Formation Zephyr RTOS : applications et drivers

Du firmware temps réel écrit avec Zephyr comme le projet l'entend : devicetree, Kconfig, driver model et code applicatif portable.

5 jours Durée	**Débutant à intermédiaire** Niveau	**Sur site ou à distance** Format	**Français ou anglais** Langue (supports en anglais)

Présentation

Cette formation apprend à écrire des applications temps réel et des drivers avec Zephyr RTOS. Elle commence par l'espace de travail et le système de build, explique comment le devicetree et le driver model décrivent et accèdent au matériel, passe par les threads, les interruptions et la mémoire, et se termine par les API des périphériques et le framework des capteurs.

Objectifs

À la fin de la formation, les participants savent :

- mettre en place un espace de travail Zephyr et maîtriser le système de build, Kconfig et les overlays ;
- décrire le matériel avec le devicetree et écrire des drivers pour le driver model de Zephyr ;
- utiliser correctement les threads, les interruptions, la synchronisation et l'allocation mémoire ;
- structurer une application avec zbus, des événements et des machines à états ;
- utiliser I2C, SPI, UART, ADC, PWM, le framework des capteurs et le stockage persistant ;
- déboguer avec les logs, le shell, les core dumps et GDB.

Public

Développeurs firmware qui débutent avec Zephyr, y compris ceux qui viennent du bare-metal ou d'un autre RTOS.

Prérequis

Bonne connaissance du C et notions de microcontrôleurs.

Programme

Le programme va des outils vers le noyau, puis vers le matériel.

Jour 1

- Zephyr, west et les espaces de travail, les toolchains
- Le système de build : configuration et compilation, Kconfig, cibles de carte et overlays, sysbuild, modules
- Les macros utilitaires, listes, ring buffers et opérations atomiques

Jour 2

- Décrire le matériel avec le devicetree : bindings, macros générées, pin control
- Le driver model : niveaux d'initialisation, API des périphériques, un driver hors de l'arbre

Jour 3

- Threads, ordonnancement et interruptions
- La gestion mémoire : allocation statique, slabs et heaps
- Structurer une application avec zbus, des événements et des machines à états

Jour 4

- Logs, shell et débogage, core dumps et analyse des piles
- I2C, SPI et UART

Jour 5

- ADC, PWM, watchdog et autres périphériques
- Le framework des capteurs
- Le stockage persistant : flash, NVS, ZMS, settings et systèmes de fichiers

Travaux pratiques et matériel

Les travaux pratiques se font sur des cartes STM32 Nucleo et Nordic nRF52840 DK. Les sessions ont lieu sur site, en France, en Belgique, en Suisse, en Allemagne et au-delà, avec le matériel fourni par LABCSMART, ou à distance. À distance, chaque participant dispose de son propre compte SSH et de son espace de travail sur notre serveur de build, et réalise les travaux pratiques sur des machines QEMU et sur de vraies cartes. Les exercices sont tirés de nos projets clients. Les formations sont données en français ou en anglais ; les supports de cours, eux, sont en anglais.

Qui anime cette formation

La formation est conçue et animée par le fondateur de LABCSMART, ingénieur Linux embarqué et noyau Linux avec plus de onze ans d'expérience industrielle. Il contribue au noyau Linux en amont pour les SoC Renesas RZ, dans les sous-systèmes PCIe, audio, Ethernet, thermique, DMA et horloges, et a écrit deux livres publiés chez Packt, *Linux Device Driver Development, Second Edition* (2022) et *Mastering Linux Device Driver Development* (2021), présentés sur notre page de publications.

Organiser une session pour votre équipe. Chaque formation est composée de modules indépendants. Une équipe peut retirer les modules dont elle n'a pas besoin et en ajouter d'autres à la place, et nous construisons aussi des formations sur mesure autour d'un SoC, d'une carte ou d'un projet. Contact : contact@labcsmart.com, https://labcsmart.com/fr/contact/. Page de la formation : https://labcsmart.com/fr/formations/formation-zephyr-rtos/