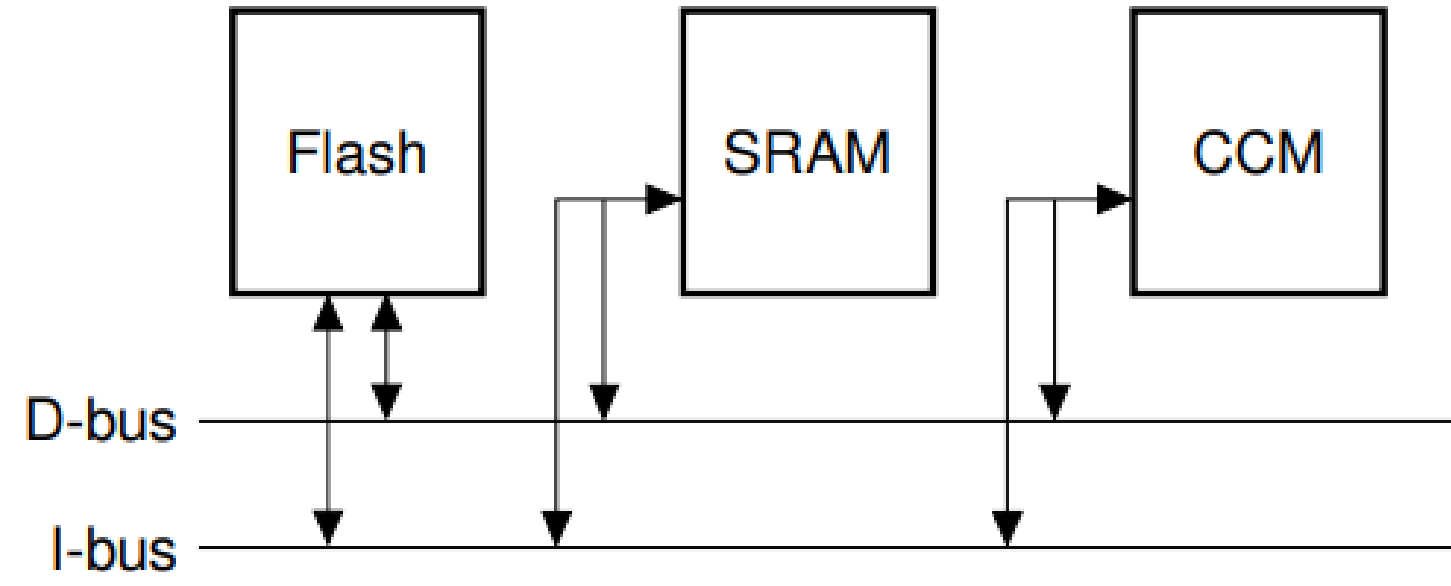




## Trouver une méthode d'allocation mémoire afin de minimiser la consommation électrique d'un système temps réel.

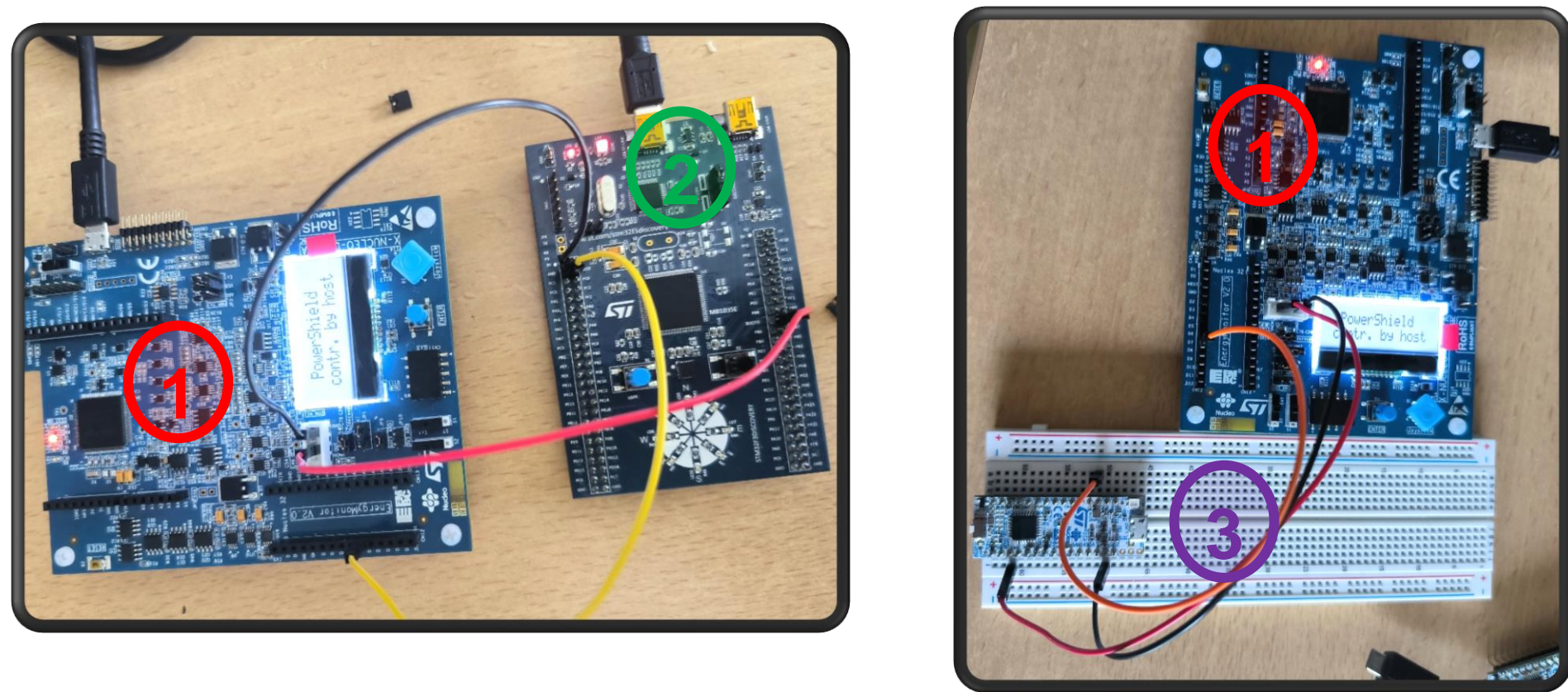
**Contexte :** J'ai effectué mon stage au sein de l'équipe VERTICS au LAAS-CNRS. J'ai étudié des travaux sur l'impact de la mémoire CCM-SRAM sur le temps d'exécution de programmes. En m'appuyant sur ces travaux, j'ai déterminé l'impact énergétique