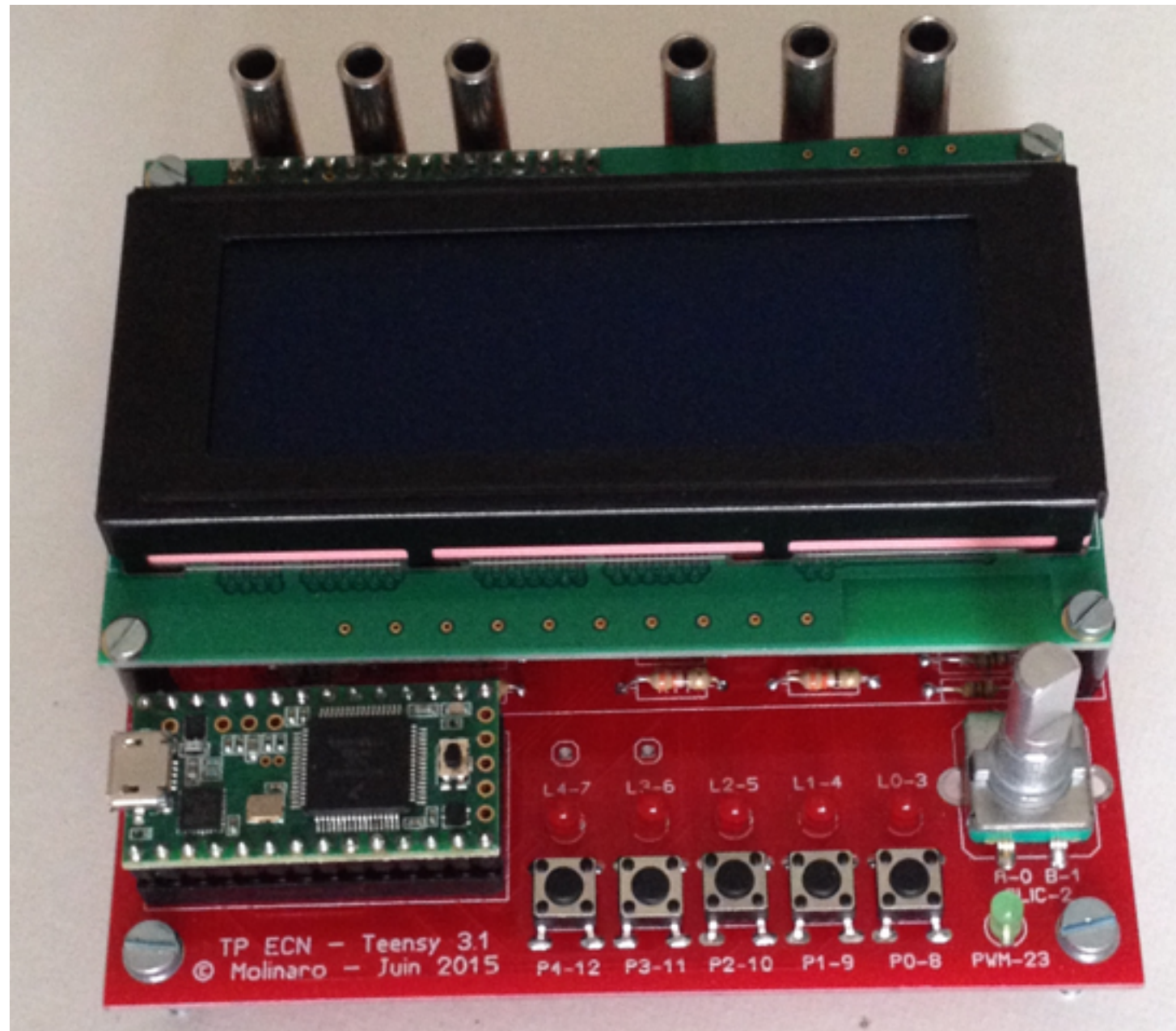


Temps Réel



But de cette partie

Objectif :

- *ajout de la fin des tâches.*

Travail à faire :

- reprendre le programme précédent, modifier les tâches de façon qu'elle se terminent.

Fin des tâches

La fin des tâches est prise en charge par un nouveau service, `task_self_terminate`. Celui-ci est implémenté via un *Service Call* par la fonction suivante :

```
void svc_task_self_terminate (KERNEL_MODE) {  
    task_descriptor * rt = kernel_runningTask (MODE) ;  
    rt->mState = TASK_TERMINATED ;  
    kernel_makeNoTaskRunning (MODE) ;  
}
```

L'appel est effectué automatiquement lorsque la routine exécutant le code de la tâche termine.

```
static void task_entry (USER_MODE_  
                       routineTaskType inTaskRoutine,  
                       const uint32_t inArgument1,  
                       const uint32_t inArgument2) {  
    //--- Run user code  
    inTaskRoutine (MODE_ inArgument1, inArgument2) ;  
    //--- Self terminate  
    task_self_terminate (MODE) ;  
}
```

Ces deux routines sont à ajouter / modifier dans `task-descriptor.c`.

