

Bienvenido(a), **Visitante**. Por favor, [ingresa](#) o [regístrate](#).  

Abril 10, 2014, 03:29:26

[Inicio](#)[Ayuda](#)[Search](#)[Ingresar](#)[Registrarse](#)

Desarrolladores Freescale » Solo Freescale » HC08 » Manejo de display lcd

[« anterior](#) [próximo »](#)

Páginas: [1]

[IMPRIMIR](#)

Autor

Tema: Manejo de display lcd (Leído 1020 veces)

**artu\_santacru:**

Mensajes: 1

**Manejo de display lcd**

« : Agosto 03, 2011, 11:19:55 »

Aca les dejo una rutina que hice hace unos años para un display lcd

```

;=====
; ARCHIVO:      Inicializacion_LCD.asm
; PROPOSITO:    Inicializa LCD en el modo de 8 bits. Tambien se escribe un mensaje
;               de ejemplo y se enciende un led conectado a PTD3
;-----
--
;=====
; Interfaz con MC68HC908JK1/JK3
; Modulo LCD (2X16, 8 bits)
; Cristal externo de 4 MHz
; Conexiones hechas al PORTB
; PTB0-PTB7 = DB0-DB7 (Datos)
; PTD6 = EN (Enable)
; PTD7 = RS (Register Select)
; R/W = GND (Siempre escribo)
;=====

;=====
; Declaración de Igualdades
;=====

RamStart      equ    $0080          ;Punteros de memoria
RomStartJK3   equ    $EC00          ;Inicio de la FLASH (JK3)
RomStartJK1   equ    $F600          ;Inicio de la FLASH (JK1)
VectorStart   equ    $FFDE

$Include 'jl3regs.inc'              ;For the 68HC908JL3, 68HC908JK3, 68HC908JK1

DELAY4        equ    $0079          ;Constante de Retardo para 4.0000 MHz
(msDelay)
Mensajes      equ    $FBE3          ;Donde se almacenan los mensajes
RS            equ    7              ;1=Dato / 0=Control (LCD), bit 7 de PORTD

EN            equ    6              ;1=Habilita / 0=deshabilita (LCD), bit 6 de PORTD
ENT           equ    5              ;Pulsador de "ENTER", bit 5 de PORTD

```

```

SEL      equ    4      ;Pulsador de "SELECCION", bit 4 de PORTD
LED      equ    3      ;Salida para led, bit 3 de PORTD
IN_ADC   equ    2      ;Entrada del sensor, bit 2 de PORTD

```

```

;-----
;--

```

```

org RomStartJK3      ;Comienzo de la memoria FLASH

```

```

;=====
; Inicializacion de Registros y Borrado de RAM
;=====

```

INICIO

```

      mov    #$01,CONFIG1  ;Deshabilito el COP
      nop
      nop
      clr    PORTB
      clr    PORTD
      mov    #$FF,DDRB
      mov    #$C8,DDRD
      clra                   ;Limpio A
      clrx                   ;Limpio X
loop   clr    RamStart,x    ;Me muevo en la RAM y borro
      incx                   ;Incremento X de $0 a $80
      txa
      cmp    #$80           ;Borre toda la RAM? ($80 + $80 = $100)
      bne    loop
      bset   LED,PORTD      ;Enciendo led

```

```

;-----
;--

```

```

;=====
; Programa Principal
;=====

```

```

      jsr    LCD_Ini        ;Inicializo LCD
      jsr    Mensaje_1      ;Escribo 1º fila del LCD
      jsr    Mensaje_2      ;Escribo 2º fila del LCD
      bra    *              ;Quedo en un bucle

```

```

;-----
;--

```

```

;=====
; Subrutinas de Mensajes
;=====

```

Mensaje\_1

```

      lda    #$85           ;Direccion DDRAM=$04 (1º fila)
      jsr    LCD_Ctrl       ;Envío al display
      jsr    usDelay        ;Espero 40 microsegundos
      clrx
Sig_Ca01   lda    msg01,x    ;Cargo A con cada caracter del 1 mensaje
      beq    outmsg_1       ;Si es igual a 0 finalizo el 1 mensaje
      jsr    LCD_Escr       ;Escribo
      incx
      bra    Sig_Ca01       ;Paso al siguiente caracter

