

## Objectif pédagogique :

Acquérir une méthode pour bâtir un projet microcontrôleur ARM Cortex-M

## Prérequis :

De bonnes connaissances sur la programmation en langage C sont requises.

## 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-M7.

**Durée :** 1 jour (1 x 8 = 8 heures)



## 1) Notions de base

Durée : 4h

- Présentation de la famille Cortex-M
- Présentation de la série STM32
- Configuration d'un MCU ARM Cortex-M/STM32 :
  - Pins de Boot
  - Pins de Débogue
- Introduction sur l'architecture MCU Cortex-M :
  - STM32FXXX Memory Map
  - Les Registres
  - Table des interruptions
  - Error HardFault
- Présentation des périphériques d'une carte d'évaluation :
  - SysTick
  - GPIO

### Références pour Formation Création Projet MCU:

- PRJ-MCU-1JD : formation en ligne / distanciel (1 jour)
- PRJ-MCU-1JP : formation inter-entreprises (1 jour)
- PRJ-MCU-1JS : formation intra-entreprise (1 jour)

## 2) Travaux pratiques

Durée : 4h

*Les travaux pratiques sont réalisés autour du concept d'empilement des "couches d'abstraction". Chaque exercice se bâtit sur l'exercice précédent pour introduire les notions de création de projet par étape, et par couche logicielle (ex : library HAL, library CMSIS, etc...)*

### Exercices

#### • TP1 – Mise en place du projet :

Avec cet exercice, nous allons partir de rien et poser une première brique pour le développement d'un projet. Nous verrons quels sont les fichiers nécessaires pour démarrer un projet et l'utilité des autres.

#### • TP2 – Mise en œuvre de périphériques :

Dans cet exercice, nous allons mettre en œuvre le périphérique SysTick du cœur ARM et un GPIO. Dans un microcontrôleur, plusieurs entreprises participent à sa création : ARM a conçu la partie centrale du cœur autour duquel ST propose ses périphériques (GPIO, ADC, TIMER, SPI...). Cette organisation implique des bibliothèques et des documentations différentes.

#### • TP3 – Mise en place de la library CMSIS HAL :

Dans cet exercice, nous allons mettre en place la library CMSIS (Cortex Microcontroller Software Interface Standard) HAL de STMicroelectronics. C'est une couche d'abstraction s'appuyant sur la couche précédente et permettant la configuration des périphériques de manière plus aisée.

#### • TP4 – Mise en place du BSP de la carte :

Maintenant que nous avons incorporé dans le projet les APIs de la library HAL, nous allons pouvoir ajouter une couche supplémentaire avec les fonctions du BSP que nous propose STMicroelectronics pour notre carte d'évaluation STM32F769I-Discovery. Ces APIs apportent une facilité de programmation pour des tâches telle que la gestion d'un écran tactile.

#### • TP5 – Travailler avec le configurateur ST STM32CubeMX :

Nous allons voir avec cet exercice l'interfaçage de ce configurateur avec un IDE, un compilateur et un debugger pour votre projet. Différentes méthodes sont possibles : nous allons en expérimenter une, puis nous donnerons les pistes pour d'autres méthodes.

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)