Instituto Politécnico Nacional

Escuela Superior de Computo

Materia:

Introducción a los microcontroladores.

Profesor:

Sanchez Aguilar Fernando

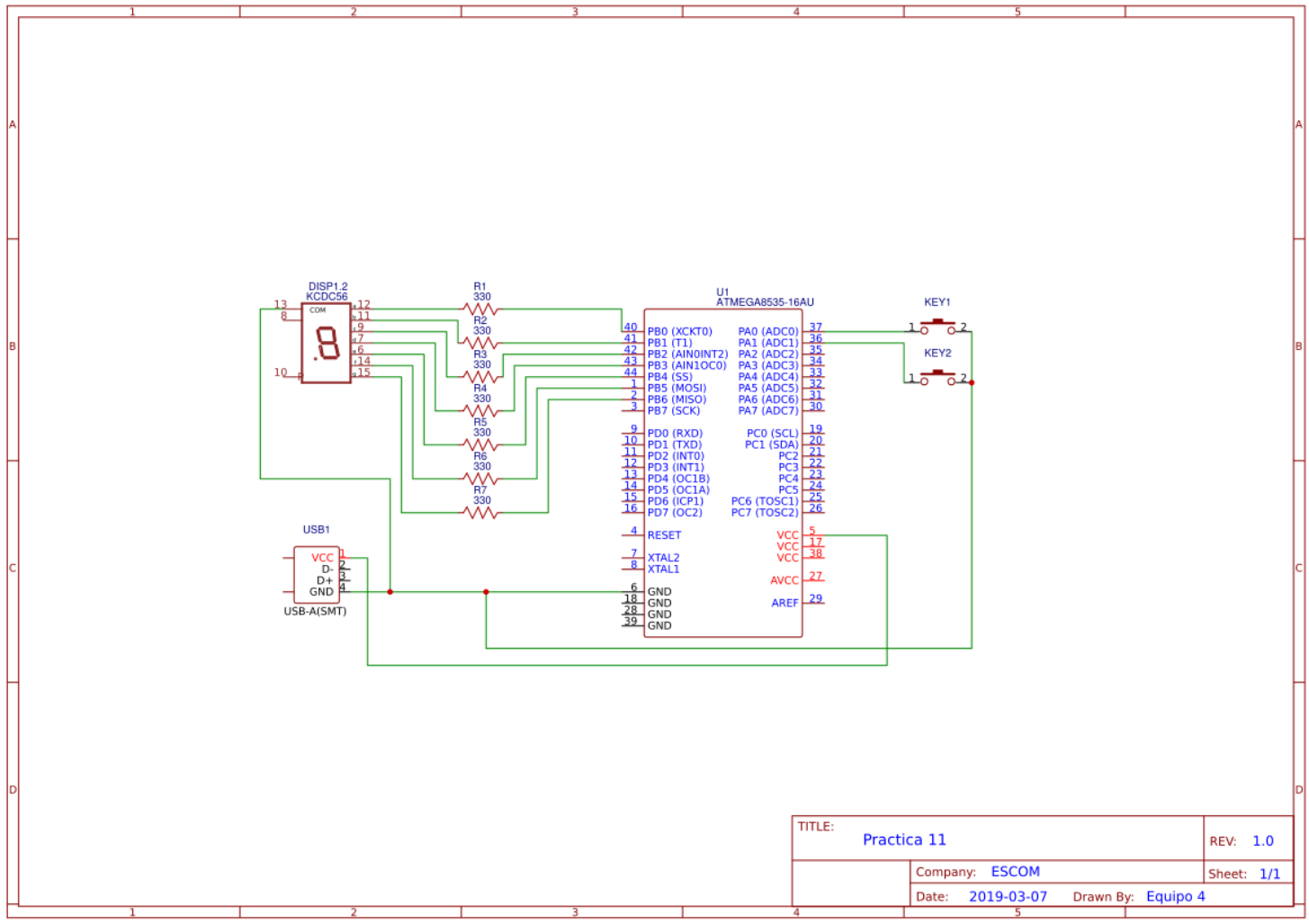
Alumnos:

