**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

Introducción a los Microcontroladores

***Práctica 18:*** *Pantalla LCD 16x2*

**Integrantes del equipo:**

Martínez Ortega Juan Yael

Sampayo Hernández Mauro

**Grupo:** 3CM6

**Profesor:** *Fernando Aguilar Sánchez* **Fecha de entrega:** 17 de enero de 2021

Práctica 6: Cronómetro de 60 segundos

3CM6

ESCOM-IPN

*1. Introducción Teórica*

*2. Desarrollo experimental de la práctica*

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This program was produced by the

CodeWizardAVR V2.05.0 Professional

Automatic Program Generator

Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.

http://www.hpinfotech.com

Project :

Version :

Date : 17/01/2021

Author : NeVaDa

Company :

Comments:

Chip type : ATmega8535

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>

// Alphanumeric LCD Module functions

#include <alcd.h>

#define cambio PIND.0

#define AnioHora PIND.1

#define MesMins PIND.2

#define DiaSegs PIND.3

float c**;**

int tem**;**

int desplz**;**

int cont\_antidelay**,**time\_antidelay**;**

bit btnp**,**btna**;**

unsigned char unidades**,**decenas**,**decimas**,**x**,**seg**=**0**,**min**=**0**,**hor**=**0**,**dia**=**26**,**mes**=**11**,**opcion**=**0**;**

unsigned short anio1**=**19**;**

unsigned short anio2**=**99**;**

#define ADC\_VREF\_TYPE 0xE0

// Read the 8 most significant bits

// of the AD conversion result

unsigned char read\_adc**(**unsigned char adc\_input**)**

**{**

ADMUX**=**adc\_input **|** **(**ADC\_VREF\_TYPE **&** 0xff**);**

// Delay needed for the stabilization of the ADC input voltage

delay\_us**(**10**);**

// Start the AD conversion

ADCSRA**|=**0x40**;**

// Wait for the AD conversion to complete

**while** **((**ADCSRA **&** 0x10**)==**0**);**

ADCSRA**|=**0x10**;**

**return** ADCH**;**

**}**

// Declare your global variables here

void main**(**void**)**

**{**

// Declare your local variables here

// Input/Output Ports initialization

// Port A initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTA**=**0x00**;**

DDRA**=**0x00**;**

// Port B initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out

// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0

PORTB**=**0x00**;**

DDRB**=**0xFF**;**

// Port C initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTC**=**0x00**;**

DDRC**=**0x00**;**

// Port D initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

// State7=T State6=T State5=T State4=T State3=P State2=P State1=P State0=P

PORTD**=**0x0F**;**

DDRD**=**0x00**;**

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0**=**0x00**;**

TCNT0**=**0x00**;**

OCR0**=**0x00**;**

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Discon.

// OC1B output: Discon.

// 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**=**0x00**;**

TCCR1B**=**0x00**;**

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**=**0x00**;**

TCCR2**=**0x00**;**

TCNT2**=**0x00**;**

OCR2**=**0x00**;**

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR**=**0x00**;**

MCUCSR**=**0x00**;**

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK**=**0x00**;**

// USART initialization

// USART disabled

UCSRB**=**0x00**;**

// Analog Comparator initialization

// Analog Comparator: Off

// Analog Comparator Input Capture by Timer/Counter 1: Off

ACSR**=**0x80**;**

SFIOR**=**0x00**;**

// ADC initialization

// ADC Clock frequency: 250.000 kHz

// ADC Voltage Reference: Int., cap. on AREF

// ADC High Speed Mode: Off

// ADC Auto Trigger Source: ADC Stopped

// Only the 8 most significant bits of

// the AD conversion result are used

ADMUX**=**ADC\_VREF\_TYPE **&** 0xff**;**

ADCSRA**=**0x82**;**

SFIOR**&=**0xEF**;**

// SPI initialization

// SPI disabled

SPCR**=**0x00**;**

// TWI initialization

// TWI disabled

TWCR**=**0x00**;**

// Alphanumeric LCD initialization

// Connections specified in the

// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:

