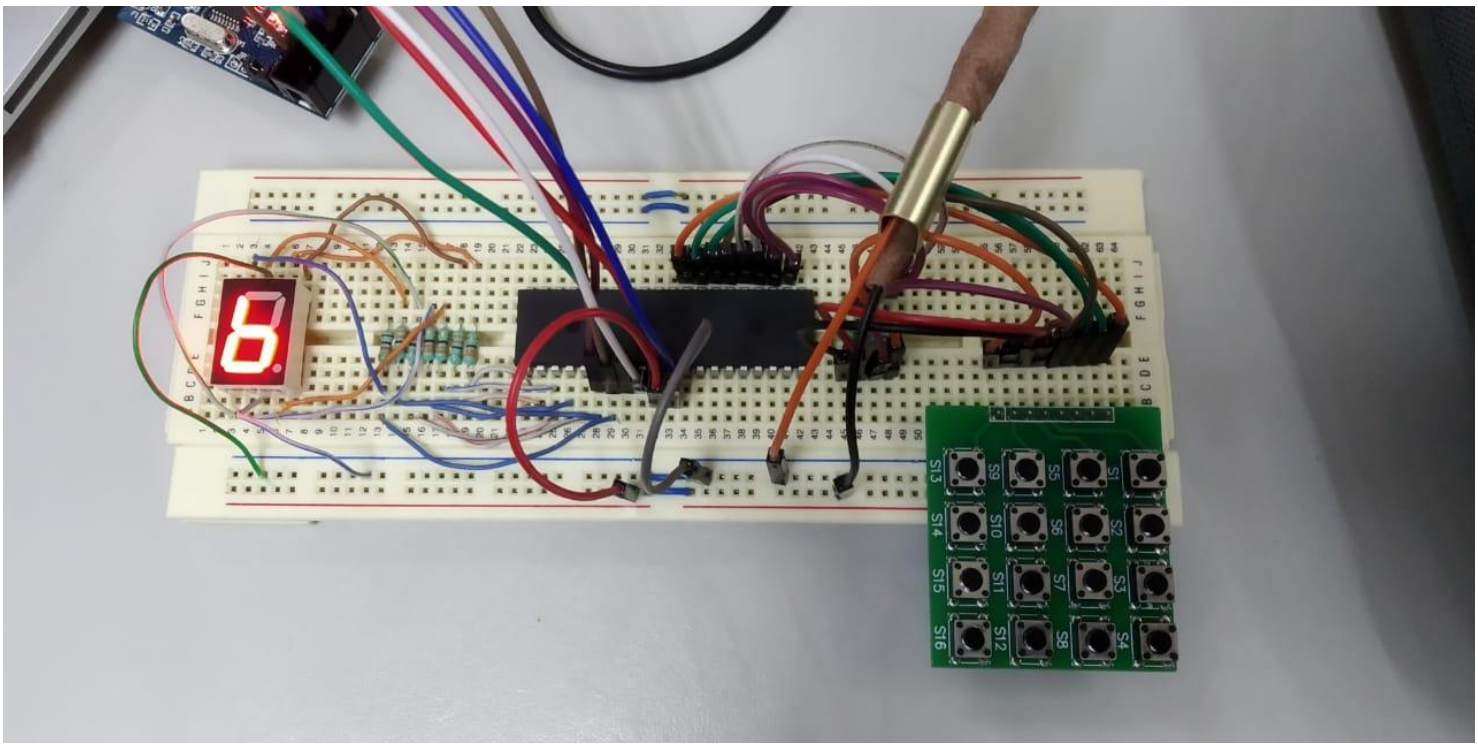
Aldavera Gallaga Iván

Lara Soto Rubén Jair

Morales Castellanos Adolfo Erik

Practica N°10


USB1
USB-AF-90-10X14.4-H6.57
VCC
D-
D+
GND
USB-A(SMT)
DISP1.1
KCDC56
COM
DP
U1
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
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
AVCC
AREF
R1 330
R2 330
R3 330
R4 330
R5 330
R6 330
R7 330
R8 330
R9 330
R10 330
R11 330
KEY1
KEY2
KEY3
KEY4
KEY5
KEY6
KEY7
KEY8
KEY9
KEY10
KEY11
KEY12
KEY13
KEY14
KEY15
KEY16
TITLE:
Teclado Matricial
REV: 1.0
Company: ESCOM
Sheet: 1/1
Date: 2019-03-07
Drawn By: Equipo 4



