

```

;E      = PE0
;R/W    = PE1
;RS      = PE2
;RS      = 0...CONTROL
;RS      = 1...DATOS

```

```

OCHO_BIT_DOT_FORMAT:
LDI R16,0b00111100 ;$0F...8 BITS+2LINEAS+5X10 DOT FORMAT
OUT PORTC,R16

```

Sólo se muestran algunos de los comandos de control del LCD en esta sección, pero para más configuraciones revisar la tabla 22.2 de configuración del LCD

```
LDI R16,$01
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

Bloque para configuración. Dato \$01

```
; PANTALLA ON/OFF
```

```
DISPLAY_CONTROL_ON:
```

```
LDI R16,$0F
```

```
; $0F...PANTALLA Y CURSOR ACTIVADA+
```

```
; PARPADEA CURSOR
```

```
OUT PORTC,R16
```

```
LDI R16,$01
```

```
OUT PORTE,R16
```

```
RCALL DELAY
```

```
LDI R16,$00
```

```
OUT PORTE,R16
```

```
RET
```

```
DISPLAY_CONTROL_OFF:
```

```
LDI R16,0b00001000
```

```
; $0F...PANTALLA Y CURSOR ACTIVADA+
```

```
; PARPADEA CURSOR
```

```
OUT PORTC,R16
```

```
LDI R16,$01
```

```
OUT PORTE,R16
```

```
RCALL DELAY
```

```
LDI R16,$00
```

```
OUT PORTE,R16
```

```
RET
```

```
MOVER_DERECHA:
```

```
LDI R16,$14
```

```
OUT PORTC,R16
```

```
LDI R16,$01
```

```
OUT PORTE,R16
```

```
RCALL DELAY
```

```
LDI R16,$00
```

```
OUT PORTE,R16
```

```
RET
```

```
HOME:
```

```
LDI R16,$02
```

```
OUT PORTC,R16
```

```
LDI R16,$01
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
CLEAR_DISPLAY:
LDI R16,$01
OUT PORTC,R16
```

```
LDI R16,$01
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

```
ESCRIBIR_EN_LINEA_UNO: ←
```

```
LDI R16,$80
OUT PORTC,R16
```

```
LDI R16,$01
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

```
ESCRIBIR_EN_LINEA_DOS: →
```

```
LDI R16,$C0
OUT PORTC,R16
```

```
LDI R16,$01
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

```
RECORRE_IZQUIERDA_CURSOR:
```

```
LDI R16,$10
OUT PORTC,R16
```

```
LDI R16,$01
OUT PORTE,R16
```

Quizá a veces sea necesario usar dos instrucciones adicionales cuando el LCD se comporta erráticamente al escribir de forma dinámica en la línea 1 o en la línea 2:  
HABILITAR\_LINEA\_UNO (Dato \$30), y HABILITAR\_LINEA\_DOS (Dato \$3C)

```
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

```
RECORRE_DERECHA_CURSOR:
LDI R16,$14
OUT PORTC,R16
```

```
LDI R16,$01
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

```
RECORRE_IZQUIERDA_DISPLAY:
LDI R16,$18
OUT PORTC,R16
```

```
LDI R16,$01
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

```
RECORRE_DERECHA_DISPLAY:
LDI R16,$1C
OUT PORTC,R16
```

```
LDI R16,$01
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

**A:**

```
LDI R16, 'A' ;EN CÓDIGO ASCII
OUT PORTC, R16
```

```
LDI R16, $05
OUT PORTE, R16
RCALL DELAY
```

```
LDI R16, $00
OUT PORTE, R16
RET
```

Siempre es mucho más "amigable" escribir los datos a desplegar en forma ASCII en lugar de hexadecimal o decimal. El AVR facilita el uso de ASCII para el display LCD

**B:**

```
LDI R16, 'B' ;EN CÓDIGO ASCII
OUT PORTC, R16
```

```
LDI R16, $05
OUT PORTE, R16
RCALL DELAY
LDI R16, $00
OUT PORTE, R16
RET
```

Bloque para ASCII. Dato \$05

**C:**

```
LDI R16, 'C' ;EN CÓDIGO ASCII
OUT PORTC, R16
```

```
LDI R16, $05
OUT PORTE, R16
RCALL DELAY
LDI R16, $00
OUT PORTE, R16
RET
```

```
.
.
.
.
```

