

BIBLIOTHEQUE TIME CANSI pour PIC18

La bibliothèque TIME procure toutes les fonctions CANSI de traitement de l'heure et de la date. Elle utilise deux variables : **t_time timer** et **t_time clocker**

timer représente l'heure calendaire (le nombre de secondes depuis le 1/1/1970) et **clocker** l'heure processeur (le nombre de secondes comptés par le processeurs). Ces deux variables doivent être incrémentées toutes les secondes (par exemple grâce à un TIMER du PIC)

Le fichier d'en-tête <time.h> contient des déclarations de types et de fonctions servant à manipuler la date et l'heure. Certaines fonctions traitent *l'heure locale*, qui peut être différente de l'heure calendaire (fuseaux horaires, décalage solaire, heure d'été). clock_t et time_t sont des types arithmétiques qui représentent des instants, et struct tm contient les composantes d'une heure calendaire :

int tm_sec;	secondes après la minute (0-61)
int tm_min;	minutes après l'heure (0-59)
int tm_hour;	heures depuis minuit (0-23)
int tm_mday;	jour du mois (1-31)
int tm_mon;	mois <i>depuis</i> janvier (0-11)
int tm_year;	année depuis 1900
int tm_wday;	jours depuis dimanche (0-6)
int tm_yday;	jours depuis le premier janvier (0-365)
int tm_isdst;	drapeau de l'heure d'été

tm_isdst est positif si l'heure d'été est en vigueur, nul sinon, et négatif si cette information n'est pas disponible.

Fonctions de la bibliothèque :

clock_t clock(void)

clock retourne le temps d'utilisation du processeur par le programme depuis le début de son exécution, ou bien -1 si cette information n'est pas disponible. clock () /CLK TCK est une durée en secondes.

time_t time(time_t *tp)

time retourne l'heure calendaire actuelle, ou bien -1 si l'heure n'est pas disponible. Si tp est différent de NULL, *tp reçoit aussi cette valeur de retour.

double difftime(time_t time2, time_t time1)

difftime retourne la durée time2-time1, exprimée en secondes.

time_t mktime(struct tm *tp)

mktime convertit l'heure locale contenue dans la structure *tp en heure calendaire, exprimée suivant la même représentation que celle employée par time. Les valeurs des composantes de l'heure seront comprises dans les intervalles donnés ci-dessus. La fonction mktime retourne l'heure calendaire, ou bien -1 si celle-ci ne peut pas être représentée.

Les quatre fonctions suivantes retournent des pointeurs sur des objets statiques qui peuvent être écrasés par d'autres appels.

char *asctime(const struct tm *tp)

asctime convertit l'heure représentée dans la structure *tp en une chaîne de la forme :

Sun Jan 3 15:14:13 1988\n\0

char *ctime(const time_t *tp)

ctime convertit l'heure calendaire *tp en heure locale ; elle équivaut à : asctime(localtime(tp))

struct tm *gmtime(const time_t *tp)

gmtime convertit l'heure calendaire en Temps Universel (TU). Elle retourne NULL si le TU n'est pas disponible. Cette fonction s'appelle gmtime pour des raisons historiques.

struct tm *localtime(const time_t *tp)

localtime convertit l'heure calendaire *tp en heure locale.

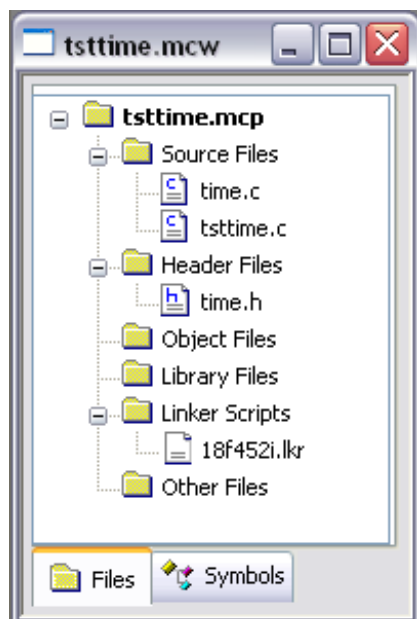
size_t strftime(char *s, size_t smax, const char *fmt, const struct tm *tp)

