

```

;*****
;
;                               *
;           (C) Gestion ENERGIE PHOTOVOLTAÏQUE pour CEE   Version 1.0 (2017)           *
;
;                               *
;*****

LIST      P=PIC16F88          ; Définition de processeur
#include <pl6f88.inc>         ; fichier include
ERRORLEVEL -302              ; suppression du message bank select
ERRORLEVEL -305              ; suppression du message absence de destination

__CONFIG __CONFIG1, _CP_OFF & _CCP1_RB0 & _DEBUG_ON & _WRT_PROTECT_OFF & _CPD_OFF & _LVP_OFF &
_BODEN_OFF & _MCLR_OFF & _PWRTE_ON & _WDT_OFF & _INTRC_IO

__CONFIG __CONFIG2, _IESO_OFF & _FCMEN_ON

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

OPTIONVAL EQU B'00000000'
INTCONVAL EQU B'00000000'
ANSELVAL EQU B'00001011'
OSCCONVAL EQU B'01100000'
ADCON1VAL EQU B'10000000'

DIRPORTA EQU B'00011111' ; Direction PORTA (1=entrée)(0=sortie)
DIRPORTB EQU B'00000000' ; Direction PORTB (1=entrée)(0=sortie)

;*****
;                               *
;           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'

;*****
;                               *
;           DEFINE
;*****

#DEFINE CARRY STATUS,C ;
#DEFINE ZERO STATUS,Z ;

#DEFINE SYNCHRO PORTA,2 ; Synchro de la tension secteur
#DEFINE JN PORTA,4 ; Contact Jour/Nuit
#DEFINE MOC PORTA,6 ; Commande du triac
#DEFINE LED PORTA,7 ; LED pour signalisation

#DEFINE LCD_PORT PORTB
#DEFINE LCD_E PORTB,1 ; Commande de controle de l'afficheur
#DEFINE LCD_RS PORTB,0 ; Sélection de l'afficheur
#DEFINE LCD_D4 2
#DEFINE LCD_D5 3
#DEFINE LCD_D6 4
#DEFINE LCD_D7 5

#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

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

#include <MACRO.asm>

; OPERATION EN EEPROM
; -----

EEPROM macro ; lire eeprom (adresse dans adeeprom & résultat en w)
    clrwdt ; reset watchdog
    bcf STATUS,RP0 ; passer en banque2
    bsf STATUS,RP1
    movf adeeprom,w ; charger adresse eeprom (passage de la valeur par une variable)
    movwf EEADR ; pointer sur adresse eeprom
    bsf STATUS,RP0 ; passer en banque3
    bcf EECON1,EEPGD ; pointer sur eeprom
    bsf EECON1,RD ; ordre de lecture

```



```

; ////////////////////////////////////////////////////////////////////
;
; P R O G R A M M E
; ////////////////////////////////////////////////////////////////////
;*****
;
; INITIALISATIONS
;*****
init

; initialisation PORTS (banque 0 et 1)
; -----
BANK0
clrf PORTA ; sélectionner banque0
movlw b'00111111' ; Sorties PORTA à 0
movwf PORTB ;
BANK1
movlw DIRPORTA ; passer en banque1
movwf TRISA ; Direction PORTA
movlw DIRPORTB ; écriture dans registre direction
movwf TRISB ; Direction PORTB
; écriture dans registre direction

; Registre d'options (banque 1)
; -----
movlw OPTIONVAL ; charger masque
movwf OPTION_REG ; initialiser registre option
movlw OSCCONVAL ; charger masque
movwf OSCCON
movlw ANSELVAL ; charger masque
movwf ANSEL ; registre du convertisseur A/D
movlw ADCON1VAL ; charger masque
movwf ADCON1 ; registre du convertisseur A/D

; registres interruptions (banque 1)
; -----
movlw INTCONVAL ; charger valeur registre interruption
movwf INTCON ; initialiser interruptions

; Effacer RAM banque 0
; -----
BANK0
movlw 0x20 ; sélectionner banque 0
movwf FSR ; initialisation pointeur
; 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 programme
; -----
call LCD_INIT ; Initialisation de l'afficheur LCD

; Initialisation des facteurs de puissance

movlw d'0'
movwf Puiss+0 ; 0% ( FactPuiss = 0 ) pas de chauffage de l'eau
movlw d'77'
movwf Puiss+1 ; 5% ( FactPuiss = 1 )
movlw d'72'
movwf Puiss+2 ; 10% ( FactPuiss = 2 )
movlw d'68'
movwf Puiss+3 ; 15% ( FactPuiss = 3 )
movlw d'65'
movwf Puiss+4 ; 20% ( FactPuiss = 4 )
movlw d'62'
movwf Puiss+5 ; 25% ( FactPuiss = 5 )
movlw d'60'
movwf Puiss+6 ; 30% ( FactPuiss = 6 )
movlw d'57'
movwf Puiss+7 ; 35% ( FactPuiss = 7 )
movlw d'55'
movwf Puiss+8 ; 40% ( FactPuiss = 8 )
movlw d'53'
movwf Puiss+9 ; 45% ( FactPuiss = 9 )
movlw d'50'
movwf Puiss+0x0A ; 50% ( FactPuiss = 10 )
movlw d'48'
movwf Puiss+0x0B ; 55% ( FactPuiss = 11 )
movlw d'45'
movwf Puiss+0x0C ; 60% ( FactPuiss = 12 )
movlw d'42'
movwf Puiss+0x0D ; 65% ( FactPuiss = 13 )
movlw d'40'
movwf Puiss+0x0E ; 70% ( FactPuiss = 14 )
movlw d'38'
movwf Puiss+0x0F ; 75% ( FactPuiss = 15 )
movlw d'36'

