

**Universidad Nacional de Ingeniería
Facultad de Ciencias**

Arquitectura de computadores

Display LCD 16x2

Prof.: Lic. César Martín Cruz S.
ccruz@uni.edu.pe

2012 - II

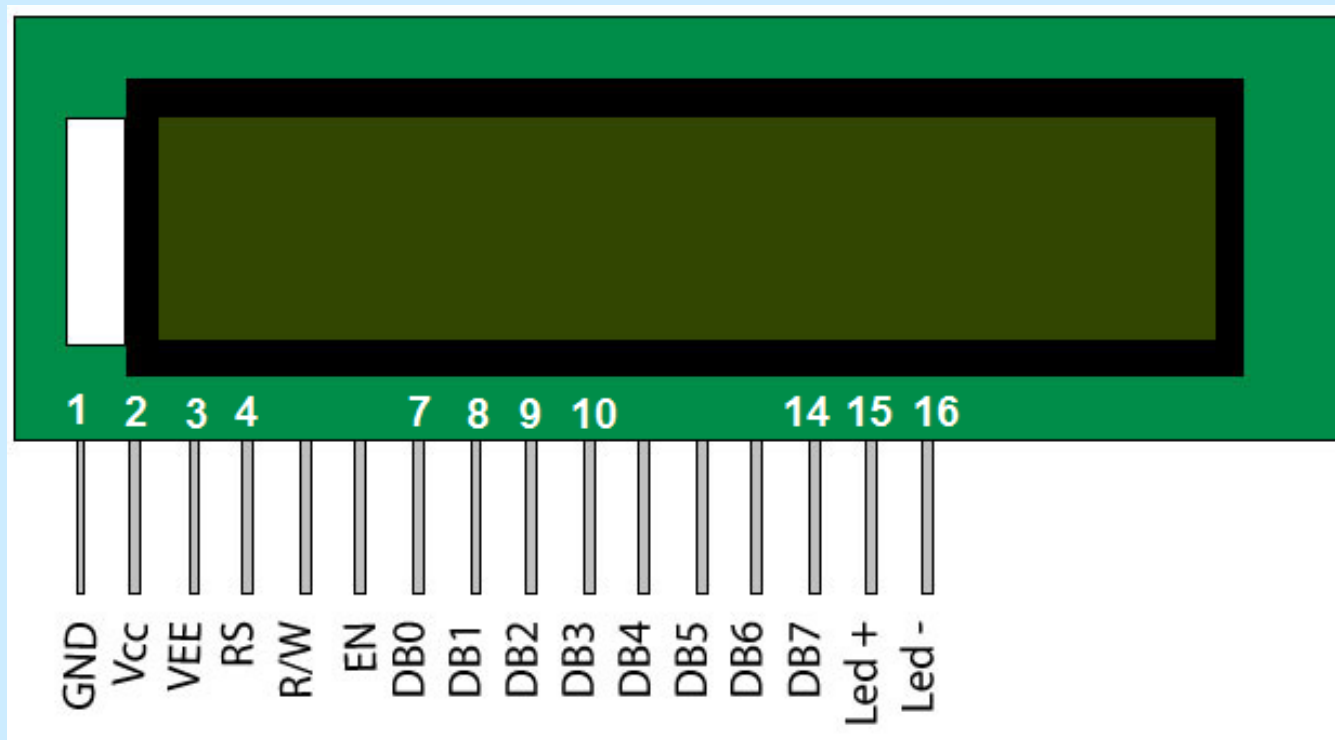
Pantalla LCD

El LCD (**Liquid Crystal Display**), display de cristal líquido es un dispositivo electrónico que hace posible la presentación efectiva de información.

Una pantalla LCD de 16x2 significa que puede mostrar 16 caracteres por línea y existen 2 líneas de este tipo. En este LCD cada carácter se visualiza en una matriz de píxeles de 5x7. Esta pantalla tiene dos registros, de *comandos* y *datos*.

El registro de comandos almacena las instrucciones dadas a la pantalla LCD. Un comando es una instrucción dada a la pantalla LCD para hacer una tarea predefinida como inicializarla, hacer la limpieza de la pantalla, ajustar la posición del cursor, etc. El registro de datos almacena los datos que se mostrarán en la pantalla LCD. Los datos son valores ASCII de los caracteres que se visualizan en la pantalla.

