Formation STM32 bare-metal avec libopencm3

Des microcontrôleurs Cortex-M programmés sans HAL constructeur ni IDE, jusqu'à ce qui s'exécute avant main().

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

Présentation

Cette formation apprend la programmation bare-metal des microcontrôleurs Arm Cortex-M avec une toolchain open source et la bibliothèque libopencm3, sans HAL constructeur ni IDE. Elle va des horloges et des GPIO jusqu'aux périphériques de communication, puis descend aux scripts de linker, à la gestion de la flash, à l'architecture Cortex-M et à un bootloader maison.

Objectifs

À la fin de la formation, les participants savent :

- configurer les horloges, les GPIO, les interruptions et les timers avec libopencm3 ;
- piloter UART, I2C, SPI et CAN, avec interruptions et DMA ;
- écrire des scripts de linker et placer le code et les données en mémoire ;
- programmer la flash et la protéger avec les option bytes ;
- écrire un bootloader pour un microcontrôleur Cortex-M ;
- réduire la consommation avec les modes basse consommation et la RTC.

Public

Développeurs firmware qui veulent maîtriser entièrement le code de leur microcontrôleur, et ingénieurs qui arrivent sur Cortex-M depuis d'autres architectures.

Prérequis

Bonne connaissance du C. Aucune expérience préalable sur Cortex-M n'est nécessaire.

Programme

La formation commence par les périphériques et se termine par ce qui se passe avant main().

Jour 1

- Les microcontrôleurs, libopencm3 et la famille STM32
- Le contrôle des resets et des horloges
- Les GPIO, le pin control et le multiplexage
- Les interruptions et le NVIC, le timer SysTick

Jour 2

- Les timers et leurs modes de fonctionnement
- Le DMA périphérique
- Les U(S)ART avec DMA

Jour 3

- I2C et SPI, en polling, sous interruption et en DMA
- Le bus CAN

Jour 4

- L'organisation mémoire et les scripts de linker
- La gestion de la flash et les option bytes
- L'architecture Cortex-M et la gestion des exceptions, l'assembleur
- Écrire un bootloader
- La gestion de l'énergie et la RTC

Travaux pratiques et matériel

Les travaux pratiques se font sur des cartes STM32 Nucleo. 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-stm32-bare-metal/