```

```

movwf Puiss+0x10      ; 80% ( FactPuiss = 16 )
movlw d'25'
movwf Puiss+0x11      ; 85% ( FactPuiss = 17 )
movlw d'20'
movwf Puiss+0x12      ; 90% ( FactPuiss = 18 )
movlw d'14'
movwf Puiss+0x13      ; 95% ( FactPuiss = 19 )

goto start            ; Vers programme principal

;*****
;                               PROGRAMME PRINCIPAL                               *
;*****

start
CALL LCD_EFFACE          ; Affichage: "INIT"
movlw d'0'
call LCD_LOCATE
MOVLW "I"
CALL LCD_DONNEE
MOVLW "N"
CALL LCD_DONNEE
MOVLW "I"
CALL LCD_DONNEE
MOVLW "T"
CALL LCD_DONNEE
call Tempo_ls
call ACQ230AC           ; Mesure de la tension du secteur (canal AN1)
call ACQINTPPV         ; Mesure de l'intensité produite par les PPV (canal AN3)

start1
call ACQCELSIUS        ; Mesure de la température de l'eau (canal AN0)
call ACQ230AC          ; Mesure de la tension du secteur (canal AN1)
call ACQINTPPV         ; Mesure de l'intensité produite par les PPV (canal AN3)
call PUISS_PPV         ; Calcul de la puissance produite par les PPV
call PUISS_CEE         ; Calcul de la puissance à appliquer sur le CEE
btfss Etat,0          ; Intensité > 0 ?
goto start6           ; Intensité > à 0: vers régulation JOUR
btfsc JN              ; Contacteur J/N ? La nuit: JN=0 de 0h00 à 8h00
goto start6           ; Intensité > 0: vers régulation JOUR (contact J/N ouvert)
clrf x+2              ; Intensité < 0: vers régulation NUIT (contact J/N fermé)
movfw Celsius+0
movwf x+0
movfw Celsius+1
movwf x+1
movly d'0',d'0',d'200' ; 200 correspond à 20°C (Minimale en régulation NUIT)
comf y+2              ; valeur négative
comf y+1
comf y+0
incf y+0
skpnz
incf y+1
skpnz
incf y+2
call FXA_2424S
btfss x+2,7          ; Test si température < ou > à 20°
goto start2          ; Température > à 20°
bsf Etat,1          ; Température < à 20°. "Etat,1" mis à 1
goto start4          ; Vers start4
bsf Etat,1

start2
btfss Etat,1          ; Test de "Etat,1"
goto start3          ; Valeur=0, vers régulation JOUR
goto start4          ; Valeur=1, vers test température 40°

start3
call Tempo_ls
goto start6

start4
clrf x+2
movfw Celsius+0
movwf x+0
movfw Celsius+1
movwf x+1
movly 0x00,0x01,0x90 ; 400 correspond à 40°C (Maximal en régulation nuit)
comf y+2              ; Valeur négative
comf y+1
comf y+0
incf y+0
skpnz
incf y+1
skpnz
incf y+2
call FXA_2424S          ; Celsius multipliée par 10
btfsc x+2,7          ; Test si résultat négatif
goto start5          ; Température < à 40°
bcf Etat,1          ; Température > à 40°. Etat,1 est mis à 0

goto start6

start5
bsf LED              ; Voyant allumé
call TRIAC_50         ; Régulation à 50%
goto start9          ; Vers affichage et bouclage du programme

start6
; Régulation JOUR

