

\*\*\*\*\*PROGRAMA\*\*\*\*\*

```

LIST          p=16F628A      ; Procesador utilizado
INCLUDE      "P16F628A.INC" ; Definición de los registros del Pic
__CONFIG    3F09            ; Configuración de fusibles

```

-----  
;Variables

```

CBLOCK 0x25
GRABAR_EEPROM y LEER_EEPROM      ; Variables para las rutinas
Addr_EE                          ; Almacena la dirección de la EEPROM a leer y/o grabar
Data_EE                          ; Almacena el dato a leer y/o grabar en la EEPROM
AUXIMPULSOS                      ; Variable para verificar la cantidad de impulsos
IMPULSOS                         ; Variable para impulsos de energía activa
PUNTO                            ; Variable para decimas de energía activa
UNIDADES                        ; Variable para unidades de energía activa
DECENAS                         ; Variable para decenas de energía activa
CENTENAS                        ; Variable para centenas de energía activa
UNIDADES_MIL                    ; Variable para unidades mil de energía activa
DECENAS_MIL                     ; Variable para decenas mil de energía activa
AYUDA_ENV_DATOS                 ; Variable de ayuda para enviar datos por RS232
ENDC

```

-----

```

#DEFINE ENTRADA_IMPULSOS  PORTB,7      ; Nombrar al pin 7 del puerto B como entrada de impulsos
#DEFINE NOLUZ_GRABAR     PORTB,6      ; Nombrar al pin 6 del puerto B como pin para sensar energía
#DEFINE LED               PORTA,0      ; Nombrar al pin 0 del puerto A como salida para un led

```

-----

```

ORG          0x00              ; Dirección de inicio del programa
goto        INICIO            ; Ir a la etiqueta Inicio
ORG          0x04              ; Dirección de Interrupciones
GOTO       INTERRUPCIONES     ; Ir a la etiqueta INTERRUPCIONES

```

-----

INICIO

```

BSF        STATUS,RP0        ; Cambio al BANCO 1
MOVLW     B'11011110'        ; Carga valor al registro de trabajo W
MOVWF     TRISA              ; Mueve el valor de W al registro TRISA
MOVLW     B'11111011'        ; Carga valor al registro de trabajo W
MOVWF     TRISB              ; Mueve el valor de W al registro
movlw     b'00100100'        ; Habilitar tx USART a 8 bits
movwf     TXSTA              ; Asincrona a alta velocidad
movlw     .25                 ; Velocidad de comuicación
movwf     SPBRG              ; a 9600 baudios con XT de 4Mhz
BCF        STATUS,RP0        ; Cambio al BANCO 0
MOVLW     d'7'               ; Carga el valor al registro de trabajo W
MOVWF     CMCON              ; Se apagan los comparadores
movlw     b'10010000'        ; Carga el valor al registro de trabajo W
movwf     RCSTA              ; Habilitar rx USART asincrona
clrf      RCREG              ; Limpiar registro RCREG
CLRF      Addr_EE            ; Limpia el registro de dirección de la EEPROM
CLRF      Data_EE           ; Limpia el registro de datos de la EEPROM
BCF        LED               ; Apagar Led
CALL      RET_50MS          ; Retardo de 50 ms