strftime transforme les informations de date et d'heure contenues dans *tp suivant le format fmt, qui est analogue à un format de printf, et place le résultat dans s. Les caractères ordinaires (y compris le '\0' final) sont copiés dans s. Chacun des % est remplacé comme indiqué ci-dessous, en utilisant des valeurs adaptées à l'environnement [local. strftime](#) place au maximum smax caractères dans s. Cette fonction retourne le nombre de caractères de s, non compris le '\0', ou bien zéro si elle a produit plus de smax caractères.

%a	abréviation du jour de la semaine.
%A	nom complet du jour de la semaine.
%b	abréviation du mois.
%B	nom complet du mois.
%c	représentation locale de la date et de l'heure.
%d	jour du mois (01-31).
%H	heure (sur 24 heures) (00-23).
%I	heure (sur 12 heures) (01-12).
%j	numéro du jour dans l'année (001-366).
%m	numéro du mois (01-12).
%M	minutes (00-59).
%p	équivalent local de AM ou PM (matin ou après-midi). %Ssecondes (00-59).
%U	numéro de la semaine dans l'année (en prenant le dimanche comme premier jour de la semaine) (00-53).
%w	numéro du jour de la semaine (0-6, 0 représentant le dimanche).
%W	numéro de la semaine dans l'année (en prenant le lundi comme premier jour de la semaine) (00-53).
%x	représentation locale de la date.
%X	représentation locale de l'heure.
%y	année dans le siècle (00-99). année, y compris le siècle.
%Z	nom du fuseau horaire, s'il existe.

Demo de la bibliothèque :

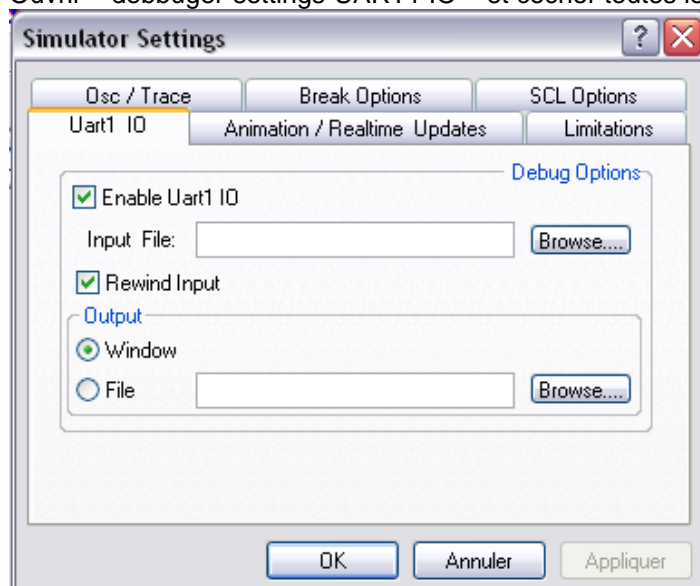
Le fichier tstTIME.zip contient tous les fichiers nécessaires, ouvre le projet tsttime avec MPLAB (V7.3x)



Time .c contient les fonctions de la librairie, time.c peut être converti en .lib
Time.h contient les #define, #include et prototypes de la bibliothèque.
Tsttime.c est un programme exemple, de teste de la bibliothèque, il retourne sur l'UART du PIC la date, l'heure, l'heure calendaire et l'heure processeur, avec comme base de temps le quartz du PIC (ici à 4MHz)
La seconde est mesurée par le TIMER1 en interruption sur débordement, il est également possible d'utiliser un quartz 32Khz externe.

L'heure et la date sont initialisées dans l'exemple comme suit :
15h25mn10s le vendredi 25 mai 2006.

Le programme peut être testé à l'aide du simulateur MPLAB :
Ouvrir « debugger-settings-UART1 IO » et cocher toutes les cases.



Après compilation et lancement du programme :