```

```

bcf          LED
clrf        x+2
movfw      Celsius+0
movwf      x+0
movfw      Celsius+1
movwf      x+1
movly     0x00,0x02,0x58 ; 600 correspond à 60°C (température maximale de l'eau)
comf      y+2           ; valeur négative
comf      y+1
comf      y+0
incf      y+0
skpnz
incf      y+1
skpnz
incf      y+2
call     FXA_2424S
btfsc     x+2,7         ; Test si résultat négatif
goto     start8        ; Négatif, Température < à 60°. Vers régulation 57-60
bsf      Etat,2       ; Positif, Température > à 60° Mise à 1 de Etat,2
call     TRIAC_0
goto     start9        ; Température > à 60°, Bouclage au début
start8    ; Température < à 60° (vers régulation 57-60)
btfss     Etat,2
goto     start10       ; Vers régulation suivant PPV
clrf      x+2         ; Vers test si température < ou > à 57°
movfw      Celsius+0
movwf      x+0
movfw      Celsius+1
movwf      x+1
movly     0x00,0x02,0x3A ; 570 correspond à 57°C
comf      y+2           ; valeur négative
comf      y+1
comf      y+0
incf      y+0
skpnz
incf      y+1
skpnz
incf      y+2
call     FXA_2424S
btfss     x+2,7         ; Test si résultat négatif
goto     start12
goto     start11
start12
movf      FactPuiss,0
movwf      x+0
clrf      x+1
clrf      x+2
movly     d'0',d'0',d'2' ; 2 pour facteur de puissance = 2
comf      y+2           ; valeur négative
comf      y+1
comf      y+0
incf      y+0
skpnz
incf      y+1
skpnz
incf      y+2
call     FXA_2424S
btfss     x+2,7
goto     start7        ; Valeur positive, vers puissance 15%
; goto     start11
start11    ; Valeur négative, vers régulation proportionnelle
bcf      Etat,2       ; Mise à 0 de Etat,2
call     TRIAC_0      ; Pas de chauffage
goto     start9        ; Affichage et bouclage au début
start7
call     TRIAC_15     ; Régulation 15%
goto     start9
start10
movlw     0x5C         ; Adresse de départ des facteurs de puissance
movwf     FSR
movfw     FactPuiss
addwf     FSR
movfw     INDF
movwf     TriacOFF    ; Délai d'attente avant déclenchement du TRIAC
call     TRIAC        ; Commande du triac pour chauffage de l'eau
start9
call     AFF_LIGNE1   ; Affichage sur 1ère ligne de l'afficheur
call     AFF_LIGNE2   ; Affichage sur 2ème ligne de l'afficheur
goto     start1

;          FIN DU PROGRAMME PRINCIPAL

;*****
;          SOUS PROGRAMMES          *
;*****

#include <q4mhz.asm>

ACQ230AC:          ; Conversion A/D (acquisition tension secteur)
BANK1
clrf     ADRESL
BANK0