```

```

;-----
INI
    Movlw    .10                ; Carga el valor al registro de trabajo W
    movwf    Addr_EE            ; Mueve el valor de W a la variable Addr_EE
    CALL     LEER_EEPROM        ; Llama a la Subrutina leer eeprom
    MOVWF    Data_EE           ; El dato leído al variable Data_EE
    XORLW    .100              ; Es igual a 100..?
    BTFSS    STATUS,Z          ; Es igual a 100..?
    CALL     GRABA_DATOS_INI    ; No llama a la subrutina inicializar
    goto     ANOTAR_VARIABLES   ; Si, ir a la etiqueta ANOTAR_VARIABLES

;----- Subrutina para grabar datos -----
GRABAR
;#AUXIMPULSOS

    MOVLW    B'00001000'       ; Carga el valor al registro de trabajo W
    MOVWF    Addr_EE           ; Mueve el valor de W a la variable Addr_EE
    movfw    AUXIMPULSOS       ; Mueve el valor de la variable AUXIMPULSOS al registro W
    MOVWF    Data_EE           ; Mueve el valor de W a la variable Data_EE
    CALL     GRABAR_EEPROM     ; llama a la subrutina GRABAR_EEPROM

;#IMPULSOS

    MOVLW    B'00000111'       ; Carga el valor al registro de trabajo W
    MOVWF    Addr_EE           ; Mueve el valor de W a la variable Addr_EE
    movfw    IMPULSOS          ; Mueve el valor de la variable IMPULSOS al registro W
    MOVWF    Data_EE           ; Mueve el valor de W a la variable Data_EE
    CALL     GRABAR_EEPROM     ; llama a la subrutina GRABAR_EEPROM

;#PUNTO

    MOVLW    B'00000110'       ; Carga el valor al registro de trabajo W
    MOVWF    Addr_EE           ; Mueve el valor de W a la variable Addr_EE
    movfw    PUNTO             ; Mueve el valor de la variable PUNTO al registro W
    MOVWF    Data_EE           ; Mueve el valor de W a la variable Data_EE
    CALL     GRABAR_EEPROM     ; llama a la subrutina GRABAR_EEPROM

;#UNIDADES

    MOVLW    B'00000101'       ; Carga el valor al registro de trabajo W
    MOVWF    Addr_EE           ; Mueve el valor de W a la variable Addr_EE
    movfw    UNIDADES          ; Mueve el valor de la variable UNIDADES al registro W
    MOVWF    Data_EE           ; Mueve el valor de W a la variable Data_EE
    CALL     GRABAR_EEPROM     ; llama a la subrutina GRABAR_EEPROM

;#DECENAS

    MOVLW    B'00000100'       ; Carga el valor al registro de trabajo W
    MOVWF    Addr_EE           ; Mueve el valor de W a la variable Addr_EE
    movfw    DECENAS           ; Mueve el valor de la variable DECENAS al registro W
    MOVWF    Data_EE           ; Mueve el valor de W a la variable Data_EE
    CALL     GRABAR_EEPROM     ; llama a la subrutina GRABAR_EEPROM

;#CENTENAS

    MOVLW    B'00000011'       ; Carga el valor al registro de trabajo W
    MOVWF    Addr_EE           ; Mueve el valor de W a la variable Addr_EE
    movfw    CENTENAS          ; Mueve el valor de la variable CENTENAS al registro W
    MOVWF    Data_EE           ; Mueve el valor de W a la variable Data_EE
    CALL     GRABAR_EEPROM     ; llama a la subrutina GRABAR_EEPROM

;#UNIDADES_MIL

```

```

MOVW B'0000010'           ; Carga el valor al registro de trabajo W
MOVWF Addr_EE             ; Mueve el valor de W a la variable Addr_EE
movfw UNIDADES_MIL       ; Mueve el valor de la variable UNIDADES_MIL al registro
                          ; W
MOVWF Data_EE            ; Mueve el valor de W a la variable Data_EE
CALL GRABAR_EEPROM       ; llama a la subrutina GRABAR_EEPROM

;#DECENAS_MIL

MOVW B'0000001'           ; Carga el valor al registro de trabajo W
MOVWF Addr_EE             ; Mueve el valor de W a la variable Addr_EE
movfw DECENAS_MIL        ; Mueve el valor de la variable DECENAS_MIL al registro W
MOVWF Data_EE            ; Mueve el valor de W a la variable Data_EE
CALL GRABAR_EEPROM       ; llama a la subrutina GRABAR_EEPROM
RETURN                    ; Retornar

;----- Programa principal -----

PROGRAMA_PRINCIPAL

esperar_peticion

BSF LED                   ; Encender Led
BTFSC RCSTA,OERR          ; Comprobar ¿ existe error en el Registro RCSTA bit 1 ?
goto RESET1              ; Si, ir a la etiqueta RESET 1
BTFSC RCSTA,FERR          ; No, Comprobar ¿ existe error en el Registro RCSTA bit 2
                          ; trama completa ?
goto RESET2              ; Si, ir a la etiqueta RESET 2
movf RCREG,w              ; No, Mueve el valor del registro RCREG al registro W
"caracter necesario para enviar lecturas"
BTFSS STATUS,Z           ; ¿ Caracter recibido es igual al necesario ?
GOTO esperar_peticion    ; No, ir a etiqueta esperar_petición
goto ENVIAR_DATOS        ; Sí, ir a etiqueta ENVIAR_DATOS

RESET1

CLRF RCREG                ; Reseteo de registro RCREG
goto esperar_peticion     ; ir a etiqueta esperar_petición

RESET2

movf RCREG,w              ; Mueve el valor de registro de RCREG a registro W
goto esperar_peticion     ; ir a la etiqueta esperar_petición

;-----Subrutina para enviar datos-----

ENVIAR_DATOS

;-----Subrutina de aviso de 90.000 Kw/h -----

MOVFW DECENAS_MIL        ; Carga el valor de variable DECENAS MIL al registro de
                          ; trabajo W
XORLW .9                  ; Compara registro W con literal
BTFSS STATUS,Z           ; ¿es igual a 9 ?
GOTO continue            ; No, ir a la etiqueta continue
MOVW .43                  ; Si, Carga el valor al registro de trabajo W
movwf TXREG               ; Mueve en valor de registro W al registro TXREG
bsf STATUS,RPO           ; Cambio al BANCO 1

azcp

btfss TXSTA,TRMT         ; ¿ Se ha terminado de enviar ?
goto axcp                 ; No, ir a la etiqueta azcp
bcf STATUS,RPO           ; Si, cambio al BANCO 0

