```
;                              ELECTRIFICATION HORLOGE COMTOISE KIT version3

       ERRORLEVEL -302
       ERRORLEVEL -305

       LIST      p=16F886            ; Définition de processeur
       #include <p16F886.inc>        ; fichier include

       __CONFIG    _CONFIG1, _LVP_OFF & _FCMEN_ON & _IESO_OFF & _BOR_OFF & _CPD_OFF & _CP_OFF &
_MCLRE_OFF & _PWRTE_ON & _WDT_OFF & _INTOSCIO
       __CONFIG    _CONFIG2, _WRT_OFF & _BOR21V

;*********************************************************************************
;                                ASSIGNATIONS SYSTEME                             *
;*********************************************************************************

; REGISTRE OPTION_REG (configuration)
; -----------------------------------
OPTIONVAL           EQU    B'00000000'

; REGISTRE INTCON (contrôle interruptions standard)
; -------------------------------------------------
INTCONVAL           EQU    B'00000000'

; REGISTRE PIE1 (contrôle interruptions périphériques)
; ----------------------------------------------------
PIE1VAL             EQU    B'00000000'

; REGISTRE PIE2 (contrôle interruptions particulières)
; ----------------------------------------------------
PIE2VAL             EQU    B'00000000'


; REGISTRE OSCCON (contrôle de la vitesse de l'oscillateur interne)
; -----------------------------------------------------------------
OSCCONVAL           EQU    B'01100000'  ; (4 MHz)

; REGISTRE CMCON (COMPARATEURS)
; -------------------------------------
CMCONVAL            EQU    B'00000111'

; REGISTRE VRCON (voltage reference module)
; -----------------------------------------
CVRCONVAL           EQU    B'00000000'

; REGISTRE ADCON1 (ANALOGIQUE/DIGITAL)
; ------------------------------------
ADCON1VAL           EQU    B'00000000' ; PORTA en mode digital

; REGISTRE ANSEL (contrôle du convertisseur A/D)
; ----------------------------------------------
ANSELVAL            EQU    B'00000000'
ANSELHVAL           EQU    B'00000000'

; DIRECTION DES PORTS I/O (0=sortie, 1=entrée)
; --------------------------

DIRPORTA            EQU    B'01000000'  ; Direction PORTA
DIRPORTB            EQU    B'00111111'  ; Direction PORTB
DIRPORTC            EQU    B'10000001'  ; Direction PORTC

; REGISTRE WPUB (Résistances de rappel du PORTB)
; ----------------------------------------------
WPUBVAL             EQU    B'00111110'  ; Résistances de rappel validées


;*********************************************************************************
;                             ASSIGNATIONS PROGRAMME                              *
;*********************************************************************************

LCD_FUNCTION_SET                EQU    B'00101000'
LCD_CURSOR_DISPLAY_SHIFT        EQU    B'00011100'
LCD_DISPLAY_CONTROL             EQU    B'00001100'
LCD_ENTRY_MODE_SET              EQU    B'00000110'

ADRS_RTCC                       EQU    B'11011110'  ; Microchip MCP 79401
```

```
;*****************************************************************************
;                                   DEFINE                                    *
;*****************************************************************************

#define      CARRY         STATUS,C
#define      ZERO          STATUS,Z

#define      LCD_PORT      PORTA
#define      LCD_RS        PORTA , 0     ; LCD: Ligne de sélection de l'afficheur
#define      LCD_E         PORTA , 1     ; LCD: Ligne de commande de contrôle de l'afficheur
#define      LCD_D4        PORTA , 2     ; D4
#define      LCD_D5        PORTA , 3     ; D5
#define      LCD_D6        PORTA , 4     ; D6
#define      LCD_D7        PORTA , 5     ; D7

#define      SCL           PORTC,3       ; Serial clock
#define      SDA           PORTC,4       ; Serial data (bidir)
#define      SDA_DIR             TRISC,4       ; Serial data dir (E/S)


#define      SNEGx         REG_MATH , 7
#define      SNEGy         REG_MATH , 6
#define      SNEGs         REG_MATH , 5
#define      SNEGas        REG_MATH , 4
#define      SNEGaz        REG_MATH , 3
#define      INTFLAG             REG_MATH , 2

#define      SNEG          REG_LCD , 7

#define B_MENU      PORTB,3       ; Bouton poussoir "Menu"
#define B_MOINS     PORTB,4       ; Bouton poussoir "défilement vers l'arrière"
#define B_PLUS      PORTB,5       ; Bouton poussoir "défilement vers l'avant"
#define B_VALID     PORTB,2       ; Bouton poussoir "Validation"

#DEFINE CELREGUL    PORTB,0       ; Cellule déclenchement régulation
#DEFINE FC1  PORTC,0      ; Contact position origine (non utilisé)
#DEFINE FC2  PORTB,1      ; Fin de course fin du remontage
#DEFINE BPREMONT    PORTA,6       ; Bouton-poussoir déclenchement remontage
#define B_ARR_S     PORTC,7       ; Bouton m/a de la sonnerie la nuit

#DEFINE MOTDIRECT   PORTC,6             ; Moteur en marche sens direct
#DEFINE MOTINVERSE  PORTC,5             ; Moteur en marche en sens inverse
#DEFINE EAREGUL     PORTC,1             ; Electro-aimant régulation de l'horloge
#DEFINE EASONNE     PORTC,2             ; Electro-aimant arrêt sonnerie la nuit

;*****************************************************************************
;                                 MACRO                                       *
;*****************************************************************************

       #include <MACRO.asm>

;*****************************************************************************
;                       VARIABLES ZONE COMMUNE                                *
;*****************************************************************************

; Zone de 16 bytes  (16)
; ----------------

       CBLOCK 0x70               ; Début de la zone (0x70 à 0x7F)
       w_temp : 1                ; Sauvegarde registre W
       status_temp : 1                 ; sauvegarde registre STATUS
       FSR_temp : 1              ; sauvegarde FSR (si indirect en interrupt)
       PCLATH_temp : 1                 ; sauvegarde PCLATH (si prog>2K)

       x:6 , y:3 , z:3

       ENDC

;*****************************************************************************
;                          VARIABLES BANQUE 0                                 *
;*****************************************************************************

; Zone de 96 bytes  (81 bytes déclarés et utilisés)
; ------------------------------------
```

```
       CBLOCK 0x20          ; Début de la zone (0x20 à 0x7F)

       seconde, minute, heure, CALIBRE, AVANCE, RETARD, HORLOGE, FLAG
       HEUDEPS, HEUARRS
       REG_MATH, BitCompte, OctCompte, Position
       temp1, temp2, temp3, cmpt0, cmpt1, cmpt2
       REG_LCD, LCD_DATA, LCD_PORT_SAVE
       t:9
       I2C_DONNEE,I2C_flag, I

       ENDC

;*****************************************************************************
;                          VARIABLES BANQUE 1                                *