**ENHE:**

```
LDI R16, 'Ñ' ;EN CÓDIGO ASCII
OUT PORTC, R16
```

```
LDI R16, $05
OUT PORTE, R16
RCALL DELAY
```

La letra "Ñ" no puede ser escrita con la etiqueta "Ñ" en el AVR Studio 4, por ello se escribe ENHE. Las letras "X", "Y" y "Z", son el nombre de los registros X, Y y Z respectivamente, y para no entrar en conflicto en el simulador escribimos en las subrutinas EQUIS, YGRIEGA y ZETA

```
LDI R16,$00
OUT PORTE,R16
RET
```

```
.
.
.
.
```

**EQUIS:**

```
LDI R16,'X' ;EN CÓDIGO ASCII
OUT PORTC,R16
```

```
LDI R16,$05
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

**YGRIEGA:**

```
LDI R16,'Y' ;EN CÓDIGO ASCII
OUT PORTC,R16
```

```
LDI R16,$05
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

**ZETA:**

```
LDI R16,'Z' ;EN CÓDIGO ASCII
OUT PORTC,R16
```

```
LDI R16,$05
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

**CERO:**

```
LDI R16,'0' ;EN CÓDIGO ASCII
OUT PORTC,R16
LDI R16,$05
OUT PORTE,R16
```

```
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

```
.
.
.
.
```

**NUEVE:**

```
LDI R16,'9'           ;EN CÓDIGO ASCII
OUT PORTC,R16
```

```
LDI R16,$05
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

```
.
.
.
.
```

**FLECHA:**

```
LDI R16,$7E           ;EN CÓDIGO ASCII
OUT PORTC,R16
```

```
LDI R16,$05
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
RET
```

**ESPACIO:**

```
LDI R16,' '           ;O DATO $20
OUT PORTC,R16
```

```
LDI R16,0b000000101
OUT PORTE,R16
RCALL DELAY
LDI R16,0b000000000
OUT PORTE,R16
RET
```

**Programa:**

```
;PROGRAMA PARA PROBAR CONFIGURACIÓN DEL DISPLAY LCD
;ACTIVANDO DOS LÍNEAS ENVIANDO MENSAJES
```

Encabezado para ATmega8515

Stack Pointer para ATmega8515

```
LDI R16,$FF
OUT DDRC,R16
```

```
LDI R16,$FF
OUT DDRE,R16
```

Se configuran "puerto C" y "puerto E"  
para datos y control del LCD

```
;PARA DARLE TIEMPO AL LCD QUE SE ESTABILICE
```

```
LDI R27,$FF
LDI R28,$0F
```



```
LDI R29,$0F
LDI R30,0
CICLO2:
DEC R27
CP R27,R30
BRNE CICLO2
```

```
DEC R28
CP R28,R30
BRNE CICLO2
```

```
DEC R29
CP R29,R30
BRNE CICLO2
```

```
;INICIALIZANDO DISPLAY
```

```
RCALL DISPLAY_CONTROL_ON
RCALL CLEAR_DISPLAY
RCALL HOME
```

Se sugiere inicializar el display antes de iniciar el "cuerpo del programa"

```
RCALL ESCRIBIR_EN_LÍNEA_UNO
RCALL H
RCALL O
RCALL L
RCALL A
```

Depende del tipo de LCD 16 x 2 usado, ya que las instrucciones para escribir en línea 1 y línea 2 pueden variar

```
RCALL ESCRIBIR_EN_LÍNEA_DOS
```

```
RCALL S
RCALL A
RCALL L
RCALL U
RCALL D
RCALL O
RCALL S
```

```
RCALL DELAY2
```

```
RCALL HOME
RCALL CLEAR_DISPLAY
```

```
RCALL ESCRIBIR_EN_LÍNEA_UNO
RCALL DOS
RCALL CERO
```

```
RCALL UNO
RCALL SEIS
```

```
RCALL ESCRIBIR_EN_LÍNEA_DOS
```

```
RCALL B
RCALL YGRIEGA
RCALL E
```

```
FIN:
RJMP FIN
```

```
DELAY_UN_SEG:
.
.
.
RET
```

```
DELAY:
LDI R27,$CB
LDI R28,$14
LDI R29,0
```

```
CICLO1:
DEC R27
CP R27,R29
BRNE CICLO1
```

```
DEC R28
CP R28,R29
BRNE CICLO1
RET
```

```
.INCLUDE "CODIGOS_ASCII_Y_CONTROL_LCD.TXT"
;*****
```

Este es el archivo incluido con .INCLUDE

**Programa:**