```

continue

```

MOVFW DECENAS_MIL           ; Mueve valor de la variable a registro W
MOVWF AYUDA_ENV_DATOS      ; Mueve valor de regsitro W a la variable
CALL  enviarRS232          ; llama a la subrutina RS232
MOVFW UNIDADES_MIL        ; Mueve valor de la variable a registro W
MOVWF AYUDA_ENV_DATOS      ; Mueve valor de regsitro W a la variable
CALL  enviarRS232          ; llama a la subrutina RS232
MOVFW CENTENAS             ; Mueve valor de la variable a registro W
MOVWF AYUDA_ENV_DATOS      ; Mueve valor de regsitro W a la variable
CALL  enviarRS232          ; llama a la subrutina RS232
MOVFW DECENAS              ; Mueve valor de la variable a registro W
MOVWF AYUDA_ENV_DATOS      ; Mueve valor de regsitro W a la variable
CALL  enviarRS232          ; llama a la subrutina RS232
MOVFW UNIDADES             ; Mueve valor de la variable a registro W
MOVWF AYUDA_ENV_DATOS      ; Mueve valor de regsitro W a la variable
CALL  enviarRS232          ; llama a la subrutina RS232
MOVFW PUNTO                ; Mueve valor de la variable a registro W
MOVWF AYUDA_ENV_DATOS      ; Mueve valor de regsitro W a la variable
CALL  enviarRS232          ; llama a la subrutina RS232
MOVWF IMPULSOS             ; Mueve valor de la variable a registro W
MOVWF AYUDA_ENV_DATOS      ; Mueve valor de regsitro W a la variable
CALL  enviarRS232          ; llama a la subrutina RS232
MOVFW AUXIMPULSOS          ; Mueve valor de la variable a registro W
MOVWF AYUDA_ENV_DATOS      ; Mueve valor de regsitro W a la variable
CALL  enviarRS232          ; llama a la subrutina RS232
MOVLW .42                  ; Cargar el valor al registro de trabajo W
movwf TXREG                 ; Mueve valor de regsitro de trabajo a TXREG
bsf   STATUS,RP0           ; Cambio al BANCO 1

```

EP

```

btfss TXSTA,TRMT           ; ¿Se ha terminado de enviar?
goto  EP                   ; No, ir a la etiqueta EP
BCF   STATUS,RP0           ; Si, Cambio al BANCO 0
call  EnviarSuministro     ; Llama a la etiqueta EnviarSuministro

```

;------Subrutina de solicitud de administración de Datos-----  
awi

```

BTFSC RCSTA,OERR           ; Comprobar ¿ existe error en el Registro RCSTA bit 1 ?
goto  RESET1               ; Si, ir a la etiqueta RESET 1
BTFSC RCSTA,FERR           ; No, Comprobar ¿ existe error en el Registro RCSTA bit 2;
                           ; trama completa ?
goto  RESET2               ; Si, ir a la etiqueta RESET 2
movf  RCREG,w              ; No, Mueve el valor del registro RCREG al registro W
"caracter necesario Administración de Datos"
BTFSS STATUS,Z             ; ¿ Caracter recibido es igual al necesario ?
GOTO  sigueawi            ; No, ir a la etiqueta sigueawi
GOTO  ADMINISTRAR_DATOS   ; Si, ir a la etiqueta ADMINISTRAR_DATOS

```

;------Subrutina necesaria para retornar al Programa principal-----

sigueawi

```

BTFSC RCSTA,OERR           ; Comprobar ¿ existe error en el Registro RCSTA bit 1 ?
goto  RESET1               ; Si, ir a la etiqueta RESET 1
BTFSC RCSTA,FERR           ; No, Comprobar ¿ existe error en el Registro RCSTA bit 2;
                           ; trama completa ?
goto  RESET2               ; Si, ir a la etiqueta RESET 2
movf  RCREG,w              ; No, Mueve el valor del registro RCREG al registro W
"caracter necesario para retornar al programa principal"
BTFSS STATUS,Z             ; ¿ Caracter recibido es igual al necesario ?