;*****************************************************************************

; Zone de 80 bytes
; ----------------

       CBLOCK 0xA0          ; Début de la zone (0xA0 à 0xEF)
       ENDC                       ; Fin de la zone

;*****************************************************************************
;                          VARIABLES BANQUE 2                                *
;*****************************************************************************

; Zone de 96 bytes
; ----------------

       CBLOCK 0x110         ; Début de la zone (0x110 à 0x16F)
       ENDC                       ; Fin de la zone

;*****************************************************************************
;                          VARIABLES BANQUE 3                                *
;*****************************************************************************

; Zone de 96 bytes
; ----------------

       CBLOCK 0x190         ; Début de la zone (0x190 à 0x1EF)
       ENDC                       ; Fin de la zone

;*****************************************************************************
;                        MEMOIRE EEPROM                                      *
;*****************************************************************************

       ORG          0x2100
;      DE           0x07         ; Valeur 7 (par défaut) pour 7h00 validation sonnerie
;      DE           0x16         ; Valeur 22 (par défaut) pour 22h00 arrêt sonnerie

;*****************************************************************************
;                        DEMARRAGE SUR RESET                                 *
;*****************************************************************************

       org    0x000               ; Adresse de départ après reset
       goto   init         ; Initialiser

;*****************************************************************************
;                          INITIALISATIONS                                   *
;*****************************************************************************

init
                   ; initialisation PORTS (banque 0 et 1)
                   ; -------------------------------------
       BANK0               ; sélectionner banque0
       clrf   PORTA        ; Sorties PORTA à 0
       clrf   PORTB        ; sorties PORTB à 0
       clrf   PORTC        ; sorties PORTC à 0
       BANK1
       movlw  DIRPORTA     ; Direction PORTA
       movwf  TRISA        ; écriture dans registre direction
       movlw  DIRPORTB     ; Direction PORTB
       movwf  TRISB        ; écriture dans registre direction
       movlw  DIRPORTC     ; Direction PORTC
       movwf  TRISC        ; écriture dans registre direction
       movlw  WPUBVAL             ; charger le masque
       movwf  WPUB         ; initialiser registre
```

```
                ; Registre de l'oscillateur
                ; --------------------------
        movlw   OSCCONVAL           ; charger le masque
        movwf   OSCCON              ; initialiser registre

                ; Registre d'options (banque 1)
                ; ------------------------------
        movlw   OPTIONVAL     ; charger masque
        movwf   OPTION_REG    ; initialiser registre option

                ; registres interruptions (banque 1)
                ; ----------------------------------
        movlw   INTCONVAL     ; charger valeur registre interruption
        movwf   INTCON        ; initialiser interruptions
        movlw   PIE1VAL             ; Initialiser registre
        movwf   PIE1          ; interruptions périphériques 1
        movlw   PIE2VAL             ; initialiser registre
        movwf   PIE2          ; interruptions périphériques 2

                ; registres A/D (banque 3)
                ; ---------------------------------
        BANK3
        movlw   ANSELVAL
        movwf   ANSEL
        movlw   ANSELHVAL
        movwf   ANSELH

                ; Effacer RAM banque 0
                ; ---------------------
        BANK0
        movlw   0x20          ; initialisation pointeur
        movwf   FSR                 ; d'adressage indirect
init1
        clrf    INDF          ; effacer ram
        incf    FSR,f         ; pointer sur suivant
        btfss   FSR,7         ; tester si fin zone atteinte (>7F)
        goto    init1         ; non, boucler

                ; Initialisation de la carte électronique

                ; --------------------------
        bcf     MOTDIRECT           ;
        bcf     MOTINVERSE          ;

        call    LCD_INIT            ; Initialisation de l'afficheur LCD
        call    I2C_INIT            ; Initialisation du bus I2C
        call    ARRS_INIT           ; Lecture EEPROM heure départ et arrêt sonnerie

;*****************************************************************************
;                       PROGRAMME PRINCIPAL                                  *
;*****************************************************************************
start
        call    AFFICHAGE           ; Affichage de heure, de l'avance et du retard
        btfsc   B_ARR_S             ; Bouton mise en service ou arrêt sonnerie la nuit
        goto    sta1
        goto    sta2
sta1
        call    SONNERIE            ; Commande d'arrêt de la sonnerie pour la nuit
        goto    start0
sta2
        bcf     EASONNE             ; Mise en fonction et arrêt électro-aimant sonnerie
start0
        btfsc   B_MENU        ; Bouton pour réglage des paramètres
        goto    start1
        call    Tempo_200ms
        btfsc   B_MENU
        goto    start1
        call    REGLAGE             ; Réglage de l'heure, heure M/A sonnerie, calibration
start1
        btfsc   HORLOGE,2
        goto    star7         ; HORLOGE,2 égal à 1
        movf    heure,0             ; minute dans w (HORLOGE,2 est égal à 0)
        sublw   d'7'          ; w moins 7  Remontage à 7h
        btfss   ZERO          ; Est-ce que heure est égal à 7 ?
        goto    star5         ; Non
        goto    start10             ; Oui
```

```
star5
      movf   heure,0            ; heure dans w
      sublw  d'19'        ; w moins 19 Remontage à 19h
      btfss  ZERO
      goto   start8
start10
      movf   minute,0           ; minute dans w
      sublw  d'45'        ; w moins 45
      btfss  ZERO         ; Est-ce que minute est égal à 45?
      goto   start8       ; Non
      bsf    HORLOGE,2          ; Oui. Positionnement flag HORLOGE,2 à minute 45
      bsf    MOTDIRECT          : Mise en marche du remontage
      call   Tempo_1s           ; Tempo 1 seconde
      bcf    MOTDIRECT          ; Arrêt du signal de remontage
      goto   start8       ; Poursuite du programme
star7
      movf   minute,0           ; minute dans w
      sublw  d'46'        ; w moins 46
      btfss  ZERO         ; Est-ce que minute est égal à 46?