Fonctionnement du svchandler (étape 13)

Le pointeur *gRunningTaskContextSaveAddress* désigne le contexte de la tâche en cours

Allumer led *Teensy*

Si ce pointeur vaut NULL, il n'y a pas de tâche en cours. Initialement, il vaut NULL.

Ancien := *gRunningTaskContextSaveAddress*

Fichier *svc-handler-prgm-13.s* fourni sur le serveur pédagogique.

Appeler *svc_XXX*

Appeler *kernel_selectTaskToRun*

gRunningTaskContextSaveAddress != NULL ?

Ancien != *gRunningTaskContextSaveAddress*

non

non

oui

oui

non

Ancien != NULL ?

oui

Sauvegarder ancien contexte

Restituer nouveau contexte

Restituer contexte tâche de fond

Retour d'exception

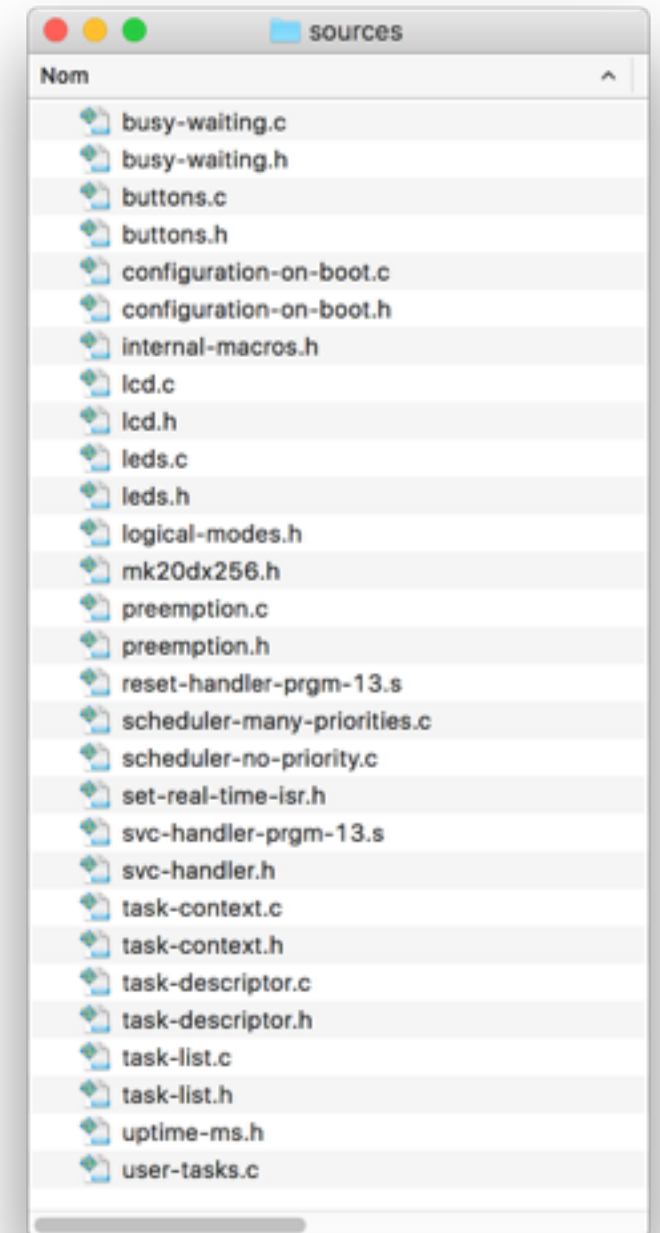
Retour d'exception

La tâche de fond (*reset-handler-prgm-13.s*) boucle sur l'extinction de la led *Teensy* et l'instruction *wfi* (Wait For Interrupt).

Travail à faire

- récupérer sur le serveur pédagogique l'archive 13-sources.tbz qui contient :
 - *reset-handler-prgm-13.s ;
 - *svc-handler-prgm-13.s ;
- écrire le service task_self_terminate en complétant task-descriptor.c et svc-handler-prgm-13.s ;
- modifier user-tasks.c, de façon que les tâches qui font chacune clignoter une led se terminent.

Rappel : la led *Teensy* est contrôlée par l'exécutif, de façon à montrer la charge du processeur.



Résultat attendu

Les leds doivent clignoter alternativement, jusqu'à ce que les tâches se terminent. À ce moment, la led Teensy doit s'éteindre complètement.