

## Objectif pédagogique :

Accélérer la mise en place d'un bootloader sur microcontrôleur du type STM32 à cœur ARM Cortex-M.

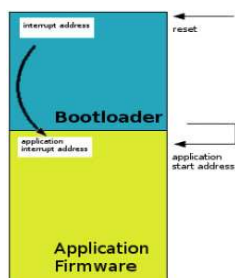
## Prérequis :

La pratique du langage C et de sa mise en œuvre dans des applications à base de microcontrôleur est indispensable. De bonnes connaissances sur le STM32 sont requises (formation réf. STM32-2J).

## Méthode :

Manipulation sur PC avec des outils de développement (IDE, Compilateur GCC, Débogueur), une sonde JTAG/SWD sur carte d'évaluation à base de STM32 à cœur ARM Cortex-M4.

**Durée :** 1 jour (1 x 8 = 8 heures)



## 1) Bases théoriques sur les Bootloaders

Durée : 2h

Comprendre la nécessité et la complexité d'un Bootloader entre le BSP de votre carte et votre programme applicatif.

- Les besoins d'un Boot pour mise à jour des logiciels applicatifs
- Discussion sur les médias présents sur votre carte (USART, SD, USB, TCP...)
- Complexité du Boot
- Sécurité du Boot
- Sécurité et contrôle des échanges
- Gestion de la mémoire
- Gestion des tables de vecteurs d'interruption
- Conditions d'écriture en mémoire Flash
- Description, contrôle et utilisation d'un fichier HEX

## 2) Travaux pratiques et mise en œuvre

Durée : 6h

- Ecriture dans la Flash du microcontrôleur
- Configuration et modification du fichier de script du linker
- Ecrire un Boot et lancer une application simple (configuration du Scatter File, fichier LD)
- Contrôle d'un programme reçu à partir d'un UART, puis écriture en flash
- Debug d'un programme d'application avec l'IDE
- Appel et utilisation d'une API du Boot à partir de l'application :
  - ✓ Modification des fichiers de script linker
  - ✓ Gestion de zone mémoire commune entre un Boot et une Application
- Utilisation d'un Boot UPG avec interpréteur et Dump mémoire
- Discussions & Echanges :
  - ✓ Discussion sur un Boot chargeant une application à partir d'Ethernet
  - ✓ Discussion sur le Boot interne d'un microcontrôleur
  - ✓ Comment faire une mise à jour d'un système composé de plusieurs cartes par USART, Bus CAN...
- Bootloader intégré au STM32 : Le *System Memory* interne du microcontrôleur est un pseudo Bootloader. Comment communiquer avec le protocole du *System Memory* Boot0=1, Boot1=1, et comment l'utiliser dans un système complexe multiprocesseur pour faire des mises à jour des programmes ?
  - ✓ Mise en place de fonctions de commandes pour se familiariser avec le protocole
  - ✓ Utiliser le *System Memory* à partir d'une application pour reprogrammer une zone de la Flash

### Références pour Formation BOOTLOADER :

- BL-1JP : formation inter-entreprises (1 jour)
- BL-1JS : formation intra-entreprise (1 jour)

### Références pour Formation STM32 + BOOTLOADER :

- STM32-BL-3JP : formation inter-entreprises (3 jours)
- STM32-BL-3JS : formation intra-entreprise (3 jours)

Numéro de déclaration d'activité de formation n° 11 75 53750

(Cet enregistrement ne vaut pas agrément de l'Etat, en application de l'article L6352-12 du code du travail)