```
;*************************************************************************
;*                                                                       *
;*  NOMBRE:   RESET_LCD                                                  *
;*                                                                       *
;*  FUNCIàN:  Realiza el RESET del LCD-Driver                            *
;*                                                                       *
;*  LLAMADA:  Al inicio del programa principal                           *
;*                                                                       *
;*************************************************************************

NAME  DECLARACIONES     ; En este m¢dulo se declaran las funciones a utilizar
                        ;  por el fichero PIA_1.ASM

PUBLIC  RESET_LCD       ; Esta funci¢n es utilizada en PIA_1.ASM

EXTRN  BIT (SCKI)       ; □stos bits se definen en el m¢dulo PIA_1.ASM
EXTRN  BIT (SI)         ;  y por eso se declaran como EXTERN en el m¢dulo
EXTRN  BIT (RESETT)     ; PIA_2.ASM
EXTRN  BIT (C_D)
EXTRN  BIT (CS)


RESET_EN_LCD_SERIE  SEGMENT  CODE


        RSEG    RESET_EN_LCD_SERIE


RESET_LCD:                ; RESET DEL LCD

        SETB    SCKI      ; Ponemos todos los bits a 1
        SETB    SI
        SETB    CS
        SETB    C_D
        SETB    RESETT
        NOP
        NOP
        CLR     RESETT      ; Ponemos a 0 RESETT
        NOP
        NOP
        NOP
        SETB    RESETT      ; Ponemos a 1 RESETT
        RET

        END
```

---

```
NAME    PIA_LCD_TEST    ; Rutina para testear LCD's


EXTRN CODE (RESET_LCD) ; Indicamos al ensamblador que la funci¢n RESET_LCD
                       ; ha de buscarla en un m¢dulo externo, en este caso
                       ; en PIA2_2.ASM

PUBLIC  SCKI           ; □stos bits se definen en el m¢dulo PIA_1.ASM
PUBLIC  SI             ;  y por eso se declaran como EXTERN en el m¢dulo
PUBLIC  RESETT         ; PIA_2.ASM
PUBLIC  C_D
PUBLIC  CS


        USING  0  ; Register-Bank 0


        ORG     0000H
        JMP     START

        ORG     0003H
        JMP     INTO0

        ORG     0013H
        JMP     INTO1


;************************************
;***   DEFINICIONES IMPORTANTES   ***
;************************************
;
        SCKI    BIT     P3.7
        SI      BIT     P1.0        ;   ESTAS ser n las definitivas
        CS      BIT     P1.1
        BUSY    BIT     P1.2
        C_D     BIT     P1.3
        RESETT  BIT     P1.4
        LED_ON  BIT     P3.5
        LED_OFF BIT     P3.4
        HELP    BIT     P3.0


;************** fin *****************



;*************************************************************************
;*********************      COMIENZO PROGRAMA       ***********************
;*************************************************************************

        ORG  0020H

START:
        MOV     TCON,#04H     ;
        MOV     IE,#85H       ; Habilitamos INT0 e INT1
        MOV     IP,#04H       ; Damos prioridad a INT1
        CALL    RESET_LCD     ; Efectuamos el RESET del LCD-Driver

        SETB    C_D           ;             COMANDO
        MOV     R7,#42H       ; Mode SET
        CALL    ENVIO_DATO

        SETB    C_D           ;             COMANDO
        MOV     R7,#20H       ; Clear Data Memory
        CALL    ENVIO_DATO

        SETB    C_D           ;             COMANDO
        MOV     R7,#00H       ; Clear Blinking Data
        CALL    ENVIO_DATO

        SETB    C_D           ;             COMANDO
        MOV     R7,#11H       ; Display On
        CALL    ENVIO_DATO

        MOV     R3,#01        ; Ponemos a 1 la opci¢n de visualizaci¢n
        CLR     LED_OFF       ; Encendemos led STOP
        SETB    LED_ON        ; Apagamos led START
SAL1:                         ; Rutina del Programa principal

        SETB    HELP
        CALL    RETARDO
        CLR     HELP
        CALL    RETARDO
        JMP     SAL1          ; Salto a bucle infinito


;*************************************************************************
;*                                                                       *
;*  NOMBRE:   RETARDO                                                    *
;*                                                                       *
;*  FUNCIàN:  Realiza una espera para encender el led HELP para indicar  *
;*            actividad                                                  *
;*  LLAMADA:  Cuerpo principal del programa                              *
;*                                                                       *
