

## Laboratoire 01 – Accès aux I/O du HPS (18.02.2026)

### Objectifs du laboratoire

Ce laboratoire d'introduction a pour but de vous familiariser avec le flow complet pour la réalisation d'un système numérique utilisant un HPS. Le contenu réalisé lors de cette introduction sera réutilisé par le prochain laboratoire.

### Spécifications

Le but est de communiquer avec les périphériques du HPS. Vous utiliserez le bouton utilisateur du HPS ainsi que la LED utilisateur du HPS et utiliserez un timer afin de pouvoir mesurer des intervalles de temps.



Vous créerez un système permettant de faire clignoter la LED un certain nombre de fois et à une certaine fréquence lors de l'appui sur le bouton

Voici le mapping du bouton et de la LED sur le HPS :

Signal Name	HPS GPIO	Register/bit	Function
HPS_KEY	GPIO54	GPIO1[25]	I/O
HPS_LED	GPIO53	GPIO1[24]	I/O

## **Travail demandé**

- 1) Récupérer le repo git du projet avec l'adresse  
[https://reds-gitlab.heig-vd.ch/reds-public/scf\\_2026.git](https://reds-gitlab.heig-vd.ch/reds-public/scf_2026.git)
- 2) Créer un projet Quartus (version 18.1) dans le dossier « hard/eda » de la hiérarchie fournie, avec le nom « DE1\_SoC » et « DE1\_SoC\_top » comme nom du top. Voir document « Tutoriel des outils de conception ».
- 3) Ouvrir Qsys, configurer le HPS avec les 2 GPIOs nécessaires et générer le code HDL, voir tutoriel. Dans l'onglet « HPS Clocks » de la fenêtre d'édition du HPS, vérifier que *EOSC1* a bien une fréquence de 25MHz
- 4) Ajouter les fichiers sources au projet Quartus (voir tutoriel) :
  - « DE1\_SoC\_top.vhd » (dans « hard/src »)
  - « DE1\_SoC.sdc » (dans « hard/eda »)
  - « qsys\_system.qip » (stocké dans « hard/eda/qsys\_system/synthesis » lors de la génération du point précédent)
- 5) Réaliser l'assignation des pins.  
Menu *Tools* → *Tcl Scripts...*, sélectionner « DE1\_SoC\_assign\_pins.tcl », puis cliquer sur *Run*.
- 6) Synthétiser le projet.
- 7) Compléter le fichier « hps\_gpio.c » dans le dossier « soft/src ».  
Au lancement du programme, vous devez demander à l'utilisateur combien de fois il souhaite voir la LED clignoter ainsi que le temps durant lequel la LED doit rester allumée/éteinte à chaque fois puis . Vous trouverez plus d'information pour les accès aux ressources dans [le manuel technique du CycloneV](#) et dans la [table des registres en ligne](#). Utilisez-les pour vous aider à compléter le programme afin que celui-ci fasse clignoter la LED le nombre de fois demandé à la fréquence demandée lors de l'appui sur le bouton.  
Nous utiliserons le timer OSC1 TIMER0. Comme nous ne gérons pas encore les interruptions, vous devez trouver comment l'utiliser sans en générer
- 8) Créer un projet ARM-DS à placer dans le dossier « soft/proj ». Voir tutoriel.
- 9) Observer le contenu de la mémoire du HPS à l'adresse du bouton et de la LED utilisateur. Voir tutoriel.
- 10) Compiler le code et le tester sur la carte DE1-SoC. Voir tutoriel.
- 11) Faire valider votre montage par le professeur ou l'assistant.

## **À rendre**

Ce laboratoire n'est pas évalué. Il n'y a pas de rapport à rédiger. Un contrôle du travail sera effectué. Vous devez rendre une archive avec les sources du projet pour Quartus, Qsys et le programme C. Utiliser le Makefile à la racine du projet pour générer votre archive à rendre en tapant « make zip » dans un terminal.