      goto   start8       ; Non (poursuite du programme)
      bcf    HORLOGE,2          ; Oui (remise à zéro du flag "HORLOGE,2" et poursuite du
programme)
start8
      call   Tempo_10ms   ; Vers programme de régulation à partir de la 58éme minute
      movf   minute,0           ;
      sublw  d'58'              ; Autorise la régulation à partir de la 58ème minute
      btfss  ZERO
      goto   start11
      goto   start13
start11
      movf   minute,0
      sublw  d'59'
      btfss  ZERO
      goto   start6             ; Pas de régulation (vers fin de prog de régulation)
      goto   start12
start12
      clrf   FLAG
      goto   start14
start13
      movlw  d'60'
      movwf  FLAG
start14
      btfss  CELREGUL           ; Cellule sur pignon petite aiguille
      goto   start
      call   REGULATION   ; Régulation en fonction de l'heure d'avance ou de retard
      goto   start

;                     FIN DU PROGRAMME PRINCIPAL
;*****************************************************************************
;*****************************************************************************
;                          SOUS PROGRAMMES                                   *
;*****************************************************************************

REGULATION:         ; Execute le blocage de la roue d'échappement par tranche de 10 secondes
                          ; Le blocage commence au changement d'heure pour une durée de 10s.
                          ; Puis se répète chaque minute en fonction du temps d'avance calculé.
                          ; Exemple: si l'avance calculée est de 45s, le programme exécute
                          ; 4 blocages de 10s et un blocage de 5s.
                          ; Le programme s'exécute toutes les 2 heures.

      btfsc  HORLOGE,0
      goto   start
      bsf    HORLOGE,0
      movf   seconde,0
      sublw  d'60'
      movwf  AVANCE
      movf   FLAG,0
      addwf  AVANCE,1           ; Calcul de AVANCE
      clrf   RETARD
      movf   AVANCE,0
      movwf  cmpt2
      movlw  d'10'              ; Temps de blocage 10s
      subwf  cmpt2,1
      btfss  cmpt2,7
      goto   start15
      goto   start16
start16
```

```
       movf   seconde,0
       andlw  0xFF
       btfss  ZERO
       goto   start161
       goto   start162
start161
       call   AFFICHAGE
       goto   start16
start162
       bsf    EAREGUL
       movf   AVANCE,0
       subwf  seconde,0
       btfss  ZERO
       goto   start163
       goto   start164
start163
       call   AFFICHAGE
       goto   start162
start164
       bcf    EAREGUL
       movf   minute,0
       sublw  0x01
       btfss  ZERO
       goto   start165
       goto   start166
start165
       call   AFFICHAGE
       goto   start164
start166
       bcf    HORLOGE,0
       goto   start
start15
       movly  d'0',d'0',d'10'          ; Découpage du temps d'avance par tranche de 10 s.
       clrf   x+2
       clrf   x+1
       movf   AVANCE,0
       movwf  x+0
       call   FXD_2424S
       movf   x+0,0
       movwf  cmpt0
       movf   z+0,0
       movwf  cmpt1
start2
       movf   minute,0
       andlw  0xFF
       btfss  ZERO
       goto   start21
       goto   start3
start21
       call   AFFICHAGE                ; Affichage lors des boucles de programme
       goto   start2
start3
       bsf    EAREGUL
       movf   seconde,0
       sublw  d'10'
       btfss  ZERO
       goto   start31
       goto   start32
start31
       call   AFFICHAGE
       goto   start3
start32
       bcf    EAREGUL
       decfsz cmpt0
       goto   start4
       goto   start5
start4
       movf   seconde,0
       andlw  0xFF
       btfss  ZERO
       goto   start41
       goto   start42
start41
       call   AFFICHAGE
       goto   start4
start42
       goto   start3
```

```
start5
       movf   seconde,0
       andlw  0xFF
       btfss  ZERO
       goto   start51
       goto   start52
start51
       call   AFFICHAGE
       goto   start5
start52
       bsf    EAREGUL
       movf   cmpt1,0
       subwf  seconde,0
       btfss  ZERO
       goto   start53
       goto   start54
start53
       call   AFFICHAGE
       goto   start52
start54
       bcf    EAREGUL
       bcf    HORLOGE,0
       goto   start
start6
       movf   minute,0
       andlw  0xFF
       btfss  ZERO
       goto   start61
       goto   start62
start61
       bcf    HORLOGE,1
       goto   start
start62
       btfss  CELREGUL
       goto   start
       btfsc  HORLOGE,1
       goto   start
       bsf    HORLOGE,1
       movf   seconde,0
       movwf  RETARD
       clrf   AVANCE
       return

;*******************************************************************************

SONNERIE:           ; Exécute de blocage de la sonnerie en fonction de l'heure d'arrêt
                    ; pour la nuit et en fonction de l'heure de déblocage le matin.
                    ; Le programme est exécuté seulement si le bouton de la sonnerie
                    ; permet ou non le blocage nocturne de la sonnerie.