// RS - PORTB Bit 0

// RD - PORTB Bit 1

// EN - PORTB Bit 2

// D4 - PORTB Bit 4

// D5 - PORTB Bit 5

// D6 - PORTB Bit 6

// D7 - PORTB Bit 7

// Characters/line: 16

lcd\_init**(**16**);**

desplz**=**0**;**

//Variables para generar un retraso cada que se presiona el boton de cambio en fecha y hora

cont\_antidelay**=**0**;**

time\_antidelay**=**10**;**

**while** **(**1**)**

**{**

//Detecta si entro un 0 en el PIND.0 (el bot󮠦ue prersionado)

**if(**cambio**==**0**)**

btna**=**0**;**

**else**

btna**=**1**;**

//Comprueba si se presiona el bot󮍊 **if((**btnp**==**1**)&&(**btna**==**0**)){**

**if(**opcion**==**0**)**//si se presiona el bton "cambio" y opcion esta en 0, se cambia esta a 1

opcion**=**1**;**

**else**//si se presiona el boton "cambio" y opcion esta en 1, se cambia esta a 0

opcion**=**0**;**

**}**

btnp**=**btna**;**

//-----FECHA Y HORA-----

/\*Cambios en fecha y hora\*/

**if(**opcion**==**1**){**

**if(**AnioHora**==**0**){**

**if(**cont\_antidelay**>**time\_antidelay**){**

cont\_antidelay**=**0**;**

hor**++;**

**}else{**

cont\_antidelay**++;**

**}**

**}**

**if(**MesMins**==**0**){**

**if(**cont\_antidelay**>**time\_antidelay**){**

cont\_antidelay**=**0**;**

min**++;**

**}else{**

cont\_antidelay**++;**

**}**

**}**

**if(**DiaSegs**==**0**){**

**if(**cont\_antidelay**>**time\_antidelay**){**

cont\_antidelay**=**0**;**

seg**++;**

**}else{**

cont\_antidelay**++;**

**}**

**}**

**}else{**

**if(**AnioHora**==**0**){**

**if(**cont\_antidelay**>**time\_antidelay**){**

cont\_antidelay**=**0**;**

anio2**++;**

**if(**anio2**>**99**){**

anio1**++;**

anio2**=**0**;**

**}**

**}else{**

cont\_antidelay**++;**

**}**

**}**

**if(**MesMins**==**0**){**

**if(**cont\_antidelay**>**time\_antidelay**){**

cont\_antidelay**=**0**;**

mes**++;**

**}else{**

cont\_antidelay**++;**

**}**

**}**

**if(**DiaSegs**==**0**){**

**if(**cont\_antidelay**>**time\_antidelay**){**

cont\_antidelay**=**0**;**

dia**++;**

**}else{**

cont\_antidelay**++;**

**}**

**}**

**}**

/\*Avance de fecha y hora\*/

//Contador de ciclos para segundos

**if(**desplz**>**49**){**

desplz**=**0**;**

seg**++;**

**}else{**

desplz**++;**

**}**

//Avance de hora

**if(**seg**>**59**){**

min**++;**

seg**=**0**;**

**}**

**if(**min**>**59**){**

hor**++;**

min**=**0**;**

seg**=**0**;**

**}**

**if(**hor**>**23**){**

dia**++;**

hor**=**0**;**

seg**=**0**;**

min**=**0**;**

**}**

**if(**dia**>**31**){**

mes**++;**