Diagrama de Pines del LCD



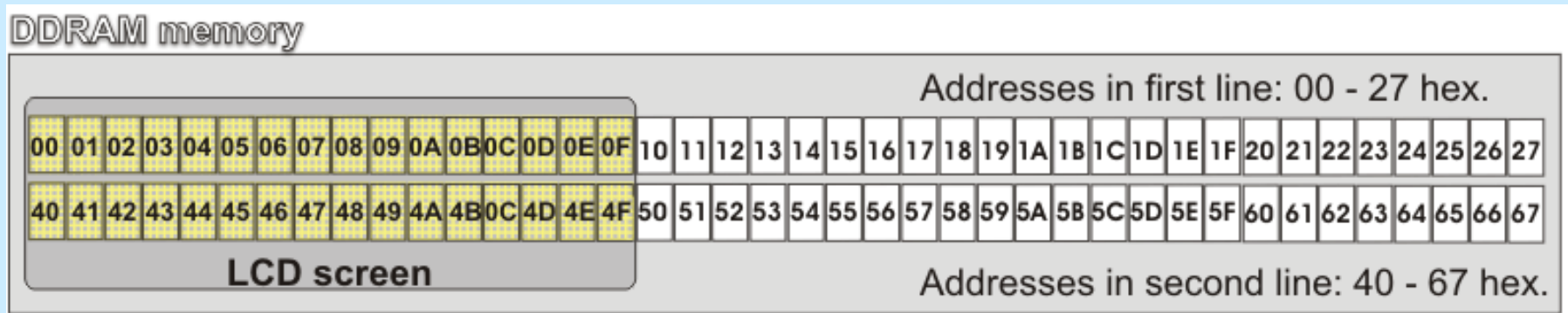
Pines del LCD

Pin	Nombre	Descripción	
1	V _{SS}	Tierra	0 V
2	V _{CC}	Fuente de voltage	+5 V
3	V _{EE}	Fuente de voltage para el contraste	Ajuste de contraste suministrado por un potenciómetro conectado a V _{CC}
4	RS	Register Select	RS=0 selecciona <i>Registro de Comando</i> RS=1 selecciona <i>Registro de Datos</i>
5	R/W	Read/write	R/W=0 escribe al registro R/W=1 lee del registro
6	EN	Enable	Un pulso de <i>bajo - alto - bajo</i> (mínimo de un ancho de 450ns) es dado cuando el dato es enviado a los pines de datos
7	DB0	Para mostrar letras o números, sus códigos ASCII son enviados a los pines de datos (con RS=1). También comandos de instrucciones son enviados a estos pines.	
8	DB1		
9	DB2		
10	DB3		Pines de datos de 8-bits
11	DB4		
12	DB5		
13	DB6		
14	DB7		
15	Led+	Backlight V _{CC}	+5 V
16	Led-	Backlight Tierra	0 V

El Display

El controlador LCD contiene una memoria **RAM** para almacenar códigos de 80 caracteres. Esta memoria se llama *Display Data RAM* o **DDRAM**. Con un display de 16 caracteres por línea y dos líneas, uno puede pensar de dos líneas virtuales de 40 caracteres cada uno. De estos 40 caracteres, solamente 16 caracteres son vistos en cualquier tiempo dado.

Para escribir un carácter en una ubicación dada, la dirección de la ubicación en el DDRAM debe primero ser seleccionado. Si más de 16 caracteres se envían, entonces todos ellos se memorizan, pero sólo los primeros dieciséis caracteres serán visibles. Con el fin de visualizar el resto de ellos, un comando de desplazamiento debe ser utilizado.



Conjunto de Caracteres

El **LCD** puede mostrar caracteres preprogramados o caracteres definidos por el usuario. El controlador LCD contiene un generador de caracteres ROM llamado **CGROM** que contiene 192 caracteres. Los caracteres son seleccionados por su código.

El controlador LCD también contiene el Character Generator RAM o **CGRAM**, que almacena 8 caracteres definidos por el usuario. Los caracteres definidos por el usuario deben primero ser cargados en CGRAM y luego llamados para ser mostrados.