       movf   heure,0           ; heure dans "w"
       sublw  d'21'        ; 21 moins "w"
       btfss  ZERO
       goto   st1
       goto   st13
st1
       movf   heure,0           ; heure dans "w"
       sublw  d'22'        ; 22 moins "w"
       btfss  ZERO
       goto   st2
       goto   st13
st2
       movf   heure,0           ; heure dans "w"
       sublw  d'23'        ; 23 moins "w"
       btfss  ZERO
       goto   st3
       goto   st13
st3
       movf   heure,0           ; heure dans "w"
       sublw  d'0'         ; 0 moins "w"
       btfss  ZERO
       goto   st4
       goto   st15
st4
       movf   heure,0           ; heure dans "w"
       sublw  d'1'         ; 1 moins "w"
       btfss  ZERO
```

```
      goto   st5
      goto   st15
st5
      movf   heure,0             ; heure dans "w"
      sublw  d'2'          ; 2 moins "w"
      btfss  ZERO
      goto   st6
      goto   st15
st6
      movf   heure,0             ; heure dans "w"
      sublw  d'3'          ; 3 moins "w"
      btfss  ZERO
      goto   st7
      goto   st15
st7
      movf   heure,0             ; heure dans "w"
      sublw  d'4'          ; 4 moins "w"
      btfss  ZERO
      goto   st8
      goto   st15
st8
      movf   heure,0             ; heure dans "w"
      sublw  d'5'          ; 5 moins "w"
      btfss  ZERO
      goto   st9
      goto   st14
st9
      movf   heure,0             ; heure dans "w"
      sublw  d'6'          ; 6 moins "w"
      btfss  ZERO
      goto   st10
      goto   st14
st10
      movf   heure,0             ; heure dans "w"
      sublw  d'7'          ; 7 moins "w"
      btfss  ZERO
      goto   st20
      goto   st14
st13
      movf   heure,0             ; heure copié dans "w"
      subwf  HEUARRS,0           ; contenu du registre HEUARRS moins "w"
      btfss  CARRY         ; test du bit de CARRY
      goto   st15          ; valeur négative ou nulle (bsf  EASONNE)
      btfss  ZERO          ; test si valeur nulle
      goto   st20          ; valeur non nulle (bcf EASONNE)
      movf   minute,0            ; minute dans "w"
      sublw  d'14'         ; 14 moins w
      btfsc  CARRY         ; test du bit de CARRY
      goto   st20          ; minute est inférieur à 14 mn (bcf    EASONNE)
      goto   st15          ; minute est égal à 14 mn (bsf EASONNE)
                                 ; Arrêt sonnerie à partir de la quinzième mn
st14
      movf   heure,0             ; heure copié dans "w"
      subwf  HEUDEPS,0           ; contenu du registre HEUDEPS moins "w"
      btfss  CARRY         ; test du bit de CARRY
      goto   st20          ; valeur négative ou nulle (bcf  EASONNE)
      btfss  ZERO          ; test si valeur nulle
      goto   st15          ; valeur non nulle (bsf EASONNE)
      movf   minute,0            ; minute dans "w"
      sublw  d'14'         ; 14 moins w
      btfsc  CARRY         ; test du bit de CARRY
      goto   st15          ; minute est inférieur à 14 mn (bsf    EASONNE)
      goto   st20          ; minute est égal à 14 mn (bcf EASONNE)
                           ; Sonnerie en fonction à partir de la quinzième mn
st15
      bsf    EASONNE             ; Sonnerie hors fonctionnement la nuit
      return
st20
      bcf    EASONNE             ; Sonnerie en fonctionnement en permanence
      return

;*****************************************************************************

      #include <q4mhz.asm>
      #include <I2C.asm>
      #include <MATH.asm>
```

```
LIRE_RTCC:                          ; Lecture des valeurs de l'horloge RTCC

            call   I2C_START_       ; Bit start condition
            movlw  ADRS_RTCC+0      ; Adresse du circuit RTCC (écriture)
            call   I2C_SEND_BYTE    ; Envoi de l'adresse du circuit RTCC
            movlw  b'00000000'      ; Adresse de lecture du circuit RTCC
            call   I2C_SEND_BYTE    ; Envoi de l'adresse du 1er octet pour la lecture
            call   I2C_START_       ; Bit start condition
            movlw  ADRS_RTCC+1      ; Adresse du circuit RTCC (lecture)
            call   I2C_SEND_BYTE    ; Envoi de l'adresse du 1er octet pour la lecture
            call   I2C_RECEIVE_BYTE         ; Lecture du 1er octet à l'adresse h00
            movwf  seconde                  ; Transfère de la valeur lue dans le registre
"seconde"
            bcf    seconde , 7      ; Effacement du bit 7 du registre "seconde"
            call   I2C_RECEIVE_BYTE         ; Lecture de l'octet suivant
            movwf  minute           ; Transfère de la valeur lue dans le registre "minute"
            call   I2C_RECEIVE_BYTE_LAST    ; Lecture du dernier octet
            movwf  heure            ; Transfère de la valeur lue dans le registre "heure"
            call   I2C_STOP_                ; Stop condition
            CALLX  conv_HD_Heure    ; Conversion codage BCD vers système binaire date et
heure

;*********************************************************************************

AFF_HEURE                           ; Affichage de l'HEURE sur la 1 ère ligne

      call   LCD_EFFACE
      call   Tempo_1ms
      call   Tempo_1ms
      movlw  "0"
      call   LCD_DONNEE
      movlw  "0"
      call   LCD_DONNEE
      clrf   x+2                ;
      clrf   x+1                ;
      movfw  heure              ;
      movwf  x+0                ;
      movlw  d'1'               ;
      call   AFFICHE_8BS        ;     Affichage "HEURES" de l'horloge à quartz
      movlw  0x68               ;
      call   LCD_DONNEE         ;     Affichage signe "h"
      movlw  "0"
      call   LCD_DONNEE
      movlw  "0"
      call   LCD_DONNEE
      clrf   x+2
      clrf   x+1
      movfw  minute
      movwf  x+0
      movlw  d'4'
      call   AFFICHE_8BS        ;Affichage "MINUTES" de l'horloge à quartz"
      movlw  0x3A
      call   LCD_DONNEE         ;Affichage signe "deux points" ":"
      movlw  "0"
      call   LCD_DONNEE
      movlw  "0"
      call   LCD_DONNEE
      clrf   x+2
      clrf   x+1
      movfw  seconde
      movwf  x+0
      movlw  d'7'
      call   AFFICHE_8BS  ;Affichage "SECONDES" de l'horloge à quartz"
      return

;*********************************************************************************

AFF_LIGNE2
      movlw  d'64'
      call   LCD_LOCATE
      movlw  B'01000001'
      call   LCD_DONNEE   ;Affichage signe "A" (pour avance)
      movlw  "0"
      call   LCD_DONNEE
      movlw  "0"
      call   LCD_DONNEE