dia**=**1**;**

**}**

**if(**mes**>**12**){**

anio2**++;**

mes**=**1**;**

**if(**anio2**>**99**){**

anio1**++;**

anio2**=**0**;**

**}**

**}**

//Posici󮠤e hora

lcd\_gotoxy**(**0**,**1**);**

lcd\_putchar**((**hor**/**10**)+**'0'**);**

lcd\_gotoxy**(**1**,**1**);**

lcd\_putchar**((**hor**%**10**)+**'0'**);**

lcd\_gotoxy**(**2**,**1**);**

lcd\_putchar**(**':'**);**

lcd\_gotoxy**(**3**,**1**);**

lcd\_putchar**((**min**/**10**)+**'0'**);**

lcd\_gotoxy**(**4**,**1**);**

lcd\_putchar**((**min**%**10**)+**'0'**);**

lcd\_gotoxy**(**5**,**1**);**

lcd\_putchar**(**':'**);**

lcd\_gotoxy**(**6**,**1**);**

lcd\_putchar**((**seg**/**10**)+**'0'**);**

lcd\_gotoxy**(**7**,**1**);**

lcd\_putchar**((**seg**%**10**)+**'0'**);**

//Posici󮠤e fecha

lcd\_gotoxy**(**0**,**0**);**

lcd\_putchar**((**anio1**/**10**)+**'0'**);**

lcd\_gotoxy**(**1**,**0**);**

lcd\_putchar**((**anio1**%**10**)+**'0'**);**

lcd\_gotoxy**(**2**,**0**);**

lcd\_putchar**((**anio2**/**10**)+**'0'**);**

lcd\_gotoxy**(**3**,**0**);**

lcd\_putchar**((**anio2**%**10**)+**'0'**);**

lcd\_gotoxy**(**4**,**0**);**

lcd\_putchar**(**'-'**);**

lcd\_gotoxy**(**5**,**0**);**

lcd\_putchar**((**mes**/**10**)+**'0'**);**

lcd\_gotoxy**(**6**,**0**);**

lcd\_putchar**((**mes**%**10**)+**'0'**);**

lcd\_gotoxy**(**7**,**0**);**

lcd\_putchar**(**'-'**);**

lcd\_gotoxy**(**8**,**0**);**

lcd\_putchar**((**dia**/**10**)+**'0'**);**

lcd\_gotoxy**(**9**,**0**);**

lcd\_putchar**((**dia**%**10**)+**'0'**);**

//-----PALABRA ESCOM-----

lcd\_gotoxy**(**11**,**0**);**

lcd\_putsf**(**"ESCOM"**);**

//-----TEMPERATURA-----

c**=**read\_adc**(**0**);**

//c=(x);

**if(**c**>**99**)**

c**=**99**;**

tem**=**c**\***10**;**

decenas**=**tem**/**100**;**

tem**=**tem**%**100**;**

decimas**=**tem**%**10**;**

unidades**=**tem**/**10**;**

lcd\_gotoxy**(**10**,**1**);**

lcd\_putchar**(**decenas**+**'0'**);**

lcd\_gotoxy**(**11**,**1**);**

lcd\_putchar**(**unidades**+**'0'**);**

lcd\_gotoxy**(**12**,**1**);**

lcd\_putchar**(**'.'**);**

lcd\_gotoxy**(**13**,**1**);**

lcd\_putchar**(**decimas**+**'0'**);**

lcd\_gotoxy**(**14**,**1**);**

lcd\_putchar**(**'0'**+**175**);**

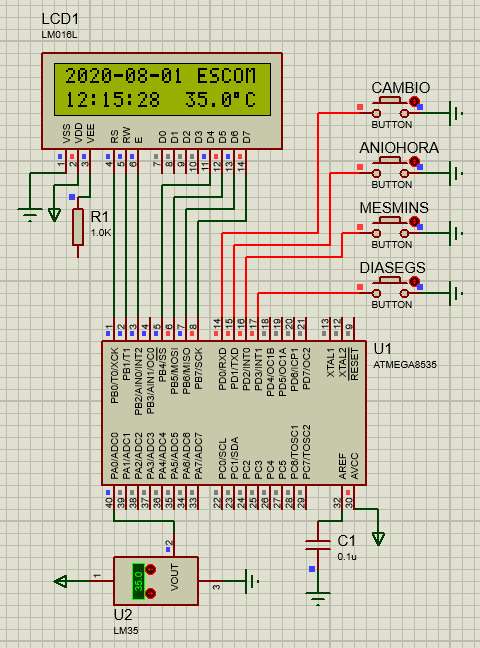
lcd\_gotoxy**(**15**,**1**);**

lcd\_putchar**(**'C'**);**

**}**

**}**

**2.2 Simulación**



*3. Conclusiones Individuales*

**3.1 Martínez Ortega Juan Yael**

**3.2 Sampayo Hernández Mauro**

*4. Anexos*

1. **Hoja de especificaciones del Microcontrolador ATMega8535**