```
;PROGRAMA DESPLIEGA LOS CARACTERES DEL TECLADO 4x4  
;EN EL DISPLAY
```

```

.INCLUDE "M8515DEF.INC"
.CSEG
.ORG 0

RJMP SETUP                ; INICIO
RJMP INGRESA_TECLADO      ; INT0

SETUP:
LDI R16, LOW(RAMEND)      ; STACK POINTER LOW
OUT SPL, R16
LDI R16, HIGH(RAMEND)    ; STACK POINTER HIGH
OUT SPH, R16

; *****
LDI R16, $00
OUT DDRA, R16            ; "PUERTO A" como entrada de teclado

LDI R16, $FF
OUT DDRC, R16            ; "PUERTO C" como salida CONTROL/DATOS
                        ; para DISPLAY

LDI R16, $FF
OUT DDRE, R16            ; "PUERTO E" COMO SALIDA PARA CONTROL
                        ; DEL DISPLAY

; *****
; INICIALIZAR EL LCD
; LDI R16, $80            ; ESCRIBE_LÍNEA_UNO
; LDI R16, $C0            ; ESCRIBE_LÍNEA_DOS
; LDI R16, $01            ; CLEAR_DISPLAY
; LDI R16, $02            ; HOME
; LDI R16, $0F            ; DISPLAY_CONTROL_ON
; LDI R16, $30            ; UNA_LÍNEA
; LDI R16, $38            ; DOS_LÍNEAS
;
LDI R16, $0F            ; DISPLAY_CONTROL_ON
RCALL CONTROL_DISPLAY

LDI R16, $01            ; CLEAR_DISPLAY
RCALL CONTROL_DISPLAY

LDI R16, $80            ; UNA LÍNEA
RCALL CONTROL_DISPLAY

LDI R16, $02            ; HOME

```

En esta tabla se escribieron los comandos usados en este programa para no introducir el archivo `.INCLUDE "CODIGOS_ASCII_Y_CONTROL_LCD.TXT"` que se introdujo en el programa 11

```
RCALL CONTROL_DISPLAY
```

```
;*****
;HABILITA INTERRUPCION=0 PARA DETECCIÓN DE TECLA APLANADA
```

```
LDI R16, (1<<ISC01) | (1<<ISC00) ;RISING EDGE INT0
OUT MCUCR, R16
```

```
LDI R16, (1<<INT0) ;HABILITA INT0
OUT GIMSK, R16
```

```
SEI
ESPERANDO_INGRESO_DE_TECLADO:
RJMP ESPERANDO_INGRESO_DE_TECLADO
```

```
;SUBROUTINA DE INTERRUPCIÓN DE TECLA APLANADA
INGRESA_TECLADO:
```

```
IN R18, PINA ;Lectura de "puerto A" desde
teclado
```

```
LDI R17, $00
CP R18, R17
```

```
BREQ TECLA_UNO_PARCHE
```

```
LDI R17, $01
CP R18, R17
BREQ TECLA_DOS_PARCHE
```

```
LDI R17, $02
CP R18, R17
BREQ TECLA_TRES_PARCHE
```

```
LDI R17, $04
CP R18, R17
BREQ TECLA_CUATRO_PARCHE
```

```
LDI R17, $05
CP R18, R17
BREQ TECLA_CINCO_PARCHE
```

```
LDI R17, $06
CP R18, R17
BREQ TECLA_SEIS_PARCHE
```

En este programa debido a la extensión del mismo, como aparecerán errores por saltos de Branch se recurre a la técnica descrita en el capítulo 21 "Error de Branch out of range"