```

```
        clrf   x+2          ;
        clrf   x+1          ;
        movf   AVANCE,w             ;
        movwf  x+0          ;
        movlw  d'67'
        call   AFFICHE_24BS ;      Affichage du registre "AVANCE"
        movlw  d'69'
        call   LCD_LOCATE
        movlw  B'01010010'
        call   LCD_DONNEE   ;      Affichage signe "R" (pour retard)
        movlw  "0"
        call   LCD_DONNEE
        movlw  "0"
        call   LCD_DONNEE
        clrf   x+2          ;
        clrf   x+1          ;
        movf   RETARD,w             ;
        movwf  x+0          ;
        movlw  d'71'
        call   AFFICHE_24BS ;      Affichage du registre "RETARD"
        return

;*******************************************************************************

AFFICHAGE:
        call   Tempo_100ms
        call   LIRE_RTCC
        call   AFF_HEURE
        call   AFF_LIGNE2
        return

;*******************************************************************************

REGLAGE:

par1
                call   LCD_EFFACE
                call   Tempo_10ms
                movlw  d'0'
                call   LCD_LOCATE
                movlw  "H"
                call   LCD_DONNEE
                movlw  "E"
                call   LCD_DONNEE
                movlw  "U"
                call   LCD_DONNEE
                movlw  "R"
                call   LCD_DONNEE
                movlw  "E"
                call   LCD_DONNEE
                call   Tempo_200ms
                btfsc  B_MOINS
                goto   $+2
                goto   par7
                btfsc  B_VALID
                goto   $+2
                goto   par2
                btfsc  B_PLUS
                goto   par1
                goto   par3
par2
                call   REGL_HEU
                call   REGL_MIN
par3
                call   LCD_EFFACE
                call   Tempo_10ms
                movlw  d'0'
                call   LCD_LOCATE
                movlw  "S"
                call   LCD_DONNEE
                movlw  "O"
                call   LCD_DONNEE
                movlw  "N"
                call   LCD_DONNEE
                movlw  "N"
                call   LCD_DONNEE
                movlw  "E"
```

```
            call  LCD_DONNEE
            movlw "R"
            call  LCD_DONNEE
            movlw "I"
            call  LCD_DONNEE
            movlw "E"
            call  LCD_DONNEE
            call  Tempo_200ms
            btfsc B_MOINS
            goto  $+2
            goto  par1
            btfsc B_VALID
            goto  $+2
            goto  par4
            btfsc B_PLUS
            goto  par3
            goto  par5
par4
            call  REGL_DEPS
            call  REGL_ARRS
par5
            call  LCD_EFFACE
            call  Tempo_10ms
            movlw d'0'
            call  LCD_LOCATE
            movlw "C"
            call  LCD_DONNEE
            movlw "A"
            call  LCD_DONNEE
            movlw "L"
            call  LCD_DONNEE
            movlw "I"
            call  LCD_DONNEE
            movlw "B"
            call  LCD_DONNEE
            movlw "R"
            call  LCD_DONNEE
            movlw "E"
            call  LCD_DONNEE
            call  Tempo_200ms
            btfsc B_MOINS
            goto  $+2
            goto  par3
            btfsc B_VALID
            goto  $+2
            goto  par6
            btfsc B_PLUS
            goto  par5
            goto  par7
par6
            call  REGL_CAL
par7
            call  LCD_EFFACE
            call  Tempo_10ms
            movlw d'0'
            call  LCD_LOCATE
            movlw "E"
            call  LCD_DONNEE
            movlw "X"
            call  LCD_DONNEE
            movlw "I"
            call  LCD_DONNEE
            movlw "T"
            call  LCD_DONNEE
            call  Tempo_200ms
            btfsc B_MOINS
            goto  $+2
            goto  par5
            btfsc B_PLUS
            goto  $+2
            goto  par1
            btfsc B_VALID
            goto  par7
            return


REGL_HEU:                       ; Réglage de l'HEURE
```

```
            movf   heure,0             ;
            movwf  cmpt0
Heu1
            movwf  x+0          ; Affichage: "HEURE:" sur la 1ère ligne
            call   LCD_EFFACE
            movlw  d'0'
            call   LCD_LOCATE
            movlw  "H"
            call   LCD_DONNEE
            movlw  "E"
            call   LCD_DONNEE
            movlw  "U"
            call   LCD_DONNEE
            movlw  "R"
            call   LCD_DONNEE
            movlw  "E"
            call   LCD_DONNEE
            movlw  d'7'
            call   AFFICHE_24BS ; Affichage de l'heure courante
            call   Tempo_200ms
Heu2
            btfsc  B_PLUS       ; Défilement de la valeur en incrémentation
            goto   Heu3
            goto   Heu5
Heu3
            btfsc  B_MOINS             ; Défilement de la valeur en décrémentation
            goto   Heu4
            goto   Heu7
Heu4
            btfsc  B_VALID             ; Validation de la valeur pour écriture dans le RTCC
            goto   Heu2
            goto   Heu9
Heu5
            movf   cmpt0,0             ; BP_AV actionné (incrémentation)
            sublw  d'22'        ; Pour limitation à 23 (heure de 0 à 23)
            btfss  CARRY
            goto   Heu6
            incf   cmpt0
            movf   cmpt0,0
            goto   Heu1
Heu6
            movlw  d'0'
            movwf  cmpt0
            goto   Heu1
Heu7
            movf   cmpt0,0             ; BP_AR actionné (décrémentation)
            sublw  d'0'         ; Pour limitation à 0 (heure de 0 à 23)
            btfsc  CARRY
            goto   Heu8
            decf   cmpt0
            movf   cmpt0,0
            goto   Heu1
Heu8
            movlw  d'23'
            movwf  cmpt0
            goto   Heu1
Heu9
            call   I2C_START_    ; BP_VAL actionné (écriture RTCC)
            movlw  ADRS_RTCC+0
            call   I2C_SEND_BYTE
            movlw  0x02
            call   I2C_SEND_BYTE
            movf   cmpt0,0
            CALLX  conv_DH             ; Conversion binaire/Hexa en BCD
            movwf  cmpt0
            call   I2C_SEND_BYTE
            call   I2C_STOP_
            call   Tempo_500ms
            return

REGL_MIN:                       ; Réglage MINUTES ET SECONDES (Secondes=00)

            movf   minute,0
            movwf  cmpt0
Min1