```

```

    clrf    ADRESH
    movlw  b'11001001'
    movwf  ADCON0      ; 2Tosc clock, A/D ON, Canal 1 (AN1)
    bsf    ADCON0,2    ; Mise en marche conversion A/D
    btfsc  ADCON0,2    ; Test fin de la conversion
    goto   $-1
    BANK1
    movfw  ADRESL
    movwf  x+0
    BANK0
    movfw  ADRESH
    movwf  x+1
    movly  d'0',d'0',d'50'      ; 5 V multiplié par 10
    CALL   FXM_2424S
    movly  d'0',d'0',d'102'     ; 1024 (10 bits) divisé par 10
    call   FXD_2424SA
    clrf   x+2
    movfw  x+0
    movwf  TensionAC+0
    movfw  x+1
    movwf  TensionAC+1
    return

ACQINTPPV:      ; Conversion A/D (acquisition intensité PPV)
    clrf   x+2
    BANK1
    clrf   ADRESL
    BANK0
    clrf   ADRESH
    movlw  b'11011001'
    movwf  ADCON0      ; INTclock, A/D ON, Canal3 (AN3)
    bsf    ADCON0,2    ; Mise en marche conversion A/D
    btfsc  ADCON0,2    ; Test fin de la conversion
    goto   $-1
    BANK1
    movfw  ADRESL
    movwf  x+0
    BANK0
    movfw  ADRESH
    movwf  x+1
    movly  d'0',d'0',d'50'      ; 5 V (Vdd) multiplié par 10
    CALL   FXM_2424S
    movly  d'0',d'0',d'102'     ; 1024 (conversion sur 10 bits) / 10
    call   FXD_2424SA
    movly  d'0',d'0',d'248'     ; correspond à 2,5V (tension sortie ACS712 pour I=0)
    comf   y+2              ; valeur négative
    comf   y+1
    comf   y+0
    incf   y+0
    skpznz
    incf   y+1
    skpznz
    incf   y+2
    call   FXA_2424S          ; Résultat: Tension (mesure de I) multipliée par 100
    movly  d'0',d'0',d'135'     ; 151 pour sensibilité 66 mV/A (15,1 A/V)
    CALL   FXM_2424S          ; (135 sensibilité: 74 mV/A)
    movly  d'0',d'0',d'100'     ; redivisé par 100
    call   FXD_2424SA          ; Résultat: Intensité multipliée par 10
    movly  d'0',d'0',d'4'       ; Si I < 0,4 A alors: I = 0
    comf   y+2              ; valeur négative
    comf   y+1
    comf   y+0
    incf   y+0
    skpznz
    incf   y+1
    skpznz
    incf   y+2
    call   FXA_2424S
    btfss  x+2,7             ; Test si I < ou = à 0,4 A
    goto   acqint1           ; Résultat: I > ou = à 0,4 A
    clrf   x+2              ; Résultat: I < à 0,4 A
    clrf   x+1
    clrf   x+0
    bsf    Etat,0
    goto   acqint2

acqint1      ; Intensité > ou = à 0,4 A
    bcf    Etat,0           ; "Etat,0" mis à 0 (I > ou = à 0,4 A)
acqint2
;           ; Test Intensité = 0 si ligne validée
    clrf   x+0
    movfw  x+0
    movwf  IntppvAC+0       ; Intensité multipliée par 10
;           ; Test Intensité = 0 si ligne validée
    clrf   x+1
    movfw  x+1
    movwf  IntppvAC+1
    bsf    Etat,0          ; "Etat,0" mis à 1 (I < à 0,4 A)
    return

ACQCELSIUS:  ; Conversion A/D (acquisition température)
    clrf   x+2
    BANK1
    clrf   ADRESL
    BANK0

```

```

clrf    ADRESH
movlw  b'11000001'
movwf  ADCON0      ; INTclock, A/D ON, Canal0 (AN0)
bsf    ADCON0,2    ; Mise en marche conversion A/D
btfsc  ADCON0,2    ; Test fin de la conversion
goto   $-1
BANK1
movfw  ADRESL
movwf  x+0
BANK0
movfw  ADRESH
movwf  x+1
movfw  x+0
movfw  Celsius+0
movfw  x+1
movfw  Celsius+1
clrf   y+2
movfw  x+1
movfw  y+1
movfw  x+0
movfw  y+0
movlw  0x0F,0x7C,0xD8      ; 1015000 - à faire varier pour l'étalonnage
call  FXD_2424SA
movly  0x00,0x07,0xD0      ; 2000 ohms
comf   y+2                  ; valeur négative
comf   y+1
comf   y+0
incf   y+0
skpznz
incf   y+1
skpznz
incf   y+2
call  FXA_2424S
movly  0x00,0x03,0xE8      ; Valeur 1000
CALL  FXM_2424S
movly  0x00,0x01,0x81      ; 385 (3,85 ohms/°C)
call  FXD_2424SA
movfw  x+0
movfw  Celsius+0          ; Température multipliée par 10
movfw  x+1
movfw  Celsius+1
return

PUISS_PPV:      ; Calcul de la puissance PPV ( P = UI )
clrf   x+2
movfw  IntppvAC+0
movwf  x+0
movfw  IntppvAC+1
movwf  x+1
movfw  TensionAC+0
clrf   y+2
movfw  y+0
movfw  TensionAC+1
movwf  y+1
CALL  FXM_2424S
movly  d'0',d'0',d'100'
call  FXD_2424SA
movly  d'0',d'0',d'10'
CALL  FXM_2424S
movfw  x+0
movfw  Puissppv+0
movfw  x+1
movfw  Puissppv+1
return

PUISS_CEE:      ; Calcul puissance pour CEE et facteur de puissance
movfw  Puissppv+0          ; Puissance PPV transférée vers "x"
movwf  x+0
movfw  Puissppv+1
movwf  x+1
movly  d'0',d'0',d'100'
call  FXD_2424U          ; Calcul du facteur de puissance
movfw  x+0
movfw  FactPuiss
movly  d'0',d'0',d'100'
call  FXM_2424S
movfw  x+0
movfw  Puisscee+0          ; Transfert de la valeur vers "Puisscee"
movfw  x+1
movfw  Puisscee+1

puis2
return

TRIAC:          ; Commande du TRIAC (signaux PWM)
bcf    STATUS,Z
movfw  FactPuiss
btfss  ZERO
goto   TRIAC1
call  Tempo_500ms
return

TRIAC1:

