

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

3era. Parte.

Continuando con el artículo anterior, nos toca describir las distintas sub – rutinas que componen el programa y que son llamadas desde el lazo principal según sea necesario.

```
* -----
* RUTINAS DE TRANSMISION SERIE:
* Init_SCI - RX_ISR
* -----
*****
* Recibe un byte y lo compara con 'T'.
* Si no coincide lo ignora y vuelve.
* Si coincide va a la subrutina TX_Dato
*****
RX_ISR:
    SEI                ; DesHabilitacion de interrupciones
    LDA    SCS1        ; lee el SCS1 para borrar el flag
    LDA    SCDR        ; lee el dato que ingresó
    CMP    #'T'        ; lo compara con 'T'
    BNE    Sale_RX     ; si no coincide ignoro el llamado
    MOV    #$01,Bandera_TX ; si coincide pone en 1 la bandera de transmicion
Sale_RX:
    CLI                ; Habilitacion de interrupciones
    RTI                ; sale de la rutina de interrupción
```

* Transmite los datos adquiridos de P1,P2,P3 y PT

TX_Dato:

```
MOV    #' ',SCDR
BRCLR  7,SCS1,*    ;Espera para TX
MOV    #' ',SCDR
BRCLR  7,SCS1,*    ;Espera para TX
LDA    P_umil
ADD    #$30
STA    SCDR
BRCLR  7,SCS1,*    ;Espera para TX
LDA    P_centena
ADD    #$30
STA    SCDR
BRCLR  7,SCS1,*    ;Espera para TX
LDA    P_decena
ADD    #$30
STA    SCDR
BRCLR  7,SCS1,*    ;Espera para TX
LDA    P_unidad
ADD    #$30
STA    SCDR
BRCLR  7,SCS1,*    ;Espera para TX
MOV    #'W',SCDR
BRCLR  7,SCS1,*    ;Espera para TX
MOV    #LF,SCDR
BRCLR  7,SCS1,*    ;Espera para TX
MOV    #CR,SCDR
BRCLR  7,SCS1,*    ;Espera para TX
RTS
```

* -----

* RUTINAS DEL ADQUISICION Y TRANSFORMACION DE DATOS PARA SU VISUALIZACION:

* Clr_Datos_RAM - Adc_Adquisicion - Adc_Promedio - Adc_Escalado - Adc_Conversion

* -----

* Borra Datos en la RAM

Clr_Datos_RAM:

```
ldhx  #Datos_P1          ;Inicio de Datos P1
C_P1  clr  ,x             ;Borro el byte apuntado por X
aix   #1                 ;Incremento X
cphx  #Datos_P1+L_Datos  ;Comparo x con final del bloque
blo   C_P1               ;Si no terminé borro otro Dato P1
```

```
ldhx  #Datos_P2          ;Inicio de Datos P2
C_P2  clr  ,x             ;Borro el byte apuntado por X
aix   #1                 ;Incremento X
cphx  #Datos_P2+L_Datos  ;Comparo x con final del bloque
blo   C_P2               ;Si no terminé borro otro Dato P2
```

```
ldhx  #Datos_P3          ;Inicio de Datos P3
C_P3  clr  ,x             ;Borro el byte apuntado por X
aix   #1                 ;incremento X
cphx  #Datos_P3+L_Datos  ;Comparo x con final del bloque
blo   C_P3               ;si no terminé borro otro
rts
```

* Adquisicion con el conversor AD, cada dato se guarda en

* una posicion de la RAM de L_Datos.

Adc_Adquisicion:

```
LDA    #%00100000
STA    ADSCR                ;Enciende el ADC canal 0
BRCLR  COCO,ADSCR,*
LDA    ADR
BRCLR  COCO,ADSCR,*
MOV    #%00011111,ADSCR    ;Apaga el ADC

LDHX   #Datos_P1          ;Apunto al Dato P1
CLRA
A_P1_1 CMP  Puntero        ;Comparo el acumulador con Puntero
BEQ    A_P1_2              ; y salto si son iguales
AIX    #1
INCA
BRA    A_P1_1
A_P1_2 MOV  ADR,X+         ;Almaceno el dato del ADC

LDA    #%00100001
STA    ADSCR                ;Enciende el ADC canal 1
BRCLR  COCO,ADSCR,*
LDA    ADR
BRCLR  COCO,ADSCR,*
MOV    #%00011111,ADSCR    ;Apaga el ADC

LDHX   #Datos_P2          ;Apunto al Dato P2
CLRA
A_P2_1 CMP  Puntero        ;Comparo el acumulador con Puntero
BEQ    A_P2_2              ; y salto si son iguales
AIX    #1
INCA
BRA    A_P2_1
A_P2_2 MOV  ADR,X+         ;Almaceno el dato del ADC

LDA    #%00100010
STA    ADSCR                ;Enciende el ADC canal 2
BRCLR  COCO,ADSCR,*
LDA    ADR
BRCLR  COCO,ADSCR,*
MOV    #%00011111,ADSCR    ;Apaga el ADC

LDHX   #Datos_P3          ;Apunto al Dato P3
CLRA
A_P3_1 CMP  Puntero        ;Comparo el acumulador con Puntero
BEQ    A_P3_2              ; y salto si son iguales
AIX    #1
INCA
BRA    A_P3_1
A_P3_2 MOV  ADR,X+         ;Almaceno el dato del ADC

RTS
```