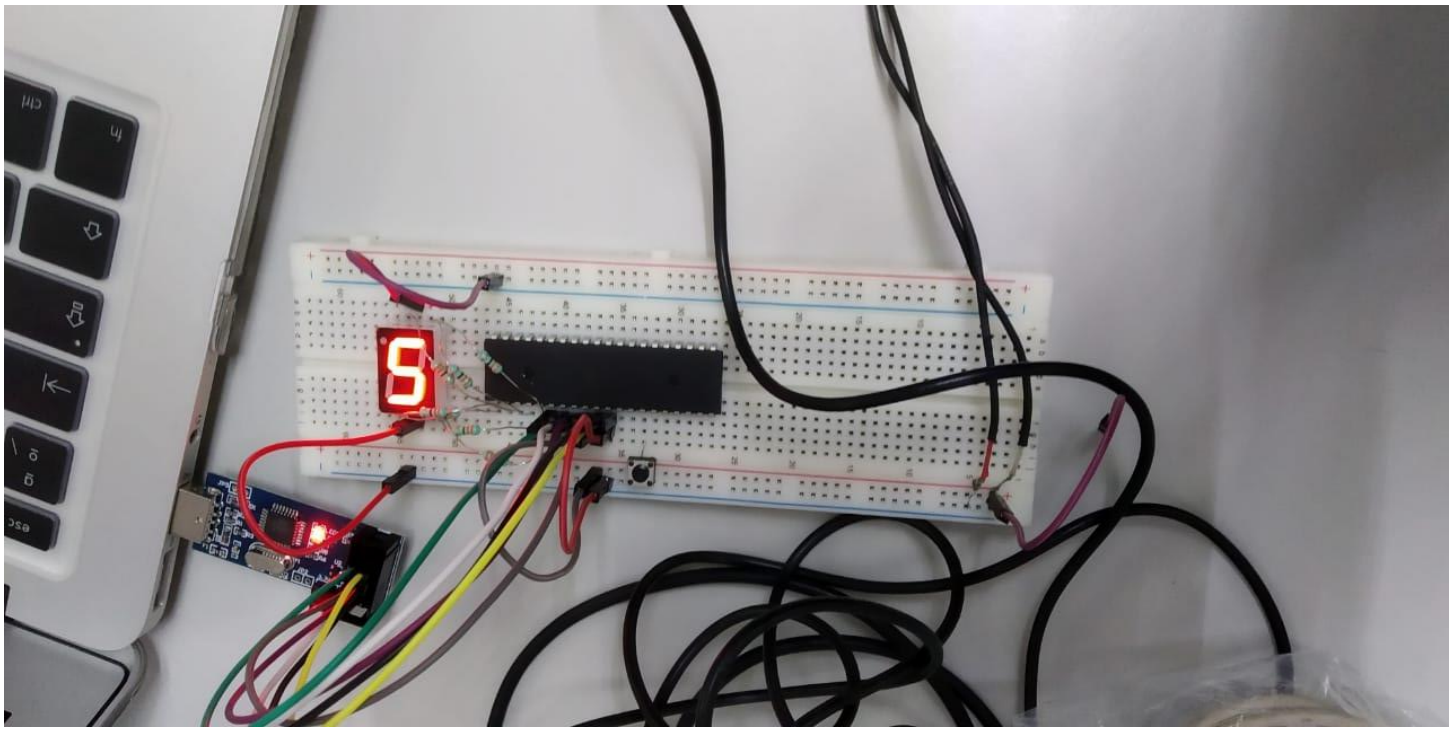
Aldavera Gallaga Iván

Lara Soto Rubén Jair

Morales Castellanos Adolfo Erik

Practica N°11


DISP1.2
KCDC56
COM
R1
330
R2
330
R3
330
R4
330
R5
330
R6
330
R7
330
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
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
KEY1
KEY2
USB1
VCC
D-
D+
GND
USB-A(SMT)
TITLE:
Practica 11
REV: 1.0
Company: ESCOM
Sheet: 1/1
Date: 2019-03-07
Drawn By: Equipo 4



```
/*****************************************************
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    : 13/02/2019
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>
#include <delay.h>
#define boton PINC.0
#define boton_guarda PINC.1
bit botonp;
bit botona;
unsigned char var;
const char tabla7segmentos [10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x6f
};
eeprom char datoaguardar;
void checa_boton(void);
// 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=In Bit2=In Bit1=In Bit0=In
```

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

```
108.       TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) |
   (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
109.
110.       // External Interrupt(s) initialization
111.       // INT0: Off
112.       // INT1: Off
113.       // INT2: Off
114.       MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
115.       MCUCSR=(0<<ISC2);
116.
117.       // USART initialization
118.       // USART disabled
119.       UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<U
   CSZ2) | (0<<RXB8) | (0<<TXB8);
120.
121.       // Analog Comparator initialization
122.       // Analog Comparator: Off
123.       ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) |
   (0<<ACIS1) | (0<<ACIS0);
124.       SFIOR=(0<<ACME);
125.
126.       // ADC initialization
127.       // ADC disabled
128.       ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<AD
   PS2) | (0<<ADPS1) | (0<<ADPS0);
129.
130.       // SPI initialization
131.       // SPI disabled
132.       SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA)
   | (0<<SPR1) | (0<<SPR0);
133.
134.       // TWI initialization
135.       // TWI disabled
136.       TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
137.
138.       if(datoaguardar>10)
139.       datoaguardar=0;
140.       var=datoaguardar;
141.
142.       while (1)
143.             {
144.               checa_boton();
145.               PORTB=tabla7segmentos[var];
146.               if(boton_guarda==0)
147.               datoaguardar=var;
148.             };
149.       }
150.
151.       void checa_boton(void){
152.
153.       if(boton==0)
154.       botona=0;
155.       else
156.       botona=1;
157.       if((botonp==1)&&(botona==0))
158.       {
159.       var++;
160.       if(var>=10)
161.       var=0;
162.       delay_ms(40);
163.       }
```

```
164.        if((botonp==0)&&(botona==1))
165.        delay_ms(40);
166.        botonp=botona;
167.
168.        }
```