```

```

        movlw 0xFF
        movwf MEM2
triac2
        btfss SYNCHRO
        goto $-1
        btfsc SYNCHRO
        goto $-1
        movfw TriacOFF
        movwf MEM1
triac3
        Call    Tempo_97us
        decfsz MEM1
        goto   triac3
        bsf    MOC
        Call    Tempo_97us
        bcf    MOC
        decfsz MEM2
        goto   triac2
        Return

TRIAC_0:
        clrf   Puisscee+0
        clrf   Puisscee+1
        call   Tempo_1s
        return

TRIAC_15:
        movlw d'77'           ; 77 pour puissance à 5% 100W (température entre 56° et 60°)
        movwf TriacOFF
        call   TRIAC1
        movlx 0x00,0x00,0x64 ; pour affichage puissance = 100 W
        movfw x+0
        movwf Puisscee+0     ; Transfert de la valeur vers "Puisscee"
        movfw x+1
        movwf Puisscee+1
        Return

TRIAC_50:
        movlw d'50'           ; 50 pour puissance à 50% (température inférieure minimum)
        movwf TriacOFF
        call   TRIAC1
        movlx 0x00,0x03,0xE8 ; pour affichage puissance = 1000 W
        movfw x+0
        movwf Puisscee+0     ; Transfert de la valeur vers "Puisscee"
        movfw x+1
        movwf Puisscee+1
        Return

AFF_LIGNE1:
        call   LCD_EFFACE     ; Affichage sur 1ère ligne de l'afficheur
        clrf   x+2
        movfw TensionAC+0
        movfw x+0
        movfw TensionAC+1
        movwf x+1
        movlw d'2'
        CALLX  AFFICHE_24BS
        movlw d'3'
        call   LCD_LOCATE
        movlw "v"
        call   LCD_DONNEE
        MOVLW d'7'
        CALL   LCD_LOCATE
        MOVLW "p"
        CALL   LCD_DONNEE
        MOVLW "p"
        CALL   LCD_DONNEE
        MOVLW "v"
        CALL   LCD_DONNEE
        MOVLW ":"
        CALL   LCD_DONNEE
        movfw Puissppv+0
        movfw x+0
        movfw Puissppv+1
        movwf x+1
        movlw d'14'
        CALLX  AFFICHE_24BS
        movlw d'15'
        call   LCD_LOCATE
        movlw "w"
        call   LCD_DONNEE
        return

AFF_LIGNE2:
        movfx Celsius
        movly d'0',d'0',d'10'
        CALLX  FXD_2424S
        movfw z+0
        movwf MEM1
        movlw d'65'
        CALLX  AFFICHE_24BS
        MOVLW d'66'