;*************************************************************************
RETARDO:
        PUSH    ACC
        MOV     A,R0
        PUSH    ACC
        MOV     R0,     #05H
SAL2:   MOV     R1,     #20H
SAL3:   MOV     R2,     #255
SAL4:   DJNZ    R2,     SAL4
        DJNZ    R1,     SAL3
        DJNZ    R0,     SAL2
        POP     ACC
```

```
        MOV     R0,A
        POP     ACC
        RET


;*************************************************************************
;*                                                                       *
;*  NOMBRE:   Rutina de atenci¢n a la interrupci¢n INT0                  *
;*                                                                       *
;*  FUNCIàN:  Atiende la intepprucI¢n de pulsado de la tecla ON          *
;*                                                                       *
;*  LLAMADA:  Interrupci¢n                                               *
;*                                                                       *
;*************************************************************************

INT0:                           ;  Rutina de Interrupci¢n   INT0
        MOV     IE,#00H         ;  Desabilito Todas las interrupciones
        PUSH    ACC             ;  Prioridad(INT0) > Prioridad(INT1)
        MOV     A,PSW           ;  START
        PUSH    ACC
        MOV     A,R0
        PUSH    ACC

        SETB    LED_OFF         ; Apagamos led STOP
        CLR     LED_ON          ; Endendemos led START

        SETB    C_D               ;              COMANDO
        MOV     R7,#18H           ; Blinking ON
        CALL    ENVIO_DATO

        CJNE    R3,#1,SALTU2
        JMP     PASO1

SALTU2:
        MOV     A,R3
        SUBB    A,#02
        JZ      PASO2

        MOV     A,R3
        SUBB    A,#03
        JZ      PASO3

        MOV     A,R3
        SUBB    A,#04
        JZ      PASO4
        JNZ     PASO2
PASO1:                          ; ENCENDER TODO EL DISPLAY

        CLR     CS              ; Bajamos CS
        CLR     C_D             ; DATO
        MOV     R6, #16         ; Repetimos 16 veces
BUC1:   MOV     R7,#255         ; Envio de 0xAFF
        CALL    ENVIO_DATOS
        DJNZ    R6, BUC1        ; Saltamos al principio de BUC1
        SETB    CS              ; Subimos CS

        MOV     R3,#02H         ; Cambiamos a PASO 2
        ;SETB    LED_ON
        ;SETB    LED_OFF
        JMP     FIN_INT0

PASO4:
        JMP     PASO44
PASO3:
        JMP     PASO33

PASO2:                          ; ENCENDER FILA 1 Y 3

        CLR     CS              ; Bajamos CS
        CLR     C_D             ; DATO
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
```

```
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#255         ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x
        CALL    ENVIO_DATOS
        MOV     R7,#0          ; Envio de 0x
        CALL    ENVIO_DATOS
        SETB    CS              ; Subimos CS

        MOV     R3,#03H         ; Cambiamos a PASO 3
        ;SETB    LED_OFF
        ;CLR     LED_ON
        JMP     FIN_INT0

PASO33:                          ; ENCENDER ICONOS K7 Y K8

        CLR     CS              ; Bajamos CS
        CLR     C_D             ; DATO
        MOV     R7,#0           ; Envio de 0X0
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x0
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x0
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x0
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x0
        CALL    ENVIO_DATOS
        MOV     R7,#0H          ; Envio de 0x0
        CALL    ENVIO_DATOS
        MOV     R7,#0H           ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0H           ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0H          ; Envio de 0x
        CALL    ENVIO_DATOS
        MOV     R7,#0           ; Envio de 0x
        CALL    ENVIO_DATOS
        MOV     R7,#128         ; Envio de 0x   K7=(15)0xF0h
        CALL    ENVIO_DATOS
        MOV     R7,#128         ; Envio de 0x   K8=(16)0xF0h
        CALL    ENVIO_DATOS
        SETB    CS              ; Subimos CS

        MOV     R3,#04H         ; Cambiamos a PASO 4
        ;SETB    LED_ON
        ;CLR     LED_OFF
        JMP     FIN_INT0

PASO44:                          ; Encendemos ICONOS K7 Y K8 y numeros 13,14,15,16

        CLR     CS              ; Bajamos CS
        CLR     C_D             ; DATO
```

```
        MOV     R7,#0        ; Envio de 0X0
        CALL    ENVIO_DATOS
        MOV     R7,#0        ; Envio de 0x0
        CALL    ENVIO_DATOS
        MOV     R7,#0        ; Envio de 0x0
        CALL    ENVIO_DATOS