```

```

GOTO awi ; No, ir a la etiqueta awi
GOTO PROGRAMA_PRINCIPAL ; Si, ir a la etiqueta Programa principal

```

-----Subrutina enviar rs232-----

enviarRS232

```

MOV LW .48 ; Carga el valor al registro de trabajo W
ADDWF AYUDA_ENV_DATOS,W ; Suma valor de registro W con el valor de la variable
AYUDA_ENV_DATOS
movwf TXREG ; Mueve valor regsitro W a registro TXREG
bsf STATUS,RPO ; Cambio al BANCO 1

espera
btfss TXSTA,TRMT ; ¿Se ha terminado de envíar?
goto espera ; No, ir a etiqueta espera
BCF STATUS,RPO ; Si, Cambio al BANCO 0
return ; retornar

```

-----Subrutina para Cargar Valores-----

ANOTAR\_VARIABLES

```

movlw .0 ; Carga el valor al registro de trabajo W
movwf Addr_EE ; Mueve el valor del registro W a la variable Addr_EE

seg
MOV LW .1 ; Carga el valor al registro de trabajo W
ADDWF Addr_EE,W ; Suma valor de la variable Addr_EE con el valor del registroW
MOVWF Addr_EE ; Mueve el valor del regsitro de W a la variable Addr_EE
CALL LEER_EEPROM ; Llamar a la subrutina LEER_EEPROM
MOVFW Data_EE ; Mueve valor de variable Data_EE a registro W
CALL ASIGNAR ; Llamar a la subrutina ASIGNAR
movfw Addr_EE ; Mueve valor de variable Addr_EE a registro W
XORLW .8 ; Compara regsitro W con literal
BTFS STATUS,Z ; ¿es igual a 8?
GOTO seg ; No, ir a la etiqueta seg
GOTO final_annotar ; Si, ir a la etiqueta final_annotar

```

ASIGNAR

a\_AUXIMPULSOS

```

movfw Addr_EE ; Mueve valor de Addr_EE a registro W
XORLW .8 ; Compara regsitro W con literal
BTFS STATUS,Z ; ¿es igual a 8?
GOTO a_IMPULSOS ; No, ir a etiqueta a_IMPULSOS
MOVFW Data_EE ; Si, valor de Data_EE a registro W
MOVWF AUXIMPULSOS ; Mueve valor de registro W a AUXIMPULSOS
RETURN ; retornar

```

a\_IMPULSOS

```

movfw Addr_EE ; Mueve valor de Addr_EE a registro W
XORLW .7 ; Compara regsitro W con literal
BTFS STATUS,Z ; ¿es igual a 7?
GOTO a_PUNTO ; No, ir a etiqueta a_PUNTO
MOVFW Data_EE ; Si, valor de Data_EE a registro W
MOVWF IMPULSOS ; Mueve valor de registro W a IMPULSOS
RETURN ; retornar

```

a\_PUNTO

```

movfw Addr_EE ; Mueve valor de Addr_EE a registro W
XORLW .6 ; Compara regsitro W con literal
BTFS STATUS,Z ; ¿es igual a 6?
GOTO a_UNIDADES ; No, ir a etiqueta a_UNIDADES
MOVFW Data_EE ; Si, valor de Data_EE a registro W

```

```

MOVWF PUNTO          ; Mueve valor de registro W a PUNTO
RETURN              ; retornar

a_UNIDADES
movfw Addr_EE       ; Mueve valor de Addr_EE a registro W
XORLW .5            ; Compara regsitro W con literal
BTSS STATUS,Z      ; ¿es igual a 5?
GOTO a_DECENAS     ; No, ir a etiqueta a_DECENAS
MOVFW Data_EE      ; Si, valor de Data_EE a registro W
MOVWF UNIDADES     ; Mueve valor de registro W a UNIDADES
RETURN            ; retornar

a_DECENAS
movfw Addr_EE       ; Mueve valor de Addr_EE a registro W
XORLW .4            ; Compara regsitro W con literal
BTSS STATUS,Z      ; ¿es igual a 4?
GOTO a_CENTENAS    ; No, ir a etiqueta a_CENTENAS
MOVFW Data_EE      ; Si, valor de Data_EE a registro W
MOVWF DECENAS      ; Mueve valor de registro W a DECENAS
RETURN            ; retornar

a_CENTENAS
movfw Addr_EE       ; Mueve valor de Addr_EE a registro W
XORLW .3            ; Compara regsitro W con literal
BTSS STATUS,Z      ; ¿es igual a 3?
GOTO a_UNIDADES_MIL ; No, ir a etiqueta a_UNIDADES_MIL
MOVFW Data_EE      ; Si, valor de Data_EE a registro W
MOVWF CENTENAS     ; Mueve valor de registro W a CENTENAS
RETURN            ; retornar

a_UNIDADES_MIL
movfw Addr_EE       ; Mueve valor de Addr_EE a registro W
XORLW .2            ; Compara regsitro W con literal
BTSS STATUS,Z      ; ¿es igual a 2?
GOTO a_DECENAS_MIL ; No, ir a etiqueta a_DECENAS_MIL
MOVFW Data_EE      ; Si, valor de Data_EE a registro W
MOVWF UNIDADES_MIL ; Mueve valor de registro W a UNIDADES_MIL
RETURN            ; retornar

a_DECENAS_MIL
movfw Addr_EE       ; Mueve valor de Addr_EE a registro W
XORLW .1            ; Compara regsitro W con literal
BTSS STATUS,Z      ; ¿es igual a 1?
GOTO PROGRAMA_PRINCIPAL ; No, ir a etiqueta PROGRAMA_PRINCIPAL
MOVFW Data_EE      ; Si, valor de Data_EE a registro W
MOVWF DECENAS_MIL ; Mueve valor de registro W a DECENAS_MIL
RETURN            ; retornar

final_annotar
bsf OLED           ; Encender led
CALL RET_50MS      ; retardo de 50 ms
BCF LED            ; apagar led
CALL RET_50MS      ; reatrdo de 50 ms
movlw b'10001000' ; Carga el valor al registro de trabajo W
movwf INTCON       ; Mueve valor de regsitro W a registro INTCON
GOTO PROGRAMA_PRINCIPAL ; ir a la etiqueta PROGRAMA_PRINCIPAL

;-----Subrutina Incrementar Impulso-----

IN_IMPULSO

CONT_IMPULSOS

MOVLW .1           ; Carga el valor al registro de trabajo W
ADDWF IMPULSOS,W   ; Suma el valor de regsitro W con variable IMPULSOS

