

COMENTARIO TÉCNICO

Buceando en los MCUs Freescale.....



Por Ing. Daniel Di Lella
Dedicated Field Application Engineer
EDUDEVICES

www.edudevices.com.ar
dilella@arnet.com.ar



www.edudevices.com.ar

“Medidor de Potencia Activa Monofásico y Trifásico”

Por Ing. Aranda, Roberto Carlos / Ing. Teseyra, Rene Julio / Lutfi David,
Ortiz Rodrigo – FMA - UCSE

4ta. y última Parte.

Finalmente, se describirán las últimas sub rutinas de esta muy interesante aplicación que solo tiene por objeto “disparar” ideas a los diseñadores de sistemas donde se necesite medir potencias activas sin la necesidad de complicar el diseño.

* -----
* Escritura de comandos en LCD, usando modo 8 bits
* El Acc debe contener el comando que se desea enviar
* -----

LCD8Ctrl:

AND #\$F0	;Mask out lower data bits
PSHA	;Store value to send temporarily
LDA PORTA	;Don't change other PORTA pins
AND #\$0F	;Mask out UPPER nibble
ADD 1,SP	;Add value to send
STA PORTA	;Store to PORTA
BSET E_LCD,PORTA	;Toggle enable line
BCLR E_LCD,PORTA	
JSR _40usDelay	;40us setup time
PULA	;Deallocate stack data
RTS	

* -----
 * Escritura de comandos en LCD, usando modo 4 bits
 * El Acc debe contener el comando que se desea enviar
 * -----

LCDCtrl:

```

;----- Nibble Alto -----
    PSHA                ;Store data on stack
    AND #$F0           ;Mask out lower nibble
    PSHA                ;Store upper nibble on stack
    LDA PORTA          ;Load PORTA contents
    AND #$0F           ;Mask out UPPER nibble
    ADD 1,SP           ;Add the data nibble
    STA PORTA          ;Present upper nibble to LCD
    BSET E_LCD,PORTA   ;Toggle Enable line
    BCLR E_LCD,PORTA
    JSR _40usDelay     ;40us setup time
;----- Nibble Bajo -----
    PULA                ;Deallocate last temp storage
    PULA                ;Get original data byte
    AND #$0F           ;Mask out upper nibble
    NSA                ;Put LOWER nibble in UPPER nibble
    PSHA                ;Store onto stack
    LDA PORTA          ;Get existing PORTA data
    AND #$0F           ;Mask out lower nibble
    ADD 1,SP           ;Add lower nibble of data byte
    STA PORTA          ;Store to PORTA
    BSET E_LCD,PORTA   ;Toggle Enable line
    BCLR E_LCD,PORTA
    PULA                ;Deallocate temp storage
    CMP #$10           ;Longer delay for commands 1 or 2
    BEQ LCLonger
    CMP #$20
    BEQ LCLonger
    JSR _40usDelay     ;40us for any other command
    RTS ;Return
  
```

LCLonger:

```

    LDA #2
    JSR msDelay
    RTS
  
```

* -----
 * Escritura de datos en LCD, usando modo 4 bits
 * El Acc debe contener el dato que se desea enviar
 * -----

LCDData:

```

;----- Nibble Alto -----
    PSHA                ;Store data on stack temporarily
    AND #$F0           ;Mask out lower nibble
    PSHA                ;Store upper nibble on stack
    LDA PORTA          ;Load PORTA contents
    AND #$0F           ;Mask out UPPER nibble
    ADD 1,SP           ;Add the data nibble
    STA PORTA          ;Store to PORTA
    BSET RS_LCD,PORTA  ;Set RS for control
    BSET E_LCD,PORTA   ;Toggle Enable line
    BCLR E_LCD,PORTA
;----- Nibble Bajo -----
    PULA                ;Deallocate temp storage
    PULA                ;Get original data
    AND #$0F           ;Mask out upper nibble
  
```

```

NSA ;Put LOWER nibble in UPPER port pins
PSHA ;Store onto stack
LDA PORTA
AND #$0F
ADD 1,SP
STA PORTA
BSET E_LCD,PORTA ;Toggle enable line
BCLR E_LCD,PORTA
JSR _40usDelay ;40us setup time
JSR _40usDelay ;40us setup time
BCLR RS_LCD,PORTA ;Clear RS for data
JSR _40usDelay ;40us setup time
JSR _40usDelay ;40us setup time
PULA ;Deallocate temp storage
RTS

```

* -----

* Borrado del LCD

* -----

LCDClear:

```

LDA #$01 ;Borra LCD, Cursor en posicion 00
JSR LCDCtrl
RTS

```

LCDHome LDA #\$02

```

JSR LCDCtrl
RTS

```

* -----

* Rutina que muestra la porcion actual de la cadena de caracteres

* en el LCD. Cuando es llamada, el indice contiene el

* offset del caracter deseado.