de cette mémoire sur des microcontrôleurs avec et sans Dynamic Voltage Frequency Scaling (DVFS). Ces résultats seront ensuite appliqués dans le domaine du temps réel.

### 1 Le CCM-SRAM sur les microcontrôleurs



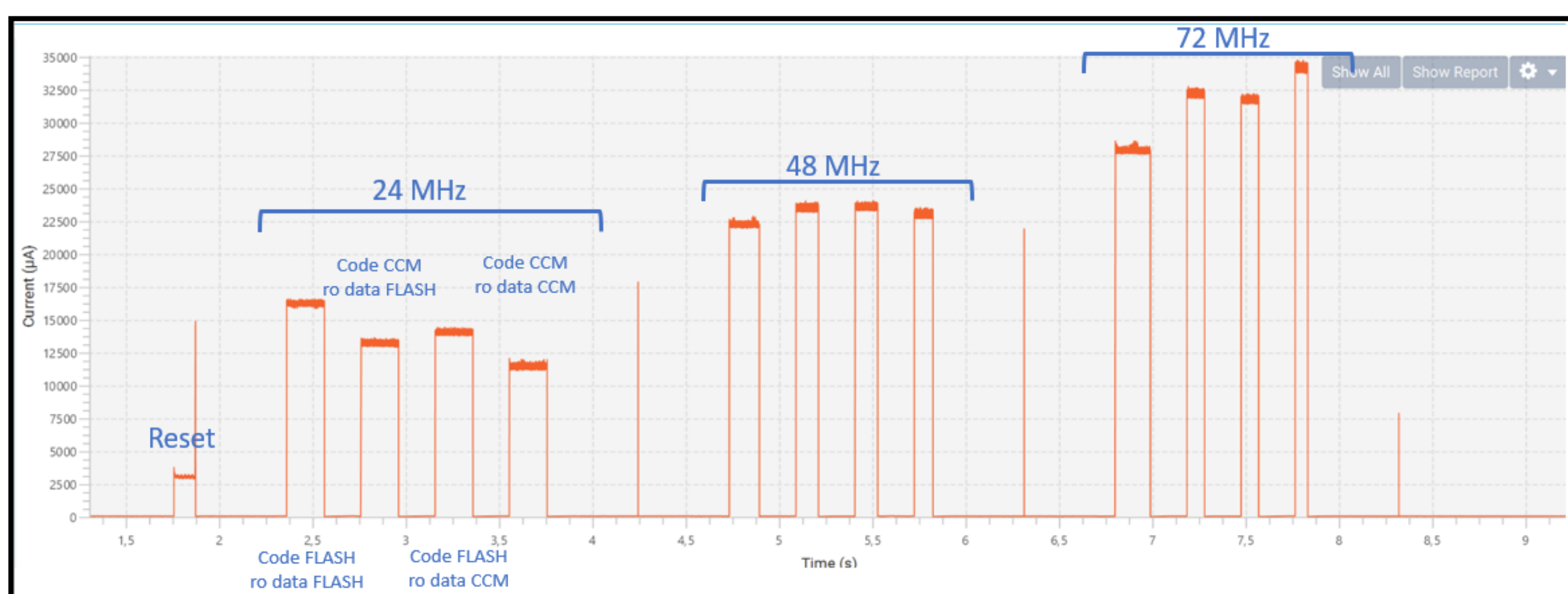
- Connecté aux bus de données et d'instructions.
- Accès mémoire **plus rapide** pouvant accélérer l'exécution d'un code.
- Espace de stockage **assez limité** (ex : 8 kB et 10 kB).