            movwf  x+0          ; Affichage: "MINUTE:" sur la 1ère ligne
```

```
            call   LCD_EFFACE
            movlw  d'0'
            call   LCD_LOCATE
            movlw  "M"
            call   LCD_DONNEE
            movlw  "I"
            call   LCD_DONNEE
            movlw  "N"
            call   LCD_DONNEE
            movlw  "U"
            call   LCD_DONNEE
            movlw  "T"
            call   LCD_DONNEE
            movlw  d'7'
            call   AFFICHE_24BS ; Affichage de la minute courante
            call   Tempo_200ms
Min2
            btfsc  B_PLUS       ; Défilement de la valeur en incrémentation
            goto   Min3
            goto   Min5
Min3
            btfsc  B_MOINS             ; Défilement de la valeur en décrémentation
            goto   Min4
            goto   Min7
Min4
            btfsc  B_VALID             ; Validation de la valeur pour écriture dans le RTCC
            goto   Min2
            goto   Min9
Min5
            movf   cmpt0,0             ; BP_AV actionné (incrémentation)
            sublw  d'58'        ; Pour limitation à 59 (minutes de 0 à 59)
            btfss  CARRY
            goto   Min6
            incf   cmpt0
            movf   cmpt0,0
            goto   Min1
Min6
            movlw  d'0'
            movwf  cmpt0
            goto   Min1
Min7
            movf   cmpt0,0             ; BP_AR actionné (décrémentation)
            sublw  d'0'         ; Pour limitation à 0 (minutes de 0 à 59)
            btfsc  CARRY
            goto   Min8
            decf   cmpt0
            movf   cmpt0,0
            goto   Min1
Min8
            movlw  d'59'
            movwf  cmpt0
            goto   Min1
Min9
            call   I2C_START_    ; B_VALID actionné (écriture RTCC)
            movlw  ADRS_RTCC+0   ; Adressage du circuit MCP7940x
            call   I2C_SEND_BYTE
            movlw  0x00          ; Mémoire 0x00 (RTCSEC)
            call   I2C_SEND_BYTE
            movlw  b'10000000'   ; 0 seconde  et bit ST positionné à 1
            call   I2C_SEND_BYTE; Envoyé à l'adresse 0x00 (RTCSEC)
            movf   cmpt0,0             ;
            call   conv_DH             ; Conversion binaire/Hexa en BCD
            call   I2C_SEND_BYTE; Envoyé à l'adresse suivante 0x01 (RTCMIN)

            call   I2C_START_    ; Start condition
            movlw  ADRS_RTCC+0   ; Adresse du circuit RTCC pour l'écriture
            call   I2C_SEND_BYTE; Envoi de l'adresse du circuit
            movlw  h'03'         ; Adresse du registre "RTCWKDAY"
            call   I2C_SEND_BYTE; Pointage sur l'adresse mémoire h'03'
            movlw  0x09          ; Validation VBATEN et semaine 1
            call   I2C_SEND_BYTE; Envoi de la valeur "00001001" à l'adresse h'03'

            call   I2C_STOP_
            call   Tempo_500ms
            return

REGL_DEPS:
```

```
            movlw  0x00
            call   LEC_EEPROM
            movwf  cmpt0
Heuds1
            movwf  x+0          ; Affichage: "DEPS" sur la 1ère ligne
            call   LCD_EFFACE
            call   Tempo_10ms
            movlw  d'0'
            call   LCD_LOCATE
            movlw  "D"
            call   LCD_DONNEE
            movlw  "E"
            call   LCD_DONNEE
            movlw  "P"
            call   LCD_DONNEE
            movlw  "S"
            call   LCD_DONNEE
            movlw  d'7'
            call   AFFICHE_24BS ; Affichage de l'heure courante
            call   Tempo_200ms
Heuds2
            btfsc  B_PLUS       ; Défilement de la valeur en incrémentation
            goto   Heuds3
            goto   Heuds5
Heuds3
            btfsc  B_MOINS             ; Défilement de la valeur en décrémentation
            goto   Heuds4
            goto   Heuds7
Heuds4
            btfsc  B_VALID             ; Validation de la valeur pour écriture dans EEPROM
            goto   Heuds2
            goto   Heuds9
Heuds5
            movf   cmpt0,0             ; BP_AV actionné (incrémentation)
            sublw  d'6'         ; Pour limitation à 8h
            btfss  CARRY
            goto   Heuds6
            incf   cmpt0
            movf   cmpt0,0
            goto   Heuds1
Heuds6
            movlw  d'5'
            movwf  cmpt0
            goto   Heuds1
Heuds7
            movf   cmpt0,0             ; BP_AR actionné (décrémentation)
            sublw  d'5'         ; Pour limitation à 6h
            btfsc  CARRY
            goto   Heuds8
            decf   cmpt0
            movf   cmpt0,0
            goto   Heuds1
Heuds8
            movlw  d'7'
            movwf  cmpt0
            goto   Heuds1
Heuds9
            movf   cmpt0,0
            movwf  HEUDEPS
            call   ECR_EEPROM0  ; Heure mise en marche sonnerie
            return


REGL_ARRS:
            movlw  0x01
            call   LEC_EEPROM
            movwf  cmpt0
Heuas1
            movwf  x+0          ; Affichage: "ARRS:" sur la 1ère ligne
            call   LCD_EFFACE
            call   Tempo_10ms
            movlw  d'0'
            call   LCD_LOCATE
            movlw  "A"
            call   LCD_DONNEE
            movlw  "R"
            call   LCD_DONNEE
```

```
            movlw  "R"
            call   LCD_DONNEE
            movlw  "S"
            call   LCD_DONNEE
            movlw  d'7'
            call   AFFICHE_24BS ; Affichage de l'heure courante
            call   Tempo_200ms
Heuas2
            btfsc  B_PLUS       ; Défilement de la valeur en incrémentation
            goto   Heuas3
            goto   Heuas5
Heuas3
            btfsc  B_MOINS            ; Défilement de la valeur en décrémentation
            goto   Heuas4
            goto   Heuas7
Heuas4
            btfsc  B_VALID            ; Validation de la valeur pour écriture dans EEPROM
            goto   Heuas2
            goto   Heuas9
Heuas5
            movf   cmpt0,0            ; BP_AV actionné (incrémentation)
            sublw  d'22'        ; Pour limitation à 23h
            btfss  CARRY
            goto   Heuas6
            incf   cmpt0
            movf   cmpt0,0
            goto   Heuas1
Heuas6
            movlw  d'21'
            movwf  cmpt0
            goto   Heuas1
Heuas7
            movf   cmpt0,0            ; BP_AR actionné (décrémentation)
            sublw  d'21'        ; Pour limitation à 21h
            btfsc  CARRY
            goto   Heuas8
            decf   cmpt0
            movf   cmpt0,0
            goto   Heuas1
Heuas8
            movlw  d'23'
            movwf  cmpt0