* -----

ShowString:

```

JSR LCDClear ;Borra el LCD
CLR Count ;Borra la variable contador
LDA #$80 ;Direccion de la linea 1
JSR LCDCtrl

```

NextByte:

```

LDA ,X ;Carga el byte ASCII del caracter a mostrar
CMP #EOT ;Verifica si es EOT, ultimo caracter (Rellena con espacios)
BEQ Padding
JSR LCDData ;Envia el dato al LCD
AIX #1 ;Incrementa el indice
INC Count ;Incrementa el contador
LDA Count ;Compara el contador
CMP #MAXCHARS ;con la Maxima cantidad de caracteres a mostrar (Sale)
BEQ Done
CMP #MAXLINE ;y con la Maxima cantidad de caracteres a mostrar por linea (Va a linea 2)
BNE SSCont
LDA #$C0 ;Direccion de la linea 2
JSR LCDCtrl
SSCont BRA NextByte ;Prepara el proximo byte

```

Padding:

```

LDA Count ;Compara el contador
CMP #$00 ;Verifica si la cadena de caracteres a rebasado el LCD (Reset)?
BEQ Reset
CMP #MAXCHARS ;y con la Maxima cantidad de caracteres a mostrar (Sale)
BEQ Done
INC Count ;Incrementa el contador

```

```

        JSR BlankSpace          ;Coloca un espacio en la posicion actual del LCD
        BRA Padding            ;Repite
Reset   JSR BlankSpace        ;Show a final space in first position
        LDHX MsgStart         ;Load start of message index
        AIX #-1               ;Compensate for INCX in UpdateLCD after RTS
        STHX MsgIndex         ;Record new message index
Done    RTS                    ;Sale

```

```

* -----
* Envia un Caracter ASCII de espacio al LCD
* -----

```

```

BlankSpace:
        LDA #$20
        JSR LCDData
        RTS

```

```

* -----
* Scroll subroutines
* -----
* Initialize the message variables for the desired output string
* Register A contains the offset of desired message.
* -----

```

```

LoadMsg STHX MsgIndex          ;Setup the message index
        STHX MsgStart         ;Store the start of the message
        RTS                    ;Return

```

```

* -----
* Update the LCD with current portion of string to be displayed
* -----

```

```

UpdateLCD:
        LDHX MsgIndex         ;Start at current index into message
        JSR ShowString        ;Show current portion of string
        LDHX MsgIndex
        CPHX MsgStart
        BNE ULgo
        LDA #$30
        JSR msDelay
ULgo    AIX #1
        STHX MsgIndex         ;Increment the index
        RTS                    ;Return

```

```

*****
* No_Int - Si alguna interrupción no esperada ingresa sale con RTI
*****

```

```

No_Int:
        RTI

```

```

* -----
* DEFINICION DE VECTORES DEL SISTEMA
* -----

```

```

        ORG TBVEC
        FDB No_Int
        ORG ADCVEC
        FDB No_Int          ; ADC Conversión Completa
        ORG KBIVEC
        FDB No_Int
        ORG SCITXVEC
        FDB No_Int
        ORG SCIRXVEC
        FDB RX_ISR          ; Recepción SCI

```

```
ORG SCIERVEC
FDB No_Int
ORG SPITXVEC
FDB No_Int
ORG SPIRXVEC
FDB No_Int
ORG T2OFVEC
FDB No_Int
ORG T2CH1VEC
FDB No_Int
ORG T2CH0VEC
FDB No_Int
ORG T1OFVEC
FDB No_Int
ORG T1CH1VEC
FDB No_Int
ORG T1CH0VEC
FDB No_Int
ORG PLLVEC
FDB No_Int
ORG IRQ1VEC
FDB No_Int
ORG SWIVEC
FDB No_Int
ORG RESETVEC           ;Reset
FDB Inicio
```

Fin ;!

Nota de Redacción: El lector puede descargar este artículo y artículos anteriores de “*Buceando...*” desde la sección “*Artículos Técnicos*” en el sitio web de **EduDevices** (www.edudevices.com.ar)



WWW.EDUDEVICES.COM.AR