```

```

outmsg_1   rtc

```

```
outmsg_1      rts
```

```
Mensaje_2
```

```
    lda    #$C5          ;Direccion DDRAM=$40 (2 fila)
    jsr    LCD_Ctrl      ;Envio al display
    jsr    usDelay       ;Espero 40 microsegundos
    clrx
```

```
Sig_Ca02      lda    msg02,x      ;Carga A con cada caracter del 2 mensaje
              beq    outmsg_2     ;Si es igual a 0 finalizo el 2 mensaje
              jsr    LCD_Escr     ;Escribo
              incx
              bra    Sig_Ca02     ;Paso al siguiente caracter
```

```
outmsg_2      rts
```

```
;-
--
```

```
;-=====
; Subrutina de Inicialización del LCD
;=====
```

```
LCD_Ini
```

```
    idx    #$0F          ;Cargo retardo
    jsr    msDelay       ;Espero 15 milisegundos
    lda    #$38          ;Inicializo (8 bits, 2 líneas y 5x7 dots)
    jsr    LCD_Ctrl      ;Subrutina de Control
    lda    #$0C          ;Display ON, cursor OFF, parpadeo OFF
    jsr    LCD_Ctrl      ;Subrutina de Control
    lda    #$06          ;(I/D=1) incrementa la dirección de DDRAM, (S=0) no
se desplaza el display
    jsr    LCD_Ctrl      ;Subrutina de Control
    lda    #$01          ;Limpio display
    jsr    LCD_Ctrl      ;Subrutina de Control
    rts                ;Retorno al programa principal
```

```
;-
--
```

```
;-=====
; Subrutina de Control del LCD
;=====
```

```
LCD_Ctrl
```

```
    sta    PORTB
    bclr   RS,PORTD      ;RS=0 es un comando de control
    bset   EN,PORTD
    bclr   EN,PORTD      ;Habilito y deshabilito LCD
    idx    #$04          ;Espero 4 milisegundos
    jsr    msDelay
    rts
```

```
;-
--
```

```
;-=====
; Subrutina de Escritura del LCD
;=====
```

```
LCD_Escr
```

```

-----
sta    PORTB
bset   RS,PORTD      ;RS=1 es un dato
bset   EN,PORTD
bclr   EN,PORTD      ;Habilito y deshabilito LCD
bsr    usDelay        ;Demora de 40 microsegundos
rts

;-----
;-----

;=====
; Subrutina de Retardo de "X" milisegundos
; Ciclos = 7+(H:X)*[121*(2+3+3)+25]+4
;=====

msDelay
    pshx                ;[2] Salva X en la pila
    pshh                ;[2] Salva H en la pila
    ldhx    #DELAY4      ;[3] Carga constante de bucle fino
Delay0009.B    aix    #-1    ;[2] Decrementa H:X en 1
    cphx    #0            ;[3] Llegó a cero (0)
    bne     Delay0009.B    ;[3] Si no es igual, salta a Delay0
    pulh                    ;[2] Si es igual, recupera H de la pila
    pulx                    ;[2] Recupera X de la pila
    aix    #-1            ;[2] Decrementa H:X en 1
    cphx    #0            ;[3] Llegó a cero?
    bne     msDelay        ;[3] Si no es igual, salta a Delay
    rts                    ;[4] Retorna al programa principal

;-----
;-----

;=====
; Subrutina de Retardo de 40 Microsegundos
; Ciclos = 6+(A*3)
;=====

usDelay
    lda    #$28          ;[2] Carga retardo
    dbnza    *            ;[3] Decrementa hasta 0
    rts                    ;[4] Retorna al programa principal

;-----
;-----

;=====
; Mensajes Almacenados
;=====

org Mensajes

msg01    db    'ARTURO'
          db    0

msg02    db    'SANTA CRUZ'
          db    0

;-----

```

```

--

;=====
; Subrutina que Atiende a las Interrupciones Involuntarias
;=====

Falsa_Int
    rti                ;Retorna al programa

;-----
--

;=====
; Mapa de Vectores
;=====

    org VectorStart

    dw Falsa_Int        ;Fin de conversion del ADC
    dw Falsa_Int        ;Interrupcion por teclado
    dw Falsa_Int        ;No usado ($FFE2-$FFE3)
    dw Falsa_Int        ;No usado ($FFE4-$FFE5)
    dw Falsa_Int        ;No usado ($FFE6-$FFE7)
    dw Falsa_Int        ;No usado ($FFE8-$FFE9)
    dw Falsa_Int        ;No usado ($FFEA-$FFEB)
    dw Falsa_Int        ;No usado ($FFEC-$FFED)
    dw Falsa_Int        ;No usado ($FFEE-$FFEF)
    dw Falsa_Int        ;No usado ($FFF0-$FFF1)
    dw Falsa_Int        ;Vector de Overflow del TIM
    dw Falsa_Int        ;Vector del canal 1 del TIM
    dw Falsa_Int        ;Vector del canal 0 del TIM
    dw Falsa_Int        ;No usado ($FFF8-$FFF9)
    dw Falsa_Int        ;Interrupcion por IRQ
    dw Falsa_Int        ;Interrupcion por SWI
    dw INICIO           ;Vector de Reset

    End                ;Fin del Programa

;-----
--

```

 En línea
**cupajuti**

Mensajes: 10

**Re: Manejo de display lcd**

« Respuesta #1 : Noviembre 03, 2011, 12:42:47 »

Qué bueno ver algo de ensablador! El "C" me aburre...

 En línea
**monini64**

Mensajes: 3

**Re: Manejo de display lcd**

« Respuesta #2 : Noviembre 05, 2011, 01:27:27 »

gracias por el programa

Páginas: [1]

**IMPRIMIR**[« anterior](#) [próximo »](#)

Ir a: =&gt; HC08 ▼

Powered by SMF 1.1.16 | SMF © 2006-2008, Simple Machines  
OrangeLine by **Crip**