### 2 Mesures sur les microprocesseurs



1. La carte **NUCLEO-LPM01A** est connectée et alimente le microcontrôleur cible afin de **relever sa consommation**. Les informations recueillies sont transmises à l'ordinateur via USB.
2. Microprocesseur **STM32F303** (utilisé pour les mesures du poster).
3. Microprocesseur **STM32G431** ayant comme fonctionnalités en plus :
  - **Dynamic voltage frequency scaling** → modes de consommation.
  - **Cache d'instructions et de données** → accès mémoire plus rapide.
  - **Pre-Fetch** → Meilleur temps d'exécution, mais consomme plus.

### 3 Obtention des données



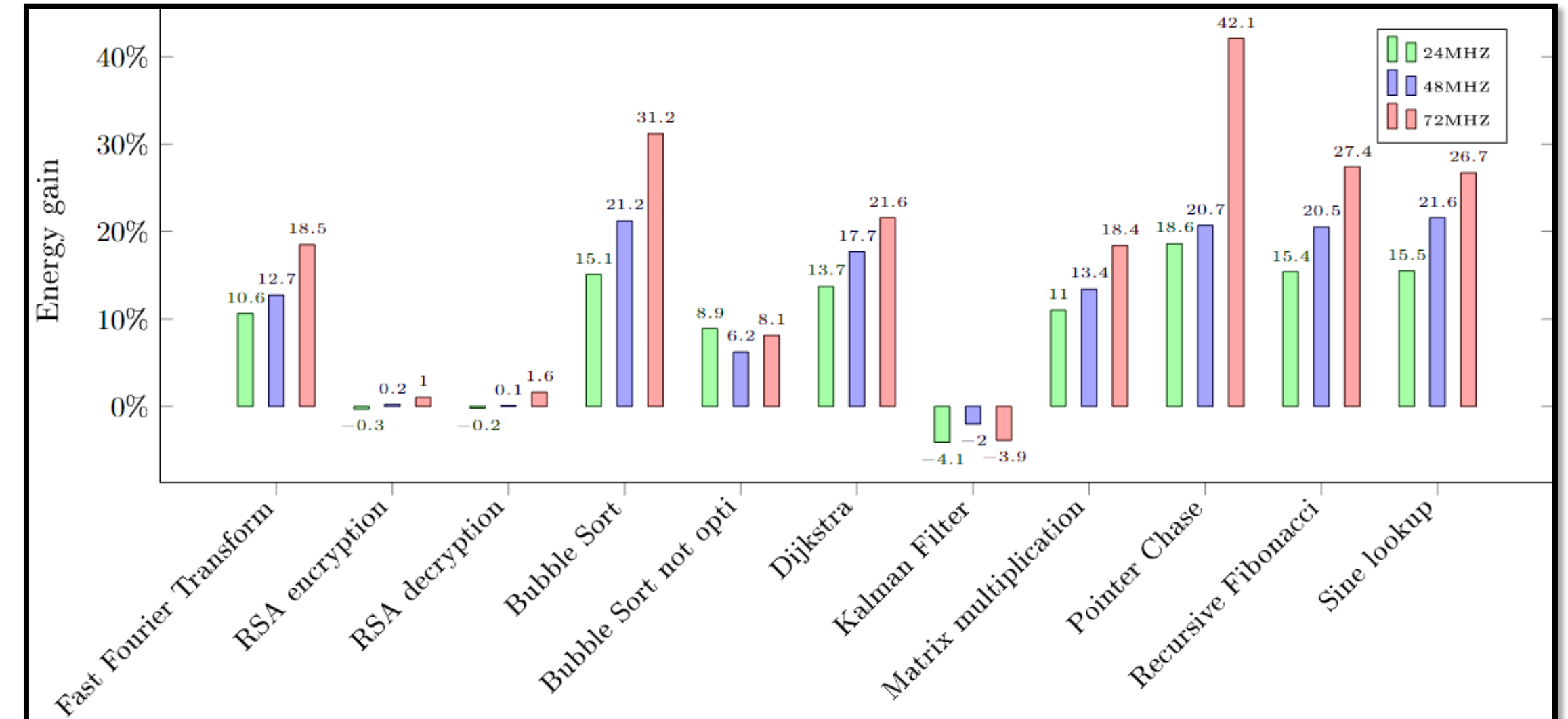
Résultat d'une mesure d'intensité sur l'exécution du pointer chase sur le STM32F303 (mesure transmise par le NUCLEO-LPM01A)