Conjunto de Caracteres

		4 higher bits in address															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
4 lower bits in address	xxxx0000	CG RAM (1)		0	1	P	`	F				-	9	3	α	p	
	xxxx0001	(2)		!	1	A	Q	a	q			。	7	チ	4	ä	q
	xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	×	ß	θ
	xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
	xxxx0100	(5)		\$	4	D	T	d	t			、	エ	ト	ト	μ	Ω
	xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	1	5	Ü
	xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
	xxxx0111	(8)		'	7	G	W	g	w			ア	キ	ヌ	ラ	g	π
	xxxx1000	(1)		<	8	H	X	h	x			イ	ク	ネ	リ	フ	Σ
	xxxx1001	(2)		>	9	I	Y	i	y			ッ	ケ	ル	ル	´	4
	xxxx1010	(3)		*	:	J	Z	j	z			エ	コ	ン	レ	j	チ
	xxxx1011	(4)		+	;	K	[k	{			オ	サ	ヒ	ロ	*	斤
	xxxx1100	(5)		,	<	L	¥	l	l			ヤ	シ	フ	ワ	¢	円
	xxxx1101	(6)		-	=	M]	m	}			ユ	ズ	ヘ	ン	も	÷
	xxxx1110	(7)		.	>	N	^	n	÷			ヨ	セ	ホ	°	ñ	
	xxxx1111	(8)		/	?	O	_	o	€			ッ	ソ	マ	°	ö	■

Operaciones en el controlador LCD

Una operación en el controlador LCD es o una operación de control o una operación de datos. Las operaciones de control pueden ser comandos o instrucciones enviados al módulo. En una aplicación típica, las instrucciones son enviadas al módulo con el fin de localizar el módulo en un cierto modo de operación.

La operación es una operación de control cuando el pin **RS** (Register Select) es sostenido en nivel bajo (0 lógico). Las instrucciones son bytes escritos al módulo.

Las operaciones de control pueden ser clasificados en cinco categorías: instrucciones de inicialización, instrucciones de cursor, instrucciones de display, instrucciones CGRAM y consulta de Status.

Instrucciones de inicialización

Power on Reset (Encendido). El módulo se enciende por si mismo cuando la energía es aplicada. Los pasos de este procedimiento de encendido pueden también ser repetidos enviando 3 bytes de la instrucción 30h. Mas específicamente, DB7=DB6=0 y DB5=DB4=1. El otro nibble de la instrucción es ignorado.

Function Set (fijar tipo de funcionalidad). El tamaño del bus de datos, el número de líneas del display, y el tamaño del Font son especificados aquí. La instrucción en binario es: **(0 0 1 DL N F x x)**

Donde:

DL = 0 para un *bus de datos de 4 bits* y DL=1 para un *bus de datos de 8 bits*;

N = 0 para un display de una línea y N=1 para un display de 2 líneas;
F = 0 para fonts de caracteres de 5x7 puntos y F=1 para fonts de 5x10 puntos.

El “x” indica que el estado de este bit no tiene interés.

Ejemplo 1:

¿Cuál es la instrucción binaria para seleccionar un bus de datos de 4 bits, un display de 2 líneas y un tamaño de carácter de 5x7 puntos?.

Sol.

Se tiene **DL = 0**, **N = 1** y **F = 0**. Entonces:

La instrucción binaria es: **0 0 1 0 1 0 0 0** que equivale en hexadecimal a: 28h.

Entry Mode (Modo de Entrada). Se refiere a las acciones del cursor y el display que sigue a la operación de escritura o lectura.

La instrucción en binario es:

(0 0 0 0 0 1 I/D S)

Donde:

I/D = 0 para decrementar el cursor (mover un lugar a la izquierda);

I/D = 1 para incrementar el cursor (mover un lugar a la derecha);

S = 1 para desplazar todo el contenido del display un lugar a la izquierda o a la derecha, dependiendo del valor de I/D. Si S = 0 el display no es desplazado.

Instrucciones del Cursor

El cursor especifica la ubicación en el cual el siguiente carácter será escrito o del cual el siguiente carácter será leído. La instrucción en modo de entrada descrito arriba determina el modo de operación del cursor.