* Promedio de L_Datos adquiridos por el conversor AD

Adc_Promedio:

```
CLR Suma_LSB
CLR Suma_MSB
MOV #L_Datos,Puntero ; Cargo Puntero con L_Datos
Mas LDA ,X ; Pone el Dato en el A
ADD Suma_LSB ; Hace una suma acumulativa
STA Suma_LSB ;
BCS Acarreo ; ¿hay acarreo?
BRA Proximo ; -no- vuelvo a sumar otro dato
Acarreo:
INC Suma_MSB ; -si- sumo 1 a MSB y vuelvo a sumar otro dato
Proximo:
AIX #1 ; Incremento X
DBNZ Puntero,Mas ; Decremento el puntero
; si no terminó suma otro dato más
LDA Suma_MSB
PSHA
PULH ; H = MSB de la suma
LDA Suma_LSB ; A = LSB de la suma
LDX #L_Datos ; X = divisor nro de datos
DIV ; divide suma total/nro datos H:A / X
RTS ; vuelve de subrutina
```

* Escalado del dato adquirido por el ADC

Adc_Escalado:

```
SUB #$80 ;Le resta 80
CMP #$00
BLE Cero
SUB #$0B ;y luego le resta 0B (total 8B)
CMP #$00
BLE Cero
LDX #$42
MUL ;Lo multiplica por 66 (X:A <- A x 66)
RTS
Cero: LDX #$00
LDA #$00
RTS
```

* Conversion del dato adquirido

Adc_Conversion:

```
PSHX
PULH
LDX #$64 ; X = 100
DIV ; A = H:A / 100
PSHH ; H = Primer Resto lo empujo a la pila
CLRH
LDX #$0A ; X = 10
DIV ; A = H:A / 10
STA P_umil
PSHH ; H = Segundo Resto lo empujo a pila
```

```

PULA                                ; Recupero segundo resto en A
STA  P_centena

PULA                                ; Recupero Primer resto en A
CLRH
LDX  #$0A                          ; X = 10
DIV  ; A = H:A / 10
STA  P_decena

PSHH                                ; H = Tercer Resto lo empujo a pila
PULA                                ; Recupero Tercer resto en A
STA  P_unidad
RTS

```

```

* -----
* RUTINAS DEL CALCULO DE POTENCIA TOTAL y POTENCIA PICO:
* Borrado_PT - Borrado_PP - Sumar_PT
* -----

```

```

*****

```

```

* Borrado de PT
*****

```

```

Borrado_PT:
  CLR  PT_umil
  CLR  PT_centena
  CLR  PT_decena
  CLR  PT_unidad
  RTS

```

```

*****

```

```

* Borrado de PP
*****

```

```

Borrado_PP:
  CLR  PP_umil
  CLR  PP_centena
  CLR  PP_decena
  CLR  PP_unidad
  RTS

```

```

*****

```

```

* Calculo de PT sumando los valores de P1, P2 y P3
*****

```

```

Sumar_PT:
  LDA  P_unidad
  ADD  PT_unidad
  CLRH
  LDX  #$0A                          ; X = 10
  DIV  ; A = H:A / 10
  ADD  PT_decena
  STA  PT_decena
  PSHH                                ; H = Resto lo empujo a pila
  PULA                                ; Recupero resto en A
  STA  PT_unidad

  LDA  P_decena
  ADD  PT_decena
  CLRH
  LDX  #$0A                          ; X = 10
  DIV  ; A = H:A / 10
  ADD  PT_centena

