# Laboratoire 4: Programme modulaire de communication série avec base de temps (sommatif)

Le laboratoire 3 vous a fait travailler à un programme qui utilise un bouton et un compteur du STM32 pour faire clignoter une DEL. Vous devez d'abord compléter cet exercice sommatif avant de faire les autres parties sommatives du laboratoire. L'étape D de ce laboratoire est formative.

## Matériel requis:

Ensemble de développement STM32 et logiciel S-0001 disponible sur le site du cours

## À réaliser:

Réaliser le travail qui est décrit dans les sections suivantes en vous assurant de respecter la nomenclature utilisée et remettre les programmes à l'enseignant avant la prochaine période de laboratoire.

### Partie A: Programme de test du module piloteTimer6Up (formatif lors du laboratoire 3)

Créez un programme de test qui réutilise du code provenant des programmes S-3000 et S-3001 dans les premières parties du laboratoire afin de créer un programme de test du module "piloteTimer6Up". Le programme devra faire clignoter une DEL ou la garder éteinte selon ce que commande l'état d'un bouton:

* Faites une autre copie du répertoire S-0001 qui vous a été fourni en classe ;
* Modifiez les fichiers qui définissent le projet et l'environnement de travail de même que les répertoires que vous venez de copier pour disposer du projet S-3002 qui vous servira à créer le deuxième programme du laboratoire (i.e. programme S-3002) ;
* Identifiez les fichiers qui forment le module à tester et consulter le fichier ".h" du module pour savoir comment l'utiliser ;
* Épurez le projet S-3001 pour qu'il ne contienne plus que le module à tester et le module et "main" en plus de ce qui fait partie des fichiers produits par ST (HAL, etc…) ;
* Modifiez votre projet pour que les fichiers des programmes S-3000 et S-3001 d'intérêts fassent partie des fichiers du projet S-3001 qui sont compilés par ce projet ;
* Épurez et modifiez les fichiers "main.c" et "main.h" pour créer le programme de test S-3002 qui vous permettra de tester le module "piloteTimer6Up":
  + La DEL T1 doit être éteinte quand on n'appuie pas sur le bouton B1 ;
  + La DEL T1 doit clignoter quand on appuie sur le bouton B1 ;
  + La DEL T1 doit rester allumer pendant 2 secondes avant d'éteindre quand elle clignote ;
  + La DEL T1 doit rester éteinte pendant 2 secondes avant d'allumer quand elle clignote ;

Le programme S-3002 ne doit pas utiliser de services de base de temps puisqu'il sert à tester le module qui servirait à mettre la base de temps en marche…

## Partie B: Faire clignoter une DEL en se servant d'une base de temps.

Le programme S-3002 que vous venez de créer utilise le piloteTimer6Up pour faire clignoter une DEL. Il ne dispose toutefois pas d'un service de base de temps qui lui permettrait de faire facilement plusieurs tâches en même temps. Vous devez donc créer le programme S-4000 qui vous mettra en place une telle base de temps. Pour y parvenir:

* Faites une copie de votre projet S-3002 et renommez le S-4000
* Faites une copie des fichiers *serviceBaseDeTemps.c* et *serviceBaseDeTemps.h* du projet S-0001 et placer ces copies dans le répertoire *Src* de votre projet S-4000
* Ajoutez les fichiers *serviceBaseDeTemps.c* et *serviceBaseDeTemps.h* au projet pour que IAR en tienne compte lorsqu'il compilera le projet (indice: "add")
* Prenez connaissance des informations qui se trouve dans le fichier *serviceBaseDeTemps.h* et consultez le fichier *main.h* du projet S-0001 pour savoir comment utiliser le module *serviceBaseDeTemps* (ex. comment définir le nombre de phases, le nombre d'interruptions par secondes, etc.)
* Assurez-vous de faire les définitions qu'il faut dans le fichier *main.h* du projet S-4000 pour qu'il y ait 2000 interruptions par secondes lorsque le programme s'exécutera
* Modifiez le fichier *main.h* pour indiquez que la base de temps n'aura qu'une phase à gérer
* Modifiez le fichier *main.c* pour qu'il inclue les fichiers du module *serviceBaseDeTemps*
* Modifiez le fichier *main.c* pour qu'il fasse l'initialisation du module *serviceBaseDeTemps*
* Le programme S-3002 utilisait le pointeur de fonction *piloteTimer6Up\_execute* pour faire appel à une fonction qui gérait la DEL et le bouton, faites-en sorte qu'il utilise plutôt le pointeur de fonction *serviceBaseDeTemps\_execute[0]* pour faire appel à cette fonction (pensez à tenir compte du fait qu'il y a 2000 interruptions par seconde)
* Faites compiler et exécuter votre programme pour constater que le bouton peut encore affecter le clignotement de la DEL comme il le fait pour le programme S-3002
* Créez un fichier *zip* qui contiendra tous les fichiers et les modules nécessaires à la compilation du programme (ne pas oublier les pilotes, etc…). Vous le ferez parvenir à l'enseignant par LÉA avant le prochain laboratoire.