Home Cursor (posición de inicio del cursor) Esta posición del cursor está en la ubicación mas a la izquierda en la línea de arriba. La instrucción en binario es:

(0 0 0 0 0 0 1 x)

Moviendo el Cursor El cursor puede ser localizado en cualquier lugar de la DDRAM. El cursor se mueve seleccionando la dirección de la DDRAM. La instrucción en binario es:

(1 A6 A5 A4 A3 A2 A1 A0)

Cursor Hide/Show/Blink El cursor puede ser ocultado. El cursor es normalmente un “underline”, llamado el cursor línea. Alternativamente, el carácter en la posición del cursor puede o no ser parpadeante.

La instrucción en binario es:

(0 0 0 0 1 D C B)

Donde:

D = 1 para encender el display, D = 0 para apagar el display (ponerlo en blanco mientras retiene los contenidos de la DDRAM);

C = 1 para encender el cursor, C = 0 para apagar el cursor;

B = 1 para parpadear el carácter en la posición del cursor, 0 no parpadea el carácter.

Shift Cursor(desplazar el cursor)

El cursor o todo el display pueden ser trasladados a la izquierda o a la derecha. La instrucción en binario es:

(0 0 0 1 S/C R/L x x)

Donde:

S/C = 0 para desplazar el cursor y S/C = 1 para desplazar el display entero.

R/L = 0 para desplazar a la izquierda y R/L = 1 para desplazar a la derecha.

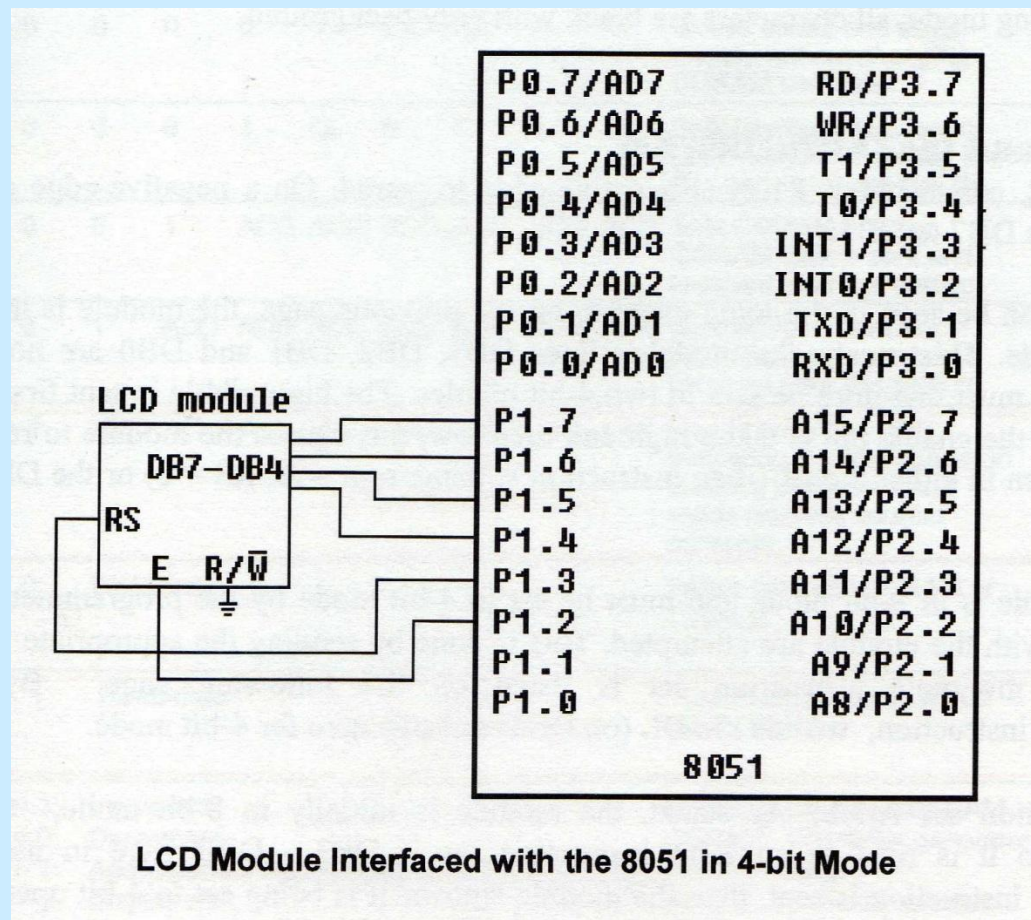
Instrucciones del Display

Clear Display(Limpiar display) Borrado de la pantalla coloca el carácter de espacio(ASCII 20H) en todas las posiciones de la DDRAM y mueve el cursor al primer carácter de la línea superior.