            goto   Heuas1
Heuas9
            movf   cmpt0,0
            movwf  HEUARRS
            call   ECR_EEPROM1  ; Heure d'arrêt de la sonnerie la nuit
            return


REGL_CAL:                       ; Réglage de la calibration

            call   I2C_START_           ; Bit start condition
            movlw  ADRS_RTCC+0          ; Adresse du circuit RTCC (écriture)
            call   I2C_SEND_BYTE        ; Envoi de l'adresse du circuit RTCC
            movlw  0x08                 ; Adresse de lecture du circuit RTCC
            call   I2C_SEND_BYTE        ; Envoi de l'adresse du 1er octet pour la lecture
            call   I2C_START_           ; Bit start condition
            movlw  ADRS_RTCC+1          ; Adresse du circuit RTCC (lecture)
            call   I2C_SEND_BYTE        ; Envoi de l'adresse du 1er octet pour la lecture
            call   I2C_RECEIVE_BYTE_LAST     ; Lecture du 1er octet à l'adresse h08
            movwf  CALIBRE                   ; Transfère de la valeur lue dans le registre
"seconde"
            call   I2C_STOP_                 ; Stop condition

Cali1
            movwf  x+0                  ; Affichage: "CALIB" sur la 1ère ligne
            call   LCD_EFFACE
            call   Tempo_10ms
            movlw  d'0'
            call   LCD_LOCATE
            movlw  "C"
            call   LCD_DONNEE
            movlw  "A"
            call   LCD_DONNEE
            movlw  "L"
            call   LCD_DONNEE
```

```
            movlw  "I"
            call   LCD_DONNEE
            movlw  "B"
            call   LCD_DONNEE
            movlw  d'7'
            call   AFFICHE_24BS ; Affichage valeur de calibration courante
            call   Tempo_200ms

Cali2
            btfsc  B_PLUS       ; Défilement de la valeur en incrémentation
            goto   Cali3
            goto   Cali5
Cali3
            btfsc  B_MOINS             ; Défilement de la valeur en décrémentation
            goto   Cali4
            goto   Cali7
Cali4
            btfsc  B_VALID             ; Validation de la valeur pour écriture dans le RTCC
            goto   Cali2
            goto   Cali9
Cali5
            movf   CALIBRE,0           ; BP_AV actionné (incrémentation)
            sublw  d'126'       ; Pour limitation à 127 (CALIBRE de 0 à 127)
            btfss  CARRY
            goto   Cali6
            incf   CALIBRE
            movf   CALIBRE,0
            goto   Cali1
Cali6
            movlw  d'0'
            movwf  CALIBRE
            goto   Cali1
Cali7
            movf   CALIBRE,0           ; BP_AR actionné (décrémentation)
            sublw  d'0'         ; Pour limitation à 0 (CALIBRE de 0 à 127)
            btfsc  CARRY
            goto   Cali8
            decf   CALIBRE
            movf   CALIBRE,0
            goto   Cali1
Cali8
            movlw  d'127'
            movwf  CALIBRE
            goto   Cali1
Cali9
            call   I2C_START_    ; BP_VAL actionné (écriture dans le circuit 79401)
            movlw  b'11011110'
            call   I2C_SEND_BYTE
            movlw  0x08
            call   I2C_SEND_BYTE
            movf   CALIBRE,0
            call   I2C_SEND_BYTE; Ecriture valeur de calibration dans le registre 0x08
            call   I2C_STOP_
            call   Tempo_200ms
            return

LEC_EEPROM:
            BANK2
            movwf  EEADR
            BANK3
            bcf    EECON1,7
            bsf    EECON1,0
            BANK2
            movf   EEDAT,w
            BANK0
            return

ECR_EEPROM0:                       ; Ecriture à l'adresse 0x00 de la mémoire EEPROM
            BANK2
            movwf  EEDAT
            movlw  0x00
            movwf  EEADR
            BANK3
            bcf    INTCON,GIE
            bcf    EECON1,EEPGD
            bsf    EECON1,WREN
            btfsc  INTCON,GIE
```

```
            goto   $-2
            movlw  0x55
            movwf  EECON2
            movlw  0xAA
            movwf  EECON2
            bsf    EECON1,WR
            bcf    EECON1,WREN
            BANK0
            call   Tempo_200ms
            return

ECR_EEPROM1:                    ; Ecriture à l'adresse 0x01 de la mémoire EEPROM
            BANK2
            movwf  EEDAT
            movlw  0x01
            movwf  EEADR
            BANK3
            bcf    INTCON,GIE
            bcf    EECON1,EEPGD
            bsf    EECON1,WREN
            btfsc  INTCON,GIE
            goto   $-2
            movlw  0x55
            movwf  EECON2
            movlw  0xAA
            movwf  EECON2
            bsf    EECON1,WR
            bcf    EECON1,WREN
            BANK0
            call   Tempo_200ms
            return

ARRS_INIT:
            BANK2
            movlw  0x00
            movwf  EEADR
            BANK3
            bcf    EECON1,7
            bsf    EECON1,0
            BANK2
            movf   EEDATA,0
            BANK0
            movwf  HEUDEPS
            BANK2
            movlw  0x01
            movwf  EEADR
            BANK3
            bcf    EECON1,7
            bsf    EECON1,0
            BANK2
            movf   EEDATA,0
            BANK0
            movwf  HEUARRS
            return

RTC_INIT:                                   ; Sous-progamme non utilisé
            call   I2C_START_
            movlw  b'11011110'
            call   I2C_SEND_BYTE
            movlw  0x08
            call   I2C_SEND_BYTE
            movlw  d'77'
            call   I2C_SEND_BYTE
            call   I2C_STOP_
            return


;*******************************************************************************

; AFFICHEUR LCD 2x8 en mode 4 bits

LCD_INIT:                       ;Initialisation de l'afficheur
      call   Tempo_30ms
      movlw  h'33'                      ;Function set
      call   LCD_CONTROLE               ;Ecriture
      movlw  h'28'                      ;Function set
      call   LCD_CONTROLE               ;Ecriture
```

```
       movlw  LCD_FUNCTION_SET                ;Function set
       call   LCD_CONTROLE             ;Ecriture
       movlw  LCD_DISPLAY_CONTROL             ;Display ON/OFF
       call   LCD_CONTROLE             ;Ecriture
       call   LCD_EFFACE
       movlw  LCD_ENTRY_MODE_SET              ;Entry mode set
       call   LCD_CONTROLE             ;Ecriture
       return

LCD_CONTROLE:                   ; Envoi d'une commande vers l'afficheur
       bcf    LCD_RS
       call   LCD_4B
       return

