```
WRITE_EEPROM:           ;SENSO SI ESTÁ LISTA LA EEPROM PARA
                        ;SER USADA
SBIC EECR,1
RJMP WRITE_EEPROM

NOP
NOP
NOP
NOP
```

```
;;;;;SUBRUTINA DE ESCRITURA DE LA EEPROM

WRITE_EEPROM:
SBIC EECR,1              ;Senso si está listo
RJMP WRITE_EEPROM

OUT EEARL,R16            ;R16 para dirección de la EEPROM
OUT EEDR,R17             ;R17 para datos de la EEPROM

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SBI EECR, EEMWE          ;SET Bit EEMWE del registro EECR
SBI EECR, EEWE           ;Pone en "1" el EEWE del registro
                         ;EECR
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

NOP
NOP
NOP
NOP
RET
```

Se usan cuatro ciclos de reloj para fijar correctamente el dato

```
;;;;;SUBRITINA DE LECTURA DE LA EEPROM

READ_EEPROM:
SBIC EECR,1              ;Senso si está listo
RJMP READ_EEPROM

OUT EEARL,R16            ;1) Primero colocar la dirección
                         ;deseada a leer

LDI R16,$01
OUT EECR,R16             ;2) Activar Bit de lectura en el REG
                         ;de control de la EEPROM

IN R30,EEDR              ;3) Leer el dato

NOP                      ;4) 4 Ciclos de reloj de DELAY
NOP
NOP
NOP
RET
```

```
;PROGRAMA PARA ENVIAR MENSAJES EN DISPLAY MEMORIZANDO LAS
;TECLAS EN LAS EEPROM USANDO UN TECLADO MATRICIAL 4X4
```

Encabezado para ATmega8515

```
RJMP SETUP
RJMP TECLA_APLANADA

SETUP:
```

Stack Pointer para ATmega8515

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

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

LDI R16,$FF
OUT DDRE,R16            ;"PUERTO E" como salida para control
                        ;del DISPLAY

LDI R16,$40
OUT GICR,R16            ;Habilitamos INT0

LDI R16,$03
OUT MCUCR,R16           ;Filo de subida

;ÉSTOS SON LOS VALORES HEXADECIMALES DE LAS TECLAS
;Y LAS DIRECCIONES DE LA MEMORIA EEPROM A USAR

;TECLA 1...$00..DIR 0
;TECLA 2...$01..DIR 1
;TECLA 3...$02..DIR 2
;TECLA 4...$04..DIR 3
```

```
;TECLA 5...$05..DIR 4
;TECLA 6...$06..DIR 5
;TECLA 7...$08..DIR 6
;TECLA 8...$09..DIR 7
;TECLA 9...$0A..DIR 8
;TECLA 0...$0D..DIR 9
;TECLA A...$03..DIR 10
;TECLA B...$07..DIR 11
;TECLA C...$0B..DIR 12
;TECLA D...$0F..DIR 13
;TECLA #...$0E..DIR 14
;TECLA *...$0C..DIR 15

;INICIALIZAR EL LCD
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
RCALL CONTROL_DISPLAY

//AHORA MEMORIZAMOS LOS DATOS DE CADA TECLA EN LA EEPROM:

;;TECLA_1

LDI R16,0               ;DIRECCIÓN 0
LDI R17,$00             ;DATO

RCALL WRITE_EEPROM

;;TECLA_2

LDI R16,1               ;DIRECCIÓN 1
LDI R17,$01             ;DATO

RCALL WRITE_EEPROM
.
.
.
;;TECLA_D
```

Verificar el comportamiento de la memoria EEPROM en el simulador

```
LDI R16,13              ;DIRECCIÓN 13
LDI R17,$0F             ;DATO

RCALL WRITE_EEPROM

;;TECLA_#

LDI R16,14              ;DIRECCIÓN 14
LDI R17,$0E             ;DATO

RCALL WRITE_EEPROM

;;TECLA_*

LDI R16,15  ;DIRECCIÓN 15
LDI R17,$0C ;DATO

RCALL WRITE_EEPROM
```

Verificar el comportamiento de la memoria EEPROM en el simulador