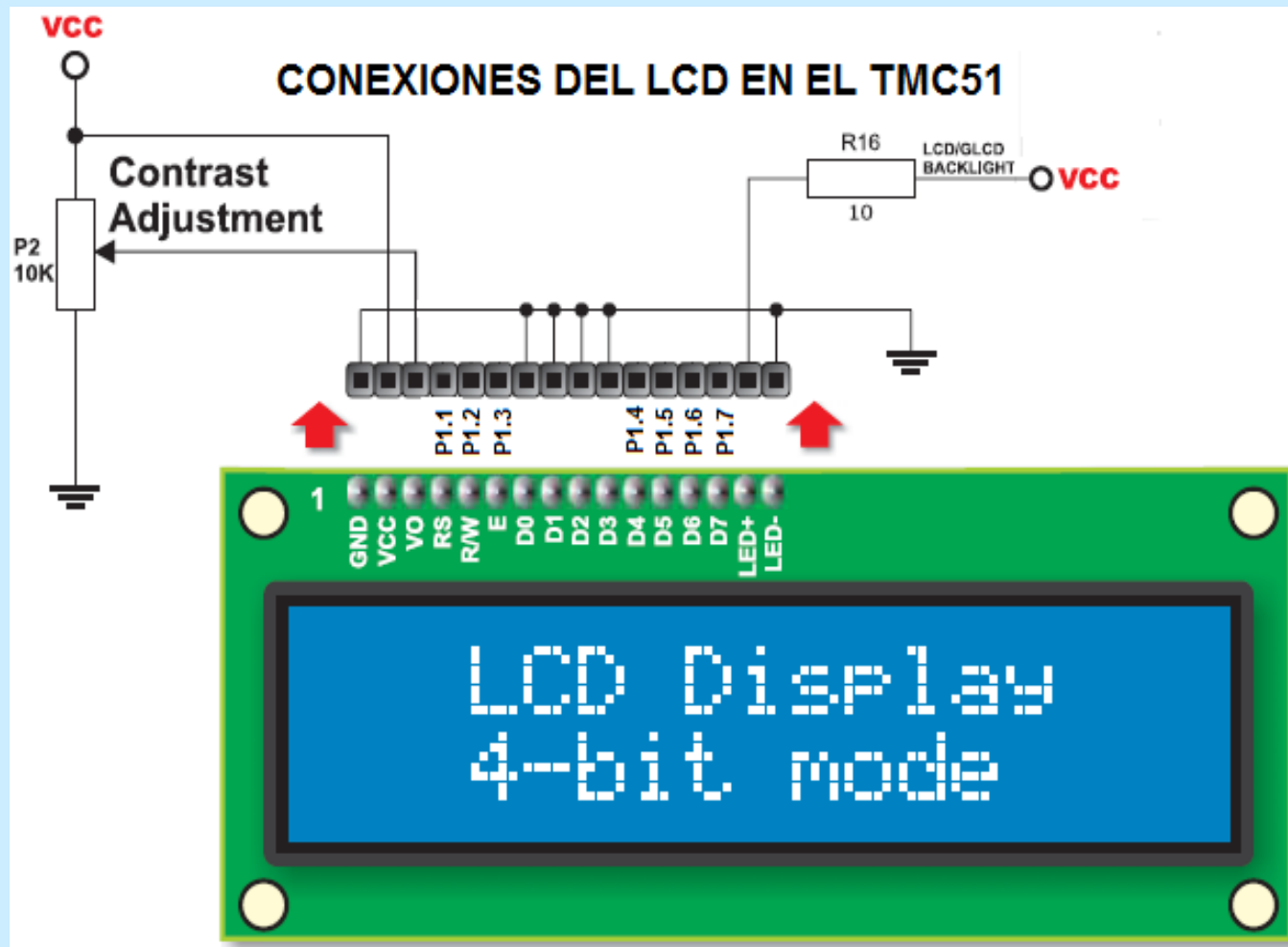
La instrucción en binario es:

(0 0 0 0 0 0 0 1)

LCD en el EDSIM51



LCD en el TMC51



Ejemplo 1:

Hacer un programa que escriba el texto “**ABCDE**” sobre el LCD.

