HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Reconfigurable & embedded Digital Systems

Prof. Yann Thoma

Laboratoire de Programmation Temps Réel

semestre automne 2019 - 2020

Laboratoire 6 : Audio recorder

Temps à disposition : 6 périodes

Le laboratoire peut être effectué par groupe de 2 personnes.

Objectifs

Le but de ce laboratoire est de réaliser un enregistreur audio capable de rejouer ce qui vient d'être enregistré, et ce sur une carte DE1-SoC.

Vue du système

La DE1-SoC possède une interface audio accessible depuis la partie logique programmable. Le système qui vous est fourni contient la partie logique ainsi qu'un driver spécialement développé pour ce laboratoire. Ce driver vous permet d'envoyer des données sur les deux canaux audios. En interne il dispose d'un buffer de 128 mots par canal. De plus, il permet de piloter et scruter les différentes E/S de la carte (afficheurs 7seg, boutons poussoirs, leds et switchs).

L'enregistrement se fera sur un fichier unique, et ce fichier sera relu pour le playback. Il n'est pas demandé de pouvoir travailler avec d'autres fichiers.

Architecture générale

L'architecture générale du système est fixée et présentée sur la figure 1.

Les rôles des différentes tâches sont les suivants :

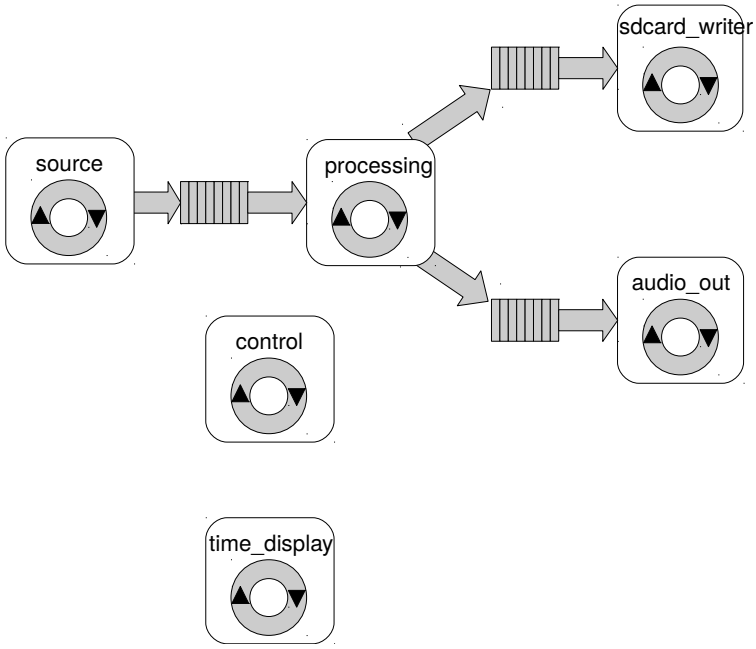

FIGURE 1 – Architecture générale du système

Tâche	Description
source	Récupération du flux audio d'entrée ou lecture du fichier précédemment enregistré. Cette tâche place ensuite les données dans un buffer lui permettant de communiquer avec la tâche processing.
processing	Cette tâche implémente la modification du volume. Elle pourrait être améliorée en offrant différents traitement sur le son (équaliseur, filtre, ...).
sdcard_writer	Cette tâche est responsable d'écrire dans le fichier les données reçues de la tâche processing.
audio_output	Cette tâche est responsable de jouer les données de son en les envoyant au driver gérant l'audio. La sortie audio est toujours active, que le système soit en enregistrement ou en lecture.
time_display	Gestion de l'affichage du temps.
control	Cette tâche contrôle l'ensemble du système, en scrutant les les boutons et en agissant en fonction.

Trois buffers (ou boites aux lettres) permettent l'échange des données sonores. A vous de les mettre en place. La synchronisation des différentes tâches est laissée à votre libre choix. Faites notamment attention à ne pas écrire et lire dans le fichier en même temps.

Contrôle

Le contrôle de l'enregistreur ce fera via les boutons présents sur la carte. Nous allons développer les fonctionnalités suivantes :

1. Record (key 0) : Lance l'enregistrement sur la carte, et stoppe l'enregistrement s'il est en cours.
2. Play/Pause (key 1) : Lance la lecture du fichier précédemment enregistré. Si la lecture est cours, celle-ci se met en pause, et si elle est en pause, elle doit reprendre.
3. Fwd (Forward) (key 2) : Avance la lecture de 10 secondes.
4. Rew (Rewind) (key 3) : Recule de 10 secondes dans le fichier.

Si un enregistrement est relancé, il doit effacer le contenu du fichier et recommencer un nouvel enregistrement.

Le volume est sélectionné par les switch présents sur la carte. Celui-ci sera appliqué dans la partie *processing*, et sera donc appliqué tant sur la sortie audio que sur les données à enregistrer sur la carte.

Pour accéder aux boutons il est nécessaire d'aller vérifier l'état des boutons de manière active (pas d'interruptions générées). Une tâche périodique devrait faire l'affaire pour aller vérifier si les boutons sont pressés ou non ainsi que l'état des switchs. Les fonction `get_buttons()` et `get_switches()` vous permet alors de récupérer ces informations.

Volume

Le volume sera compris entre 0 et 10. Les switchs correspondants doivent permettre de choisir le niveau sonore voulu. Les 10 leds de la carte vous permettent de représenter le volume actuel. La fonction `set_volume_leds()` permet d'appliquer la valeur de chacune des Leds à disposition.

L'intensité du signal audio est codé sur 16 bits. A vous d'appliquer la bonne opération en fonction du volume choisi. Considérez que le volume maximal correspond à jouer le fichier tel quel.

Avance/recule rapide

Cette fonctionnalité nécessite de synchroniser correctement la tâche lisant les capteurs et les autres. A vous de réfléchir au meilleur moyen de gérer cette synchronisation.

Play/pause/record

Cette fonctionnalité nécessite de synchroniser correctement la tâche lisant les capteurs et les autres. A vous de réfléchir au meilleur moyen de gérer cette synchronisation.

Affichage

Lors d'un enregistrement, l'affichage doit indiquer le temps écoulé depuis le début de l'enregistrement. Si une pause est effectuée, le temps ne doit pas défiler.

Lors de la lecture, l'affichage indique le temps courant par rapport au début de l'enregistrement.

Travail à effectuer

Le code qui vous est fourni ne contient qu'un squelette de programme, avec les différentes tâches déja en place. A vous de réfléchir à la manière de gérer le contrôle du système, à la synchronisation nécessaire, et de réaliser le tout. Il est clair que le code du labo précédent peut être repris, notamment pour l'envoi de l'audio et pour l'affichage du temps écoulé. Le code final devra être correctement commenté.

Un petit rapport présentant vos choix architecturaux vous est également demandé.

Le tout sera à rendre sur cyberlearn.