```
SEI

CICLO:
RJMP CICLO              ;Esperando a que se presione una
                        ;tecla

;VAMOS A INGRESAR CADA TECLA Y COMPARARLA CON EL CONTENIDO DE
;LA EEPROM
TECLA_APLANADA:

IN R18,PINA

;TECLA 1...DATO $00..DIR 0
LDI R16,0               ;DIRECCIÓN
RCALL READ_EEPROM
CP R18,R30              ;R30 contiene el dato de EEPROM
BREQ TECLA_1

;TECLA 2...DATO $01..DIR 1
LDI R16,1               ;DIRECCIÓN
RCALL READ_EEPROM
CP R18,R30              ;R30 contiene el dato de EEPROM
BREQ TECLA_2
.
.
.
```

```
;TECLA D...DATO $0F..DIR 13
LDI R16,13              ;DIRECCIÓN
RCALL READ_EEPROM
CP R18,R30              ;R30 contiene el dato de EEPROM
BREQ TECLA_D

;TECLA #...DATO $0E..DIR 14
LDI R16,14              ;DIRECCIÓN
RCALL READ_EEPROM
CP R18,R30              ;R30 contiene el dato de EEPROM
BREQ TECLA_GATO

;TECLA *...DATO $0C..DIR 15
LDI R16,15              ;DIRECCIÓN
RCALL READ_EEPROM
CP R18,R30              ;R30 contiene el dato de EEPROM
BREQ TECLA_ASTERISCO

RETI

;;;;;SUBRUTINAS DE TECLAS PARA MOSTRAR EN LCD

TECLA_1:
LDI R16,'1'
RCALL LETRAS_DISPLAY
RETI

TECLA_2:
LDI R16,'2'
RCALL LETRAS_DISPLAY
RETI
.
.
.
TECLA_GATO:
LDI R16,'#'
RCALL LETRAS_DISPLAY
RETI

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

Por ejemplo, la **TECLA_C** en lugar de mostrar el caracter ASCII "**C**" se puede sustituir por un comando de "CLEAR_DISPLAY":

```
LDI R16,$01
;CLEAR_DISPLAY
RCALL CONTROL_DISPLAY
```

Así cada vez que presionemos la **TECLA_C** el display se borrará para seguir escribiendo

```
;;;;;SUBRUTINA DE ESCRITURA DE LA EEPROM
WRITE_EEPROM:
SBIC EECR,1             ;SENSO SI ESTÁ LISTO
RJMP WRITE_EEPROM

OUT EEARL,R16           ;R16 PARA DIRECCIÓN DE LA EEPROM

OUT EEDR,R17            ;R17 PARA DATOS DE LA EEPROM
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SBI EECR, EEMWE         ;SET Bit EEMWE DEL REGISTRO EECR
SBI EECR, EEWE          ;PON EN 1 EL EEWE DEL REG EECR
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

NOP
NOP
NOP
NOP
RET

;;;;;SUBRITINA DE LECTURA DE LA EEPROM
READ_EEPROM:
SBIC EECR,1             ;SENSO SI ESTÁ LISTO
RJMP READ_EEPROM

OUT EEARL,R16           ;1) PRIMERO LA DIRECCIÓN DESEADA A
                        ;LEER

LDI R16,$01
OUT EECR,R16            ;2) ACTIVAR Bit DE LECTURA EN EL REG
                        ;DE CONTROL DE LA ;EEPROM

IN R30,EEDR             ;3) LEER EL DATO

NOP                     ;4) 4 CICLOS DE RELOJ DE DELAY
NOP
NOP
NOP
RET

;SUBRUTINAS 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
OUT PORTE,R16
RCALL DELAY_DISPLAY
LDI R16,$00
OUT PORTE,R16
SEI
RET

;****************
CONTROL_DISPLAY:

OUT PORTC,R16

LDI R16,$01
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
```

Sugerencia: ésta es una forma de usar los registros `R0` a `R15` para una subrutina de delay. Así podemos usar los registros `R16` a `R31` para subrutinas generales