LCD_DONNEE:                     ; Envoi d'une donnée vers l'afficheur
       bsf    LCD_RS
       call   LCD_4B
       return

LCD_LOCATE:                     ; Positionnement de la donnée à écrire
       addlw  d'128'
       CALL   LCD_CONTROLE
       return

LCD_EFFACE:                     ; Effacement de l'afficheur
       movlw  h'01'
       call   LCD_CONTROLE
       call   Tempo_1530us
       return

LCD_4B                          ; Transfert d'un nibble (mode 4 bits)
       movwf  LCD_DATA
       bcf           LCD_D4
       bcf           LCD_D5
       bcf           LCD_D6
       bcf           LCD_D7
       btfsc  LCD_DATA,7
       bsf           LCD_D7
       btfsc  LCD_DATA,6
       bsf           LCD_D6
       btfsc  LCD_DATA,5
       bsf           LCD_D5
       btfsc  LCD_DATA,4
       bsf           LCD_D4
       bsf           LCD_E
       nop
       nop
       bcf           LCD_E
       bcf           LCD_D4
       bcf           LCD_D5
       bcf           LCD_D6
       bcf           LCD_D7
       btfsc  LCD_DATA,3
       bsf           LCD_D7
       btfsc  LCD_DATA,2
       bsf           LCD_D6
       btfsc  LCD_DATA,1
       bsf           LCD_D5
       btfsc  LCD_DATA,0
       bsf           LCD_D4
       bsf           LCD_E
       nop
       nop
       bcf           LCD_E
       call   Tempo_11us
       return

;*****************************************************************************

AFFICHE_8BS:              ; Affichage d'une valeur sur 1 octet avec signe

              clrf   x+2           ; Effacement de "x+2"
              clrf   x+1           ; Effacement de "x+1"
              btfss  x+0 , 7              ; Test si "x+0" est négatif
              goto   AFFICHE_24BS ; Si "x+0" est positif, saut vers "AFFICHE_24BS"
              decf   x+2           ; Si "x+0" est négatif, "x+2" = 0xFF
              decf   x+1           ; "x+0" négatif, "x+1" = 0xFF
```

```
;*******************************************************************************

AFFICHE_24BS:                   ; Affichage d'une valeur sur 3 octets avec signe

            movwf  Position           ; Transfert de w dans le registre "Position"
            movlw  0x08
            movwf  OctCompte          ; Valeur "8" dans le registre "OctCompte"
            movlw  0x00
            movwf  y+2          ; Valeur "0" dans le registre "y+2"
            movlw  0x00
            movwf  y+1          ; Valeur "0" dans le registre "y+1"
            movlw  0x0a
            movwf  y+0          ; Valeur 10 (0x0A) dans "y+0"
            bcf    SNEG         ; Mise à 0 du bit "SNEG"
            btfss  x+2 , 7            ; Test si "x" est négatif
            goto   Affiche            ; Si "x" est négatif, saut vers "boucle"
            bsf    SNEG         ; Mise à 1 de "SNEG"
            comf   x+2          ; Complément à 2 de "x" (x en valeur positive)
            comf   x+1          ;
            comf   x+0          ;
            incf   x+0          ;
            skpnz               ;
            incf   x+1          ;
            skpnz               ;
            incf   x+2          ;
Affiche
            CALLX  FXD_2424U          ; Division de x par y (x/y)
            movlw  d'8'
            subwf  Position , w ; w est soustrait de "position" (Position - w)
            addwf  OctCompte , w; "OctCompte" + w ((OctCompte + Position - 8)
            call   LCD_LOCATE   ; Positionnement de l'affichage à (OctCompte + Position - 8)
            movfw  z+0          ; "z+0" transféré dans w
            addlw  0x30         ; w + d'48'
            call   LCD_DONNEE   ; Affichage du caractère code ASCII de (w + 0x30)
            decf   OctCompte          ; Décrémentation de "OctCompte"
            btfsc  ZERO         ; Test si "OctCompte" est égal à 0
            goto   Signe        ; Si "OctCompe" est égal à à, saut vers Etiq "Signe"
            movfw  x+2          ; Si "OctCompte" n'est pas égal à 0,....
            btfss  ZERO         ; Test si (x+2) = 0
            goto   Affiche            ; Si (x=2) n'est pas égal à 0, saut vers Etiq "Boucle"
            movf   x+1,0
            btfss  ZERO         ; Test si (x+1) = 0
            goto   Affiche            ; Si (x+1) n'est pas égal à 0, saut vers Etiq "Boucle"
            movf   x+0,0
            btfss  ZERO         ; Test si (x+0) = 0
            goto   Affiche            ; Si (x+0) n'est pas égal à 0, saut vers Etiq "Boucle"
            movlw  d'8'         ; Si x = 0, 8 dans w
            subwf  Position , w ; w est soustrait de "position" (Position - w)
            addwf  OctCompte , w; "OctCompte" + w ((OctCompte + Position - 8)
            call   LCD_LOCATE   ; Positionnement de l'affichage à (OctCompte + Position - 8)
Signe
            btfss  SNEG         ; Test si bit "SNEG"
            goto   $+3          ; Si x est positif, saut 3 lignes plus loin
            movlw  "-"          ; x négatif, caractère "-" dans w
            call   LCD_DONNEE   ; Affichage du caractère "-"
            incf   Position , w ; Incrémentation du registre "Position" (résultat dans w)
            call   LCD_LOCATE   ; Positionnement de l'affichage à la valeur "Position"
            return

;*******************************************************************************

conv_HD_Heure:                  ; Conversion registres RTCC codés en BCD
                                ; vers le système binaire (ou hexa)
            movfw  seconde
            call   conv_HD
            movwf  seconde
            movfw  minute
            call   conv_HD
            movwf  minute
            movfw  heure
            call   conv_HD
            movwf  heure
            return

conv_HD:                        ; Conversion codage BCD vers binaire (ou vers hexa)
```

```
            movwf  t
            movlx  h'00',h'00',h'00'
            swapf  t,w
            andlw  h'0f'
            movwf  x+0
            movly  h'00',h'00',h'0a'   ; Valeur d'10'
            call   FXM_2424S
            movfw  t
            andlw  h'0f'
            addwf  x+0,w
            return

conv_DH:                       ; Conversion binaire (ou hexa) vers codage en BCD

            clrf   x+2
            clrf   x+1
            movwf  x+0
            movly  h'00',h'00',h'0a'   ; Valeur d'10'
            call   FXD_2424S
            swapf  x+0,w
            addwf  z+0,w
            return
            END                ; directive fin de programme
```