```

```

MOVWF IMPULSOS           ; Mueve el valor de registro W a variable IMPULSOS
movfw  IMPULSOS          ; Mueve el valor de variable IMPULSOS a registro W
XORLW  .80                ; Compara registro W con literal
BTFSS  STATUS,Z          ; ¿es igual a 80?
RETURN                    ; No, retornar
MOVLW  .1                 ; Si, Carga el valor al registro de trabajo W
ADDWF  AUXIMPULSOS,W     ; Suma el valor de registro W con variable AUXIMPULSOS
MOVWF  AUXIMPULSOS       ; Mueve el valor de registro W a variable AUXIMPULSOS
movfw  AUXIMPULSOS       ; Mueve el valor de variable AUXIMPULSOS a registro W
XORLW  .2                 ; Compara registro W con literal
BTFSS  STATUS,Z          ; ¿es igual a 2?
GOTO   OTRAVEZ           ; No, ir a la etiqueta OTRAVEZ
GOTO   INC_PUNTO         ; Si, ir a la etiqueta INC_PUNTO

OTRAVEZ
movlw  .0                 ; Carga el valor al registro de trabajo W
MOVWF  IMPULSOS          ; Mueve el valor de registro W a variable IMPULSOS
RETURN                    ; retorna

INC_PUNTO
movlw  .0                 ; Carga el valor al registro de trabajo W
MOVWF  IMPULSOS          ; Mueve el valor de registro W a variable IMPULSOS
MOVWF  AUXIMPULSOS       ; Mueve el valor de registro W a variable AUXIMPULSOS
MOVLW  .1                 ; Carga el valor al registro de trabajo W
ADDWF  PUNTO,W           ; Suma el valor de registro W con variable PUNTO
MOVWF  PUNTO             ; Mueve el valor de registro W a variable PUNTO
movfw  PUNTO             ; Mueve el valor de variable PUNTO a registro W
XORLW  .10                ; Compara registro W con literal
BTFSS  STATUS,Z          ; ¿es igual a 10?
RETURN                    ; No, retorna

;INC_UNIDADES
movlw  .0                 ; Si, Carga el valor al registro de trabajo W
MOVWF  PUNTO             ; Mueve el valor de registro W a variable PUNTO
MOVLW  .1                 ; Carga el valor al registro de trabajo W
ADDWF  UNIDADES,W        ; Suma el valor de registro W con variable UNIDADES
MOVWF  UNIDADES          ; Mueve el valor de registro W a variable UNIDADES
movfw  UNIDADES          ; Mueve el valor de variable UNIDADES a registro W
xorlw  .10                ; Compara registro W con literal
btfs  STATUS,Z           ; ¿es igual a 10?
GOTO   uninormal         ; No, ir a etiqueta uninormal
GOTO   unicero           ; Si, ir a etiqueta unicero

unicero
movlw  .0                 ; Carga el valor al registro de trabajo W
movwf  UNIDADES          ; Mueve el valor de registro W a variable UNIDADES
call   GRABAR            ; No, Llama a la librería GRABAR
goto   INC_DECENAS       ; Si, ir a etiqueta INC_DECENAS

uninormal
;-----
call   GRABAR            ; Llama a la librería GRABAR
movfw  UNIDADES          ; Mueve el valor de variable UNIDADES a registro W
XORLW  .10                ; Compara registro W con literal
BTFSS  STATUS,Z          ; ¿es igual a 10?
RETURN                    ; No, retornar

INC_DECENAS
movlw  .0                 ; Carga el valor al registro de trabajo W
MOVWF  UNIDADES          ; Mueve el valor de registro W a variable UNIDADES
MOVLW  .1                 ; Carga el valor al registro de trabajo W
ADDWF  DECENAS,W        ; Suma el valor de registro W con variable DECENAS