```

```
STA PT_centena
PSHH ; H = Resto lo empujo a pila
PULA ; Recupero resto en A
STA PT_decena
```

```
LDA P_centena
ADD PT_centena
CLRH
LDX #$0A ; X = 10
DIV ; A = H:A / 10
ADD PT_umil
STA PT_umil
PSHH ; H = Resto lo empujo a pila
PULA ; Recupero resto en A
STA PT_centena
```

```
LDA P_umil
ADD PT_umil
STA PT_umil
RTS
```

* Subrutina para mostrar los datos adquiridos en el LCD

Adc_MostrarP1:

```
LDA Posicion
CMP #1
BNE Sale_Mostrar
BRA Linea1
```

Adc_MostrarP2:

```
LDA Posicion
CMP #1
BNE SaleP2
BRA Linea2
SaleP2 LDA Posicion
CMP #2
BNE Sale_Mostrar
BRA Linea1
```

Adc_MostrarP3:

```
LDA Posicion
CMP #2
BNE SaleP3
BRA Linea2
SaleP3 LDA Posicion
CMP #3
BNE Sale_Mostrar
BRA Linea1
```

Adc_MostrarPT:

```
LDA Posicion
CMP #3
BNE SalePT
BRA Linea2
SalePT LDA Posicion
CMP #4
BNE Sale_Mostrar
BRA Linea1
```

Adc_MostrarPP:
LDA Posicion
CMP #5
BNE SalePP
BRA Linea2
SalePP LDA Posicion
CMP #6
BNE Sale_Mostrar
BRA Linea1

Sale_Mostrar:
RTS

Linea1:
LDA #\$89 ;Direccion de unidad de mil
JSR LCDCtrl
LDA P_umil
ADD #\$30
JSR LCDData ;Envia el dato al LCD
LDA P_centena
ADD #\$30
JSR LCDData ;Envia el dato al LCD
LDA P_decena
ADD #\$30
JSR LCDData ;Envia el dato al LCD
LDA P_unidad
ADD #\$30
JSR LCDData ;Envia el dato al LCD
BRA Sale_Mostrar

Linea2:
LDA #\$C9 ;Direccion de unidad de mil
JSR LCDCtrl
LDA P_umil
ADD #\$30
JSR LCDData ;Envia el dato al LCD
LDA P_centena
ADD #\$30
JSR LCDData ;Envia el dato al LCD
LDA P_decena
ADD #\$30
JSR LCDData ;Envia el dato al LCD
LDA P_unidad
ADD #\$30
JSR LCDData ;Envia el dato al LCD
BRA Sale_Mostrar

* Subrutina de atención a tecla

Atencion_Tecla:
LDA #\$14 ;Retardo 20ms
JSR msDelay
LDA Tecla
CBEQA #1,Atencion_Tecla1
CBEQA #2,Atencion_Tecla2
CBEQA #3,Atencion_Tecla3
BRA Sale_Teclas

Atencion_Tecla1:
BRSET 1,PORTC,Sale_Teclas ;Se solto tecla 1?
LDA Posicion ;Verifica Posicion
CMP #1
BEQ Sale_Teclas
CMP #6
BEQ Sale_Teclas
DEC Posicion ;Decrementa Posicion
BRA Mostrar

Atencion_Tecla2:
BRSET 0,PORTC,Sale_Teclas ;Se solto tecla 2?
LDA Posicion ;Verifica Posicion
CMP #5
BEQ Sale_Teclas
CMP #6
BEQ Sale_Teclas
INC Posicion ;Incrementa Posicion
BRA Mostrar

Atencion_Tecla3:
BRSET 3,PORTC,Sale_Teclas ;Se solto tecla 3?
LDA Posicion ;Verifica Posicion
CMP #5
BEQ Mostrar5
CMP #6
BEQ Mostrar6
BRA Sale_Teclas

Mostrar5:
INC Posicion
BRA Mostrar

Mostrar6:
DEC Posicion

Mostrar:
LDA Posicion
CBEQA #1,Cartel1
CBEQA #2,Cartel2
CBEQA #3,Cartel3
CBEQA #4,Cartel4
CBEQA #5,Cartel5
CBEQA #6,Cartel6
MOV #1,Posicion ;Inicializo Posicion
BRA ASuelta

Cartel1 LDHX #Cartel_1
BRA ASuelta

Cartel2 LDHX #Cartel_2
BRA ASuelta

Cartel3 LDHX #Cartel_3
BRA ASuelta

Cartel4 LDHX #Cartel_4
BRA ASuelta

Cartel5 LDHX #Cartel_5
BRA ASuelta

Cartel6 LDHX #Cartel_6
BRA ASuelta

ASuelta JSR ShowString

Suelta LDA #\$14 ;Retardo 20ms
JSR msDelay
BRCLR 1,PORTC,Suelta ;Sigue presionada tecla 1?
BRCLR 0,PORTC,Suelta ;Sigue presionada tecla 2?
BRCLR 3,PORTC,Suelta ;Sigue presionada tecla 3?