## Partie C: Faire clignoter 2 DEL en se servant d'une base de temps.

Le programme S-4000 que vous venez de créer utilise le piloteTimer6Up pour faire clignoter une DEL. Si vous définissez 2 phases de temps, il pourra gérer 2 DEL en même temps. Vous devez maintenant créer le programme S‑4001 à partir du programme S‑4000 et faire en sorte qu'une deuxième DEL de l'ensemble STM32 clignote à une fréquence qui sera nettement différente de celle à laquelle la première le fait. Pour y parvenir:

* Faites une copie de votre projet S-4000 et renommez le S-4001
* Effacez les fichiers *serviceBaseDeTemps.c* et *serviceBaseDeTemps.h* du projet S-4001 et faites en sorte que le projet utilise les fichiers qui font partie du projet S-4000 à la place (voir l'annexe)
* Modifiez le programme pour que le module *serviceBaseDeTemps* gére 2 phases
* Modifiez le fichier *main.h* pour y définir que la phase de calcul "0" se nomme MAIN\_PHASE\_T1 (un #define)
* Modifiez le fichier *main.h* pour y définir que la phase de calcul "1" se nomme MAIN\_PHASE\_T2
* Copiez les fichiers des modules *piloteIOT1* et *interfaceT1* dans le répertoire *Src* du projet S-4001
* Renommez les fichiers que vous venez de copiez pour créer les modules *piloteIOT2* et *interfaceIOT2*
* Modifiez le contenu des fichiers des *piloteIOT2* et *interfaceIOT2* pour remplacer tous les T1 par des T2
* Modifiez les fichiers *main.h* et *main.c* et le projet de IAR pour que IAR tienne compte des modules que vous venez de créer
* Modifiez votre fichier *main.c* pour que le pointeur de fonction *serviceBaseDeTemps[MAIN\_PHASE\_T1]* serve à gérer la première DEL et le bouton comme avant
* Modifiez votre fichier *main.c* pour que le pointeur de fonction *serviceBaseDeTemps[MAIN\_PHASE\_T2]* serve à faire clignoter une deuxième DEL à une fréquence de 3 Hz.
* Faites compiler et exécuter votre programme pour constater que la deuxième DEL peut clignoter en même temps que le bouton peut affecter le clignotement de la DEL comme il le fait pour le programme S-4000
* Modifiez le fichier *zip* que vous avez créé à l'étape B pour qu'il contienne tous les fichiers et les modules nécessaires à la compilation du programme et faites-le parvenir à l'enseignant (ne pas oublier les pilotes, etc…) avant le prochain laboratoire

## Étape D: Transmettre des octets en plus de faire clignoter des DEL

Créez le projet S-4002 et Inspirez-vous de ce que vous avez fait à l'étape C pour ajouter une phase au programme pendant laquelle il y aura transmission d'un octet par un port série à toute les secondes.

# Annexe:

Le laboratoire vous demande d'en savoir plus à propos de IAR que les laboratoires précédents. Les informations qui suivent vous seront utiles pour le réaliser.

## Pour copier un projet IAR:

IAR ne permet pas de copier et de renommer un projet facilement. Vous devez suivre les étapes suivantes pour y parvenir:

1. Copiez et renommez le répertoire du projet (ex. copier S-0001 et le renommer S-3000)
2. Renommez les sous-répertoires dont le nom n'est pas approprié (ex. renommer 247‑637 S‑0001 V7 pour 247-637 S-3000 V7)
3. Ouvrez le sous-répertoire EWARM du projet (une procédure semblable s'applique au cas du compilateur pour 8051)
4. Effacer le répertoire "settings" et l'autre répertoire qui porte le nom du programme produit par le projet original (ex. 247‑637 S‑0001 V7)
5. Effacer les fichiers ".ewd", ".ewt" et ".dep"
6. Éditez les fichier ".eww" et ".ewp" pour corriger toutes les références au nom du projet original. Utilisez CTRL+H pour vous faciliter la tâche (ex. CTRL+H et remplacer S-0001 par S-3000 et faire "replace all"
7. Ouvrez votre projet IAR nouvellement créé (ex. S-3000) et faites un clic droit sur le nom du programme (ex. 247-637 S-3000 V7) et sélectionnez "Options…"
8. Cliquez sur l'élément "Debugger" et sélectionnez "ST-LINK" dans le menu déroulant "Driver" et faites OK
9. Cliques sur l'élément "ST-LINK" et cochez la case "SWD"

## Pour ajouter une référence à un répertoire qui contient des fichiers à inclure

Pour éviter d'avoir plusieurs copies d'un même fichier avec tous les problèmes de maintenance qui viennent avec l'emploi d'une multitude de copies d'un même fichier, faites les opérations suivantes:

1. Dans IAR, faites un clic droit sur le nom du programme (ex. 247-637 S-3000 V7) et sélectionnez "Options"
2. Sélectionnez l'item "C/C++ Compiler" et sélectionnez ensuite l'onglet "Preprocessor" de la fenêtre qui s'est ouverte
3. Cliquez sur la case qui contient trois points à la droite de l'écran pour ajouter un répertoire de référence au projet
4. Dans la fenêtre qui s'est ouverte, cliquez sur l'item "<Click to add>"
5. Sélectionnez le répertoire de référence dans lequel se trouve les fichiers d'intérêts et faites Ok
6. Cliquez sur le nom du répertoire que vous venez d'ajouter à la liste des répertoires pris en compte par IAR pour le projet
7. Cliquez sur la case qui contient un triangle inversé pour pouvoir modifier le nom complet du répertoire de référence que vous venez d'ajouter
8. Parmi les choix qui vous ont offerts par la fenêtre qui vient de s'ouvrir, sélectionnez celle qui commence par "$PROJ\_DIR$\.."
9. Faites les clics qu'il faut pour compléter l'opération.