```

```

MOVWF DECENAS          ; Mueve el valor de registro W a variable DECENAS
movfw DECENAS          ; Mueve el valor de variable DECENAS a registro W
XORLW .10              ; Compara regsitro W con literal
BTFSZ STATUS,Z        ; ¿es igual a 10?
RETURN                 ; No, retornar

;INC_CENTENAS
movlw .0               ; Si,Carga el valor al registro de trabajo W
MOVWF DECENAS          ; Mueve el valor de registro W a variable DECENAS
MOVLW .1               ; Carga el valor al registro de trabajo W
ADDWF CENTENAS,W      ; Suma el valor de regsitro W con variable CENTENAS
MOVWF CENTENAS        ; Mueve el valor de registro W a variable CENTENAS
movfw CENTENAS        ; Mueve el valor de variable CENTENAS a registro W
XORLW .10              ; Compara regsitro W con literal
BTFSZ STATUS,Z        ; ¿es igual a 10?
RETURN                 ; No, retornar

;INC_UNIDADES_MIL
movlw .0               ; Si,Carga el valor al registro de trabajo W
MOVWF CENTENAS        ; Mueve el valor de registro W a variable CENTENAS
MOVLW .1               ; Carga el valor al registro de trabajo W
ADDWF UNIDADES_MIL,W ; Suma el valor de regsitro W con variable UNIDADES_MIL
MOVWF UNIDADES_MIL    ; Mueve el valor de registro W a variable UNIDADES_MIL
movfw UNIDADES_MIL    ; Mueve el valor de variable UNIDADES_MIL a registro W
XORLW .10              ; Compara regsitro W con literal
BTFSZ STATUS,Z        ; ¿es igual a 10?
RETURN                 ; No, retornar

;INC_DECENAS_MIL
movlw .0               ; Si,Carga el valor al registro de trabajo W
MOVWF UNIDADES_MIL    ; Mueve el valor de registro W a variable UNIDADES_MIL
MOVLW .1               ; Carga el valor al registro de trabajo W
ADDWF DECENAS_MIL,W   ; Suma el valor de regsitro W con variable DECENAS_MIL
MOVWF DECENAS_MIL     ; Mueve el valor de registro W a variable DECENAS_MIL
movfw DECENAS_MIL     ; Mueve el valor de variable UNIDADES_MIL a registro W
XORLW .10              ; Compara regsitro W con literal
BTFSZ STATUS,Z        ; ¿es igual a 10?
RETURN                 ; No, retornar

GAMEOVER
SLEEP                  ; Si, Fin de Capacidad de lectura del registrador electrónico
                       ; monofásico.
GOTO GAMEOVER

;-----SUBROUTINA PRINCIPAL DE INTERRUOCIONES-----

INTERRUPCIONES

intgrabar
    bcf STATUS,RP0     ; Cambio al BANCO 0
    BTFSZ NOLUZ_GRABAR ; se fue la luz?
    GOTO aNOLUZ_GRABAR ; Si, ir al etiqueta aNOLUZ_GRABAR
    BTFSZ ENTRADA_IMPULSOS ; No, ¿Ingreso de Impulso ?
    CALL aENTRADA_IMPULSO ; Si, ir a la etiqueta aENTRADA_IMPULSO

heree
    bcf INTCON,RBIF    ; No, reseteo bit RBIF
    RETFIE             ; retornar de la Interrupción