Código

```
/*****************************************************
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    : 12/02/2019
Author  : Equipo 4
Company : ESCOM
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>
unsigned char tecla,lectura;
const char tabla7segmentos[17]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x6f,
0x77,0x7f,0x39,0x3f,0x79,0x71,0x7d};

// Declare your global variables here

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<D
DA1) | (0<<DDA0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
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=In Bit6=In Bit5=In Bit4=In Bit3=Out Bit2=Out Bit1=Out Bit0=Out

DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (1<<DDC3) | (1<<DDC2) | (1<<D
DC1) | (1<<DDC0);
```

```
// State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTC=(1<<PORTC7) | (1<<PORTC6) | (1<<PORTC5) | (1<<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=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<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);
```

```
103.
104.       // External Interrupt(s) initialization
105.       // INT0: Off
106.       // INT1: Off
107.       // INT2: Off
108.       MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
109.       MCUCSR=(0<<ISC2);
110.
111.       // USART initialization
112.       // USART disabled
113.       UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<U
      CSZ2) | (0<<RXB8) | (0<<TXB8);
114.
115.       // Analog Comparator initialization
116.       // Analog Comparator: Off
117.       ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) |
      (0<<ACIS1) | (0<<ACIS0);
118.       SFIOR=(0<<ACME);
119.
120.       // ADC initialization
121.       // ADC disabled
122.       ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<AD
      PS2) | (0<<ADPS1) | (0<<ADPS0);
123.
124.       // SPI initialization
125.       // SPI disabled
126.       SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA)
      | (0<<SPR1) | (0<<SPR0);
127.
128.       // TWI initialization
129.       // TWI disabled
130.       TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
131.
132.       while (1)
133.             {
134.             //se prueba la primera columna se envia 1110
135.             PORTC=0b11111110;
136.
137.             lectura=PINC&0b11110000;
138.             if(lectura==0b11100000)
139.             tecla=13;
140.             if(lectura==0b11010000)
141.             tecla=14;
142.             if(lectura==0b10110000)
143.             tecla=15;
144.             if(lectura==0b01110000)
145.             tecla=16;
146.
147.             //se prueba la segunda columna se envia 1101
148.             PORTC=0b11111101;
149.
150.             lectura=PINC&0b11110000;
151.             if(lectura==0b11100000)
152.             tecla=9;
153.             if(lectura==0b11010000)
154.             tecla=10;
155.             if(lectura==0b10110000)
156.             tecla=11;
157.             if(lectura==0b01110000)
158.             tecla=12;
159.
```

```
160.            //se prueba la tercera columna se envia 1011
161.
162.            PORTC=0b11111011;
163.
164.            lectura=PINC&0b11110000;
165.            if(lectura==0b11100000)
166.            tecla=5;
167.            if(lectura==0b11010000)
168.            tecla=6;
169.            if(lectura==0b10110000)
170.            tecla=7;
171.            if(lectura==0b01110000)
172.            tecla=8;
173.
174.            //se prueba la cuarta columna se envia 0111
175.
176.            PORTC=0b11110111;
177.
178.            lectura=PINC&0b11110000;
179.            if(lectura==0b11100000)
180.            tecla=1;
181.            if(lectura==0b11010000)
182.            tecla=2;
183.            if(lectura==0b10110000)
184.            tecla=3;
185.            if(lectura==0b01110000)
186.            tecla=4;
187.
188.            PORTB=tabla7segmentos[tecla];
189.            };
190.       }
```