

Yocto et application multimédia

L'objectif de cet exercice est de mettre au point l'image d'un OS embarqué d'un dispositif qui joue un fichier audio avec traitements à chaque démarrage. L'environnement de compilation utilise Yocto pour la compilation de l'image et le développement de l'applicatif se fera dans un langage bas niveau (C/C++/Rust). Dans une deuxième partie de l'exercice, on cherchera à organiser le travail d'une équipe de 3 développeurs pour implémenter une amélioration du système.

Étape 0 : Mise en place de l'environnement de développement (dépôt git public sur Gitlab ou Github et structuration des sources à partir du repository: `git@github.com:jpiat/test-code-embedded-system.git`)

Étape 1 : Développer une application en C/C++/Rust qui lit un fichier mp3 dont le chemin est passé en argument. L'application applique un fade-in linéaire sur les 5 premières secondes de la lecture et un fade-out linéaire sur les 5 dernières secondes

Étape 2 : Écrire une recette Yocto dans l'environnement pré-configuré pour ajouter l'applicatif développé dans l'étape 1 et l'activer à chaque démarrage du système via systemd

Étape 3 : Configurer l'environnement yocto pour ajouter une image qui instancie la recette décrite dans l'étape 2 (tester en dry-run)

Étape 4 : Le product owner (PO) a besoins de connaître le calendrier de développement associé à une nouvelle fonctionnalité. La fonctionnalité est décrite par le PO avec le cahier des charges suivant :

- Possibilité de spécifier le fichier à jouer au démarrage via une interface web avec upload dans l'interface.
- Possibilité de spécifier le temps de fade dans l'interface.

Vous devez structurer le développement de ces fonctionnalités avec une équipe de 3 personnes :

- Un développeur frontend
- Un développeur backend
- Nn développeur embarqué

Décrivez l'organisation de l'équipe et les outils utilisés pour la gestion de projet.