```



```

aNOLUZ_GRABAR
    bcf     STATUS,RP0           ; Cambio al BANCO 0
    CALL   GRABAR               ; Llama a la subrutina GRABAR
    bcf     LED                  ; apagar led
    call   RET_50MS             ; retardo de 50 ms
    bsf     LED                  ; encender led

Esperaluz
    BTFSC  NOLUZ_GRABAR        ; ¿existe alimentación de energía?
    goto   Esperaluz           ; No, ir a la etiqueta Esperaluz
    goto   heree                ; Si, ir a la etiqueta heree

aENTRADA_IMPULSO
    BCF    STATUS,RP0           ; Cambio al BANCO 0
    CALL   IN_IMPULSO          ; Llama a la Subrutina IN_IMPULSO

Esperaimp
    BTFSC  ENTRADA_IMPULSOS    ; ¿acabo de ingresar impulso?
    goto   Esperaimp           ; No, ir a la etiqueta BTFSC
    return ; Si, retornar

;-----Grabar EEPROM-----

GRABAR_EEPROM

    BANKSEL Addr_EE            ; Selecciona BANCO
    MOVF   Addr_EE,W           ; Mueve el valor del Addr_EE a W
    bsf    STATUS,RP0          ; Cambio al BANCO 1
    MOVWF  EEADR                ; Mueve el valor de W al registro EEADR
    BANKSEL Data_EE            ; Selecciona BANCO
    MOVF   Data_EE,W           ; Mueve el valor del Data_EE a W
    BSF    STATUS,RP0          ; Cambio al BANCO 1
    MOVWF  EEDATA              ; Mueve el valor de "W" al registro "EEDATA"
    BSF    EECON1,WREN         ; Habilita el BIT de escritura en la EEPROM (WREN)
    MOVLW  0x55                ; Secuencia obligatoria
    MOVWF  EECON2              ; para activar la grabación
    MOVLW  0xAA                ; de la memoria
    MOVWF  EECON2              ; EEPROM
    BSF    EECON1,WR           ; Inicia el ciclo de escritura Bit WR
    BCF    STATUS,RP0          ; Cambio al BANCO 0

GRABAR_EEPROM_ESPERAR
    BTFSS  PIR1,EEIF           ; Lee ek EEIF y salta una instrucción si el valor es 1
    GOTO   GRABAR_EEPROM_ESPERAR ; Espera de grabación de la EEPROM
    BCF    PIR1,EEIF           ; Limpia el flag EEIF
    BSF    STATUS,RP0          ; Cambio al BANCO 1
    BCF    EECON1,WREN         ; Deshabilita el Bit de escritura de la EEPROM (WREN)
    BANKSEL Data_EE            ; Selecciona BANCO

GRABAR_EEPROM_VERIFICACION
    MOVFW  Data_EE              ; Coloca el valor leído en W
    BSF    STATUS,RP0          ; Cambio al BANCO 1
    BSF    EECON1, RD          ; Leer la EEPROM
    SUBWF  EEDATA, W           ; Compara el valor leído con el grabado
    BTFSS  STATUS,Z            ; Si son iguales, salta una instrucción
    GOTO   GRABAR_EEPROM ; Si no son iguales, comienza de nuevo la rutina
    BCF    STATUS,RP0          ; Cambio al BANCO 0
    RETURN

;-----Leer EEPROM-----
-
LEER_EEPROM

EEPROM_LeerDato

```

```

MOVFW Addr_EE          ; Mueve valor de Addr_EE a registro de W
bsf   STATUS,RP0       ; Cambio al BANCO 1
movwf EEADR            ; Mueve valor de W a EEADR
BSF   EECON1,RD        ; Inicia el ciclo de lectura bit RD

```

EEPROM\_Sigueleyendo

```

btfsc EECON1,RD        ; ¿termino de leer?
goto  EEPROM_Sigueleyendo; No, ir a la etiqueta EEPROM_Sigueleyendo
movf  EEDATA,W         ; Si, Mueve EEDATA a W
BCF   STATUS,RP0       ; Cambio a BANCO 0
MOVWF Data_EE         ; Mueve valor de W a Data_EE
return                               ; retornar

```

-----Subrutina Administrar Datos-----

ADMINISTRAR\_DATOS

; Esta subrutina es la encargada de ingresar remotamente los valores de la lectura una vez ingresados los valores se retorna a la etiqueta ANOTAR VARIABLES ;por seguridad de la Empresa no se muestra las lineas de dicha subrutina, la misma que se explica teoricamente en el anexo 11.

```

goto ANOTAR_VARIABLES

```

-----

```

INCLUDE "L:\TESIS\MICROS\RETARDOS\retardos.inc" ; librería de retardos
INCLUDE "L:\TESISREMOTA\EI Diseño\inicializar.inc" ; librería de inicializar
INCLUDE "L:\TESISREMOTA\EI Diseño\suministro.inc" ; librería de suministro

END ; FIN DEL PROGRAMA