```
LDI R17,$08
CP R18,R17
BREQ TECLA_SIETE_PARCHE
```

```
LDI R17,$09
CP R18,R17
BREQ TECLA_OCHO_PARCHE
```

```
LDI R17,$0A
CP R18,R17
BREQ TECLA_NUEVE_PARCHE
```

```
LDI R17,$0D
CP R18,R17
BREQ TECLA_CERO_PARCHE
```

```
LDI R17,$0E
CP R18,R17
BREQ TECLA_GATO_PARCHE
```

```
LDI R17,$0C
CP R18,R17
BREQ TECLA_ASTERISCO_PARCHE
```

```
LDI R17,$03
CP R18,R17
BREQ TECLA_A_PARCHE
```

```
LDI R17,$07
CP R18,R17
BREQ TECLA_B_PARCHE
```

```
LDI R17,$0B
CP R18,R17
BREQ TECLA_C_PARCHE
```

```
LDI R17,$0F
CP R18,R17
BREQ TECLA_D_PARCHE
```

```
RETI
;*****
;PARCHES PARA ERRORES DE BRANCHES
```

```
TECLA_UNO_PARCHE:
RJMP TECLA_UNO
```

```
TECLA_DOS_PARCHE:  
RJMP TECLA_DOS
```

```
TECLA_TRES_PARCHE:  
RJMP TECLA_TRES
```

```
TECLA_CUATRO_PARCHE:  
RJMP TECLA_CUATRO
```

```
TECLA_CINCO_PARCHE:  
RJMP TECLA_CINCO
```

```
TECLA_SEIS_PARCHE:  
RJMP TECLA_SEIS
```

```
TECLA_SIETE_PARCHE:  
RJMP TECLA_SIETE
```

```
TECLA_OCHO_PARCHE:  
RJMP TECLA_OCHO
```

```
TECLA_NUEVE_PARCHE:  
RJMP TECLA_NUEVE
```

```
TECLA_CERO_PARCHE:  
RJMP TECLA_CERO
```

```
TECLA_GATO_PARCHE:  
RJMP TECLA_GATO
```

```
TECLA_ASTERISCO_PARCHE:  
RJMP TECLA_ASTERISCO
```

```
TECLA_A_PARCHE:  
RJMP TECLA_A
```

```
TECLA_B_PARCHE:  
RJMP TECLA_B
```

```
TECLA_C_PARCHE:  
RJMP TECLA_C
```

```
TECLA_D_PARCHE:  
RJMP TECLA_D
```

```

;*****
;SUBROUTINAS PARA DISPLAY

LETRAS_DISPLAY:
;E = PE0
;R/W = PE1
;RS = PE2
;RS=0 PARA CONTROL
;RS=1 PARA DATO

;AQUÍ VA EL DATO ASCII
OUT PORTC,R16

LDI R16,$05 ;Este dato corresponde al ASCII
OUT PORTE,R16
RCALL DELAY_DISPLAY
LDI R16,$00
OUT PORTE,R16
SEI
RET

;*****
CONTROL_DISPLAY:
;AQUÍ VA EL DATO QUE CORRESPONDE AL CONTROL DEL LCD (los
;valores de la tabla 22.2 "TABLA DE FUNCIONES DEL LCD"
;comentada líneas arriba).

OUT PORTC,R16

LDI R16,$01 ;Este dato corresponde al CONTROL del
;LCD
OUT PORTE,R16
RCALL DELAY_DISPLAY
LDI R16,$00
OUT PORTE,R16
RET

;*****
DELAY_DISPLAY:

LDI R16,$CB
MOV R0,R16

LDI R16,$14
MOV R1,R16

```



```
LDI R16,0
MOV R3,R16
```

```
CICLO_DISPLAY1:
DEC R0
CP R0,R3
BRNE CICLO_DISPLAY1
CICLO_DISPLAY2:
DEC R1
CP R1,R3
BRNE CICLO_DISPLAY1
RET
```

```
;*****
```

```
TECLA_UNO:                ;Estos datos se observarán en EL LCD
LDI R16,'1'                ;si son invocados
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_DOS:
LDI R16,'2'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_TRES:
LDI R16,'3'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_CUATRO:
LDI R16,'4'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_CINCO:
LDI R16,'5'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_SEIS:
LDI R16,'6'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_SIETE:
LDI R16,'7'
```

```
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_OCHO:
LDI R16,'8'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_NUEVE:
LDI R16,'9'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_CERO:
LDI R16,'0'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_GATO:
LDI R16,'#'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_ASTERISCO:
LDI R16,'*'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_A:
LDI R16,'A'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_B:
LDI R16,'B'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_C:
LDI R16,'C'
RCALL LETRAS_DISPLAY
RET
```

```
TECLA_D:
LDI R16,'D'
RCALL LETRAS_DISPLAY
RET
```