



INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



INTRODUCCIÓN A LOS MICROCONTROLADORES

PRACTICA #9- LCD

Integrantes:

- ✓ *Lomelí García Martín*
- ✓ *Pacchiano Alemán Alain*
- ✓ *Trejo Martínez Francisco*

PROFESOR: Pérez Pérez José Juan

GRUPO: 3CM3

08/11/2016

MARCO TEÓRICO

LCD

Los Display LCD son visualizadores pasivos, esto significa que no emiten luz como el visualizador o display alfanumérico hecho a base de un arreglo de diodos LEDs. Es por esa razón que, algunas veces, cuando intentamos ver la hora en un reloj que utiliza esta tecnología, es necesario una fuente de luz adicional.

El Display LCD tiene muy bajo consumo de energía si se lo compara con el display o visualizador alfanumérico y son compatibles con la tecnología CMOS, característica que permite que se utilice en equipos portátiles (ejemplo: los relojes de pulsera, calculadoras, etc.). Tiene una vida aproximada de 50,000 horas. Hay diferentes tipos de presentaciones y son muy fáciles de configurar. Hay desde visualizadores comunes de 7 segmentos, hasta una matriz de puntos, todos ellos muy delgados.

El LCD modifica la luz que lo incide. Dependiendo de la polarización que se esté aplicando, el LCD reflejará o absorberá más o menos luz. Cuando un segmento recibe la tensión de polarización adecuada no reflejará la luz y aparecerá en la pantalla del dispositivo como un segmento oscuro. Seguro que más de un lector habrá visto este fenómeno en calculadoras, relojes, etc.



Código fuente del programa

Esta práctica consistió en la visualización de un mensaje que se pudiera visualizar en dos líneas del display, por lo cual se procedió a la programación en ensamblador de una manera poco convencional pero que para nuestro objetivo funcionó bastante bien.

A continuación se muestra el código implementado para este programa.

```
.equ lcd_Clear = 0b00000001
.equ lcd_Home = 0b00000010
.equ lcd_EntryMode = 0b00000110
.equ lcd_DisplayOff = 0b00001000
.equ lcd_DisplayOn = 0b00001100
.equ lcd_FunctionReset = 0b00111000
.equ lcd_SetCursor = 0b10000000
.equ lcd_ShiftDisplay = 0b00011100
.equ lcd_ShiftCursor = 0b00010100
.def D=R20
.def aux=r21
    SER D
    OUT DDRC,D
    OUT DDRA,D
    ldi d,2
    out portc,d
start:
    LDI D, lcd_FunctionReset
    OUT PORTA,D
    rcall escribe_instruccion
    ldi r18, 6
    ldi r19, 82

L1: dec r19
    brne L1
    dec r18
    brne L1
    nop
    LDI D, lcd_FunctionReset
    OUT PORTA,d
    rcall escribe_instruccion
    ldi r18, 33

L2: dec r18
    brne L2
    nop

LDI D, lcd_FunctionReset
OUT PORTA,D
rcall escribe_instruccion
ldi r18, 33

L3: dec r18
    brne L3
    nop
    LDI D, lcd_DisplayOff
    out porta,d
    rcall escribe_instruccion
    LDI D, lcd_Clear
    out porta,d
    rcall escribe_instruccion
    LDI D, lcd_EntryMode
    out porta,d
    rcall escribe_instruccion
    LDI D, lcd_DisplayOn
    out porta,d
    rcall escribe_instruccion

//Datos LCD
LDI D,1
OUT PORTC,D
//HOLA PROFE

//H
ldi d, 0B01001000
OUT PORTA,D
rcall escribe_instruccion

//O
ldi d, 0B01001111
OUT PORTA,D
rcall escribe_instruccion
```

```

//L
ldi d, 0B01001100
OUT PORTA,D
rcall escribe_instruccion

//A
ldi d, 0B01000001
OUT PORTA,D
rcall escribe_instruccion

//<space>
ldi d, 0B00100000
OUT PORTA,D
rcall escribe_instruccion

//segundo display
ldi aux,59
if:
cpi aux,0
brne disminuye
rjmp sigue

disminuye:
    dec aux;
    ldi d,0
    out portc,d
    ldi d, lcd_ShiftCursor
    OUT PORTA,D
    rcall escribe_instruccion
    ldi d,1
    out portc,d
    rjmp if

sigue:
//P
ldi d, 0B01010000
OUT PORTA,D
rcall escribe_instruccion

//R
ldi d, 0B01010010
OUT PORTA,D
rcall escribe_instruccion

//O
ldi d, 0B01001111

```

```

OUT PORTA,D
rcall escribe_instruccion

//F
ldi d, 0B01000110
OUT PORTA,D
rcall escribe_instruccion

//E
ldi d, 0B01000101
OUT PORTA,D
rcall escribe_instruccion

//<space>
ldi d, 0B00100000
OUT PORTA,D
rcall escribe_instruccion

//J
ldi d, 0B01001010
OUT PORTA,D
rcall escribe_instruccion

//U
ldi d, 0B01010101
OUT PORTA,D
rcall escribe_instruccion

//A
ldi d, 0B01000001
OUT PORTA,D
rcall escribe_instruccion

//N
ldi d, 0B01001110
OUT PORTA,D
rcall escribe_instruccion
LOOP:
NOP
rjmp LOOP

escribe_instruccion:
sbi portc,1
rcall delay
cbi portc,1
ret

```

delay:		brne WGLOOP1
ldi R17, \$1F		dec R17
WGLOOP0:		brne WGLOOP0
ldi R18, \$2A		nop
WGLOOP1:		ret
dec R18		

CONCLUSIONES

Lomelí García Martín: En la práctica 10 se utilizó un LCD para mostrar un mensaje en dos líneas del mismo, se utilizó el lenguaje ensamblador aunque de una manera poco arcaica pero para lo que nosotros necesitábamos realizar no funcionó muy bien.

Pacchiano Alemán Alain: La práctica solo consistió en el uso de un display LCD que mostrará mensajes como si fuera una barra de display como en anteriores prácticas.

Trejo Martínez Francisco: En esta ocasión aprendimos a programar un LCD que mostrará mensajes determinados ya sea en una sola línea o en dos como se aplicó en esta práctica.