Sol.

```
LCD_RS    equ 093h    ; p1.3 Register Select del LCD
LCD_E     equ 092h    ; p1.2 Enable del LCD
Config    equ 28h ; Function Set: Tamaño del bus de datos(4 o 8 bits), número de líneas y
; tamaño del font
entryMode equ 6       ; Modo de Entrada: Incrementa el cursor, no desplaza display
combnCur  equ 0Fh     ; Encender el Display, encender cursor y parpadea el caracter en la
; posición del cursor
    org 0000h
; poniendo datos en la RAM
    MOV 30H, #'A'
    MOV 31H, #'B'
    MOV 32H, #'C'
    mov 33H, #'D'
    mov 34H, #'E'
    MOV 35H, #0       ; caracter de indicación de fin de envío
; se inicializa el display
    mov R0, #Config
    lcall wrLCDcom4
```



```

; Configuración del "Modo de Entrada"
    mov R0,#entryMode
    lcall wrLCDcom4

;=====
; "Control del Display on/off y Configuración del cursor"
    mov R0,#combnCur
    lcall wrLCDcom4

;=====
; Comienza el envío de datos
    MOV R1, #30H      ; datos a enviar ya han sido almacenados en la RAM
                      ; del 8051, dirección de inicio 30H

loop:
    mov A, @R1        ;
    jz finish         ; si A es 0, entonces termina salta a "finish"
    mov R0,A
    lcall wrLCDdata4   ; envia datos en el registro "A" al módulo LCD
    inc R1            ;
    jmp loop          ; repite

finish:
    jmp $

;=====
; subrutina wrLCDdata4

```

; escribe una palabra de datos al LCD

; datos deben ser localizados en R0

; -----

wrLCDdata4:

 ;clr LCD_E

 setb LCD_RS ; selecciona envio de datos

 ;clr LCD_RW ; selecciona operación de escritura

 push ACC ; salva el acumulador

 mov A, R0 ; pone byte de datos en el acumulador

 mov C, ACC.7 ;

 mov P1.7, C ;

 mov C, ACC.6 ;

 mov P1.6, C ;

 mov C, ACC.5 ;

 mov P1.5, C ;

 mov C, ACC.4 ;

 mov P1.4, C ;hasta aquí se envia el nibble mas significativo

;-----

 setb LCD_E ;

 clr LCD_E ;

 mov C, ACC.3 ;

 mov P1.7, C ;

 mov C, ACC.2 ;

 mov P1.6, C ;

```

    mov C, ACC.1      ;
    mov P1.5, C       ;
    mov C, ACC.0      ;
    mov P1.4, C       ; se envia el nibble menos significativo
;-----
    setb LCD_E      ;
    clr LCD_E       ;
    lcall delay      ; retardo
    pop ACC
    ret
; =====
; subrutina wrLCDcom4
; escribe una palabra comando al LCD
; el comando debe ser localizado en r0
; -----
wrLCDcom4:
    ;clr LCD_E
    clr LCD_RS      ; limpia RS - se indica que son instrucciones
    ;clr LCD_RW      ; selecciona operación de escritura
    push ACC
    mov A,R0
    mov C, ACC.7     ;
    mov P1.7, C      ;

```

```

mov C, ACC.6      ;
mov P1.6, C      ;
mov C, ACC.5      ;
mov P1.5, C      ;
mov C, ACC.4      ;
mov P1.4, C      ;hasta aquí se envia el nibble mas significativo

```

;------

```

setb LCD_E      ;
clr LCD_E      ;
mov C, ACC.3      ;
mov P1.7, C      ;
mov C, ACC.2      ;
mov P1.6, C      ;
mov C, ACC.1      ;
mov P1.5, C      ;
mov C, ACC.0      ;
mov P1.4, C      ;se envia el nibble menos significativo

```

;------

```

setb LCD_E      ;
clr LCD_E      ;
lcall delay      ; retardo
pop ACC
ret

```

```
delay:
    MOV R0, #50
    DJNZ R0, $
    ret
end
```

Ejercicio 1:

Hacer un programa que escriba el texto “Arquitectura de computadores” sobre el display LCD.