```

```

CALL    LCD_LOCATE
movlw   ", "
CALL    LCD_DONNEE
movfw   MEM1
movwf   x+0
movlw   d'67'
CALLX   AFFICHE_24BS
MOVLW   0xB2
CALL    LCD_DONNEE
MOVLW   "C"
CALL    LCD_DONNEE
movlw   d'71'
call    LCD_LOCATE
movlw   "P"
call    LCD_DONNEE
movlw   "C"
call    LCD_DONNEE
movlw   "E"
call    LCD_DONNEE
movlw   ":"
call    LCD_DONNEE
movfw   Puisscee+0
movwf   x+0
movfw   Puisscee+1
movwf   x+1
movlw   d'78'
CALLX   AFFICHE_24BS
movlw   d'79'
call    LCD_LOCATE
movlw   "W"
call    LCD_DONNEE
return

; TEMPORISATION

Tempo_97us:
movlw   d'32'           ; 1 cycle
movwf   templ          ; 1 cycle
Tempo_97:
decfsz  templ,1        ; 2 cycles si templ=0 sinon 1 cycle
goto    Tempo_97       ; 2 cycles
return

; AFFICHEUR LCD 2x16 en mode 4 bits

LCD_INIT:
call    Tempo_30ms     ;Initialisation de l'afficheur
movlw   h'33'           ;Function set
call    LCD_CONTROLE   ;Ecriture
movlw   h'32'           ;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
movlw   LCD_CURSOR_DISPLAY_SHIFT
return

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

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

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

LCD_EFFACE:
movlw   h'01'          ; Effacement de l'afficheur
call    LCD_CONTROLE
call    Tempo_530us
call    Tempo_530us
return

LCD_4B
movfw   LCD_PORT       ; Transfert d'un nibble (mode 4 bits)
movwf   LCD_PORT_SAVE

```

```

bcf          LCD_PORT_SAVE,LCD_D4
bcf          LCD_PORT_SAVE,LCD_D5
bcf          LCD_PORT_SAVE,LCD_D6
bcf          LCD_PORT_SAVE,LCD_D7
btfsc       LCD_DATA,4
bsf         LCD_PORT_SAVE,LCD_D4
btfsc       LCD_DATA,5
bsf         LCD_PORT_SAVE,LCD_D5
btfsc       LCD_DATA,6
bsf         LCD_PORT_SAVE , LCD_D6
btfsc       LCD_DATA,7
bsf         LCD_PORT_SAVE,LCD_D7
movfw       LCD_PORT_SAVE
movwf       LCD_PORT
bsf         LCD_E
bcf         LCD_E
call        Tempo_39us
return

```

```

;*****

```

```

AFFICHE_8BS:          ; Affichage d'une valeur sur 1 octet avec signe
                    ; Effacement de "x+2"
    clrfs         x+2
                    ; Effacement de "x+1"
    clrfs         x+1
                    ; Test si "x+0" est négatif
    btfss        x+0 , 7
                    ; Si "x+0" est positif, saut vers "AFFICHE_24BS"
    goto         AFFICHE_24BS
                    ; Si "x+0" est négatif, "x+2" = 0xFF
    decf         x+2
                    ; "x+0" négatif, "x+1" = 0xFF
    decf         x+1

```

```

;*****

```

```

AFFICHE_24BS:        ; Affichage d'une valeur sur 3 octets avec signe
                    ; Transfert de w dans le registre "Position"
    movwf       Position
                    ; Valeur "8" dans le registre "OctCompte"
    movlw       0x08
    movwf       OctCompte
                    ; Valeur "0" dans le registre "y+2"
    movlw       0x00
    movwf       y+2
                    ; Valeur "0" dans le registre "y+1"
    movlw       0x00
    movwf       y+1
                    ; Valeur 10 (0x0A) dans "y+0"
    movlw       0x0a
    movwf       y+0
                    ; Mise à 0 du bit "SNEG"
    bcf         SNEG
                    ; Test si "x" est négatif
    btfss       x+2 , 7
                    ; Si "x" est négatif, saut vers "boucle"
    goto         Affiche
                    ; Mise à 1 de "SNEG"
    bsf         SNEG
                    ; Complément à 2 de "x" (x en valeur positive)
    comf        x+2
                    ;
    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         ; Décrémenter de "OctCompte"
btfsc       ZERO              ; Test si "OctCompte" est égal à 0
goto         Signe            ; Si "OctCompte" est égal à 0, saut vers Etiquette "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 Etiquette "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 Etiquette "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 Etiquette "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      ; Incrémenter du registre "Position" (résultat dans w)
call         LCD_LOCATE        ; Positionnement de l'affichage à la valeur "Position"
return

```

```

; org 0x0800

```

```

#include <MATH.asm>

```

END

; directive fin de programme