Sale_Teclas:
RTS

* -----
* CARTELES
* -----

;----- Presentación -----

CartelPres:
FCB " MEDIDOR DE POTENCIA ACTIVA "
FCB EOT
FCB " MONO/TRIFASICO ELECTROLAB UCSE"
FCB EOT
FCB FINPRES

Cartel_1:
FCB "Pfase1= 0000W Pfase2= 0000W "
FCB EOT

Cartel_2:
FCB "Pfase2= 0000W Pfase3= 0000W "
FCB EOT

Cartel_3:
FCB "Pfase3= 0000W PTotal= 0000W "
FCB EOT

Cartel_4:
FCB "PTotal= 0000W ConsA= 0000W/H "
FCB EOT

Cartel_5:
FCB "ConsA= 0000W/H PtPico= 0000W "
FCB EOT

Cartel_6:
FCB "PtPico= 0000W 07/11/06 12:10"
FCB EOT

* -----
* RUTINAS DE RETARDO
* -----

* cycles = 5 + X(6+7+1205) + 3 + 6
* cycles = 14 + X(1218)
* where X is value loaded into Acc
* Causes 1ms delay for BUSFREQ values of (1-8 integer values)
* -----

_1msDelay:
PSHA ;2 cycles
LDA #BUSFREQ ;2
INCA ;1
DLLoop DBNZA DLSub ;3
BRA DLDone ;3

```

DLSub  MOV #$F1,Counter      ;4
        DBNZ Counter,*       ;5
        BRA DLLoop           ;3
DLDone PULA                  ;2
        RTS                  ;4

```

```

* -----
* Retardo en ms
* El Acc debe contener el número de ms que se desea esperar
* -----

```

```

msDelay JSR _1msDelay
        DBNZ msDelay
        RTS

```

```

* -----
* Rutina de retardo de 40us:
* Esta rutina es importante para el modulo LCD, el cual requiere retardos
* de 40us para muchos de sus comandos, incluso para la carga de datos.
* Cantidad de Ciclos = 8+X(3) = 8+14(3) = 50
* Donde X = BUSFREQ*USCOUNT = USDELAY = 14 = $0E
* Para un Tiempo de Ciclo de 813,8nS se necesitan 50 Ciclos para completar
* los 40uS aproximadamente.
* -----

```

```

_40usDelay:
        LDA #BUSFREQ*USCOUNT      ;2
        DBNZ *                      ;3
        NOP                          ;2
        RTS                          ;4

```

```

* -----
* Retardo de 1 Segundo
* FA = 250mS X 4 = 1S Aprox
* -----

```

```

_1sDelay:
        PSHA
        LDA #$FA
        JSR msDelay
        LDA #$FA
        JSR msDelay
        LDA #$FA
        JSR msDelay
        LDA #$FA
        JSR msDelay
        PULA
        RTS

```

```

* -----
* RUTINAS DEL LCD:
* LCDInit - LCDCtrl - LCDData - LCDClear
* -----

```

```

* Inicializacion del LCD
* -----

```

```

LCDInit:
;----- Operacion en 8 bits -----
        LDA #$0F                      ;Retardo 15ms
        JSR msDelay
        LDA #$30                      ;Function set
        JSR LCD8Ctrl
        LDA #$05                      ;Retardo 5ms

```

```

JSR msDelay
LDA #$30 ;Function set
JSR LCD8Ctrl
LDA #$01 ;Retardo 1ms
JSR msDelay
LDA #$30 ;Function set
JSR LCD8Ctrl
LDA #$20 ;Funcion configuracion de 4-bit
JSR LCD8Ctrl
;----- Operacion en 4 bits -----
LDA #$28 ;Le indico al LCD que se trabaja con una interface
; de 4 bits y 2 lineas activadas de 5x7 pixeles

JSR LCDCtrl
LDA #$06 ;Autoincremento del Address Counter
JSR LCDCtrl
LDA #$01 ;Borra LCD, Cursor en posicion 00
JSR LCDCtrl
LDA #$0C ;LCD on, Cursor off, Blink off
JSR LCDCtrl
RTS

```

Continuará

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