```

```

;*****Librería Inicializar*****
;.....Subrutina inicializar.....

GRABA_DATOS_INI
;#AUX_IMPULSOS
    MOVLW    B'00001000'    ; Carga el valor de registro de trabajo W
    MOVWF   Addr_EE        ; Mueve el valor W a Addr_EE
    MOVLW   .0              ; Carga el valor al registro de trabajo W
    MOVWF   Data_EE        ; Mueve el valor de W a Data_EE
    CALL    GRABAR_EEPROM  ; Llama a la subrutina GRABAR_EEPROM

;#IMPULSOS
    MOVLW   B'00000111'    ; Carga el valor de registro de trabajo W
    MOVWF   Addr_EE        ; Mueve el valor W a Addr_EE
    MOVLW   .0              ; Carga el valor al registro de trabajo W
    MOVWF   Data_EE        ; Mueve el valor de W a Data_EE
    CALL    GRABAR_EEPROM  ; Llama a la subrutina GRABAR_EEPROM

;#PUNTO
    MOVLW   B'00000110'    ; Carga el valor de registro de trabajo W
    MOVWF   Addr_EE        ; Mueve el valor W a Addr_EE
    MOVLW   .0              ; Carga el valor al registro de trabajo W
    MOVWF   Data_EE        ; Mueve el valor de W a Data_EE
    CALL    GRABAR_EEPROM  ; Llama a la subrutina GRABAR_EEPROM

;#UNIDADES
    MOVLW   B'00000101'    ; Carga el valor de registro de trabajo W
    MOVWF   Addr_EE        ; Mueve el valor W a Addr_EE
    MOVLW   .0              ; Carga el valor al registro de trabajo W
    MOVWF   Data_EE        ; Mueve el valor de W a Data_EE
    CALL    GRABAR_EEPROM  ; Llama a la subrutina GRABAR_EEPROM

;#DECENAS
    MOVLW   B'00000100'    ; Carga el valor de registro de trabajo W
    MOVWF   Addr_EE        ; Mueve el valor W a Addr_EE
    MOVLW   .0              ; Carga el valor al registro de trabajo W
    MOVWF   Data_EE        ; Mueve el valor de W a Data_EE
    CALL    GRABAR_EEPROM  ; Llama a la subrutina GRABAR_EEPROM

;#CENTENAS
    MOVLW   B'00000011'    ; Carga el valor de registro de trabajo W
    MOVWF   Addr_EE        ; Mueve el valor W a Addr_EE
    MOVLW   .0              ; Carga el valor al registro de trabajo W
    MOVWF   Data_EE        ; Mueve el valor de W a Data_EE
    CALL    GRABAR_EEPROM  ; Llama a la subrutina GRABAR_EEPROM

;#UNIDADES_MIL
    MOVLW   B'00000010'    ; Carga el valor de registro de trabajo W
    MOVWF   Addr_EE        ; Mueve el valor W a Addr_EE
    MOVLW   .0              ; Carga el valor al registro de trabajo W
    MOVWF   Data_EE        ; Mueve el valor de W a Data_EE
    CALL    GRABAR_EEPROM  ; Llama a la subrutina GRABAR_EEPROM

;#DECENAS_MIL
    MOVLW   B'00000001'    ; Carga el valor de registro de trabajo W
    MOVWF   Addr_EE        ; Mueve el valor W a Addr_EE
    MOVLW   .0              ; Carga el valor al registro de trabajo W
    MOVWF   Data_EE        ; Mueve el valor de W a Data_EE
    CALL    GRABAR_EEPROM  ; Llama a la subrutina GRABAR_EEPROM

;#
    MOVLW   .10             ; Carga el valor de registro de trabajo W
    MOVWF   Addr_EE        ; Mueve el valor W a Addr_EE
    MOVLW   .100           ; Carga el valor al registro de trabajo W
    MOVWF   Data_EE        ; Mueve el valor de W a Data_EE
    CALL    GRABAR_EEPROM  ; Llama a la subrutina GRABAR_EEPROM
    RETURN                ; retornar

```

```
,*****Librería Suministro*****
;.....Subrutina suministro.....
```

#### EnviarSuministro

```

movlw      .0          ; Carga el valor al registro de trabajo W
movwf     AYUDA_ENV_DATOS ; Mueve el valor de W a AYUDA_ENV_DATOS
CALL     enviarRS232    ; Llama a la subrutina enviarRS232

movlw      .1          ; Carga el valor al registro de trabajo W
movwf     AYUDA_ENV_DATOS ; Mueve el valor de W a AYUDA_ENV_DATOS
CALL     enviarRS232    ; Llama a la subrutina enviarRS232

movlw      .4          ; Carga el valor al registro de trabajo W
movwf     AYUDA_ENV_DATOS ; Mueve el valor de W a AYUDA_ENV_DATOS
CALL     enviarRS232    ; Llama a la subrutina enviarRS232

movlw      .0          ; Carga el valor al registro de trabajo W
movwf     AYUDA_ENV_DATOS ; Mueve el valor de W a AYUDA_ENV_DATOS
CALL     enviarRS232    ; Llama a la subrutina enviarRS232

movlw      .9          ; Carga el valor al registro de trabajo W
movwf     AYUDA_ENV_DATOS ; Mueve el valor de W a AYUDA_ENV_DATOS
CALL     enviarRS232    ; Llama a la subrutina enviarRS232

movlw      .3          ; Carga el valor al registro de trabajo W
movwf     AYUDA_ENV_DATOS ; Mueve el valor de W a AYUDA_ENV_DATOS
CALL     enviarRS232    ; Llama a la subrutina enviarRS232

movlw      .4          ; Carga el valor al registro de trabajo W
movwf     AYUDA_ENV_DATOS ; Mueve el valor de W a AYUDA_ENV_DATOS
CALL     enviarRS232    ; Llama a la subrutina enviarRS232

MOVLW    .45          ; Carga el valor al registro de trabajo W
movwf     TXREG        ; Mueve el valor de W a TXREG
bsf      STATUS,RPO   ; Cambio al BANCO 1

btfss    TXSTA,TRMT   ; ¿Termino de enviar?
goto     er           ; No, ir a la etiqueta er
bcf      STATUS,RPO   ; Si, Cambio al BANCO 0
return   ; retornar

```

er