        MOV     R7,#0        ; Envio de 0x0
        CALL    ENVIO_DATOS
        MOV     R7,#0        ; Envio de 0x0
        CALL    ENVIO_DATOS
        MOV     R7,#0H       ; Envio de 0x0
        CALL    ENVIO_DATOS
        MOV     R7,#0H        ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0H        ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0        ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0        ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0        ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#0        ; Envio de 0x255
        CALL    ENVIO_DATOS
        MOV     R7,#224      ; Envio de un 1 y COL
        CALL    ENVIO_DATOS
        MOV     R7,#3EH      ; Envio de un 2
        CALL    ENVIO_DATOS
        MOV     R7,#250      ; Envio de 0x   K7=(15)0xF0h y 3
        CALL    ENVIO_DATOS
        MOV     R7,#227      ; Envio de 0x   K8=(16)0xF0h
        CALL    ENVIO_DATOS
        SETB    CS           ; Subimos CS


        SETB    C_D            ;             COMANDO
        MOV     R7,#1BH        ; Blinking ON
        CALL    ENVIO_DATO

        CLR     CS
        SETB    C_D          ;  COMANDO
        MOV     R6, #32      ; Repetimos 32 veces
BUC4:   MOV     R7,#207      ; Write Blinking Memory 0xCF
        CALL    ENVIO_DATOS
        DJNZ    R6, BUC4     ; Saltamos al principio de BUC4
        SETB    CS           ; Subimos CS

        MOV     R3,#01H      ; Cambiamos a PASO 1
        ;CLR     LED_ON
        ;CLR     LED_OFF

        JMP     FIN_INT0


FIN_INT0:
        CALL    RETARDO
        POP     ACC
        MOV     R0,A
        POP     ACC
        MOV     PSW,A
        POP     ACC
        MOV     IE,#85H   ; Habilito todas las interrupciones
        RETI

INTO1:                          ; Rutina de Interrupci¢n  INT1
        MOV     IE,#00H      ;  Desabilito Todas las interrupciones
        PUSH    ACC            ; Prioridad(INT1) < Prioridad(INT0)
        MOV     A,PSW          ; STOP
        PUSH    ACC
        MOV     A,R0
        PUSH    ACC

        SETB    LED_ON
        CLR     LED_OFF


        SETB    C_D          ;   COMANDO
        MOV     R7,#20H      ;  Clear Data Memory
```

```
        CALL    ENVIO_DATO

        MOV     R3,#01       ; Ponemos PASO a 1

        POP     ACC
        MOV     R0,A
        POP     ACC
        MOV     PSW,A
        POP     ACC
        MOV     IE,#85H   ; Habilito todas las interrupciones
        RETI
;  ********************************************************************
;
;  NOMBRE:    ENVIO_DATO
;
;  FUNCIàN:  Realiza el RESET del LCD-Driver
;
;  LLAMADA:  Al inicio del programa principal
;  ********************************************************************
ENVIO_DATO:                  ; Rutina para el envio de datos Serie

        SETB    SCKI
        CLR     CS           ; Todas las l¡neas preparadas
                             ; CS=0

WAIT1:  JNB     BUSY,WAIT1   ; Esperamos el ACK del LCD-Driver

        MOV    A,  R7        ; Cargamos el dato en el Acumulador
        MOV    R4, #8        ; Realizamos un bucle de 8 para enviar
                             ;  los datos
BUCLE:  CLR     SCKI         ; Bajamos la SCKI
        RLC     A            ; Desplazamos el primer bit
        MOV     SI, C        ; Dato al SI
        NOP
        SETB    SCKI         ; Subimos el reloj
        NOP
        DJNZ    R4, BUCLE    ; Saltamos al principio de BUCLE

WAIT2:  JNB     BUSY,WAIT2   ; Esperamos el ACK del LCD-Driver

        SETB    CS               ; Levantamos CS=1


        RETI

;  ********************************************************************
;
;  NOMBRE:    ENVIO_DATOS
;
;  FUNCIàN:  Envia datos al LCD, la diferencia con ENVIO_DATO es que no
;            baja y sube la se¤al de CS
;
;  LLAMADA:  Cuando se quieran enviar varios datos sin variar CS
;  ********************************************************************

ENVIO_DATOS:                  ; Rutina para el envio de datos Serie
        SETB    SCKI

WAIT11:  JNB     BUSY,WAIT11  ; Esperamos el ACK del LCD-Driver

        MOV    A,  R7        ; Cargamos el dato en el Acumulador
        MOV    R4, #8        ; Realizamos un bucle de 8 para enviar
                             ;  los datos
BUCLEE: CLR     SCKI         ; Bajamos la SCKI
        RLC     A            ; Desplazamos el primer bit
        MOV     SI, C        ; Dato al SI
        NOP
        SETB    SCKI         ; Subimos el reloj
        NOP
        DJNZ    R4, BUCLEE   ; Saltamos al principio de BUCLEE

WAIT22: JNB     BUSY,WAIT22  ; Esperamos el ACK del LCD-Driver

        RETI

        END
```