**10 benchmarks** ont été portés sur les deux microcontrôleurs afin d'obtenir assez de données. Nous avons mesuré **l'intensité** lors de l'exécution des benchmarks à **différentes fréquences** et avec toutes les **configurations mémoire** possible :

- **Instructions** dans le CCM, la SRAM ou la FLASH
- **Données d'entrées** dans la SRAM
- Données **en lecture seule** dans le CCM, la SRAM ou la FLASH
- Activation du **cache** et du **pre-fetch** (STM32G)
- Paramétrage du **DVFS** (STM32G)

Les mesures sont traitées grâce à un script Python qui crée des fichiers CSV en sortie.

### 4 Étude de l'impact du CCM sur l'énergie

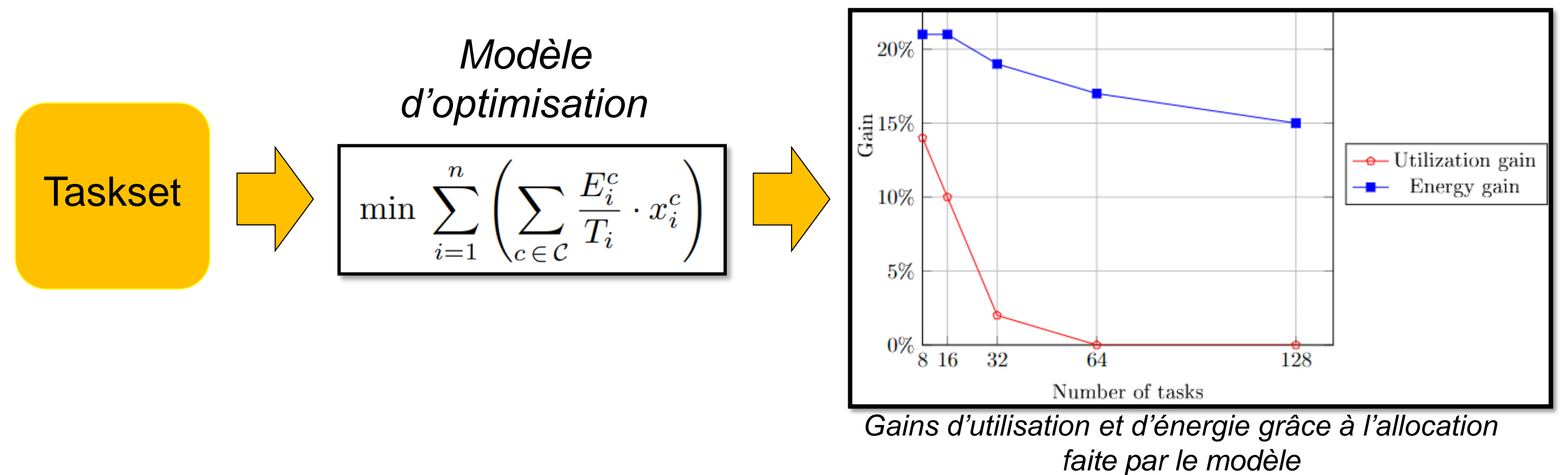


Amélioration de l'énergie en déplaçant les instructions de la FLASH vers le CCM

Le FLASH a une fréquence maximale plus faible que celle du processeur. Exécuter du code depuis le FLASH peut mettre le processeur en **état d'attente**, ce qui **consomme plus d'énergie**.

Le CCM va toujours aussi vite que le processeur. Il n'y a donc aucune attente. Le CCM sera **plus performant** que le FLASH à hautes fréquences.

### 5 Minimiser l'énergie d'un système temps réel



Nous avons les données de temps d'exécution et d'énergie de chacune des configurations mémoire et pour toutes les fréquences. Avec ces données, le modèle d'optimisation peut **trouver la meilleure configuration** pour chacune des tâches afin de **minimiser l'énergie**.  
→ **jusqu'à 21% d'énergie économisée.**

Le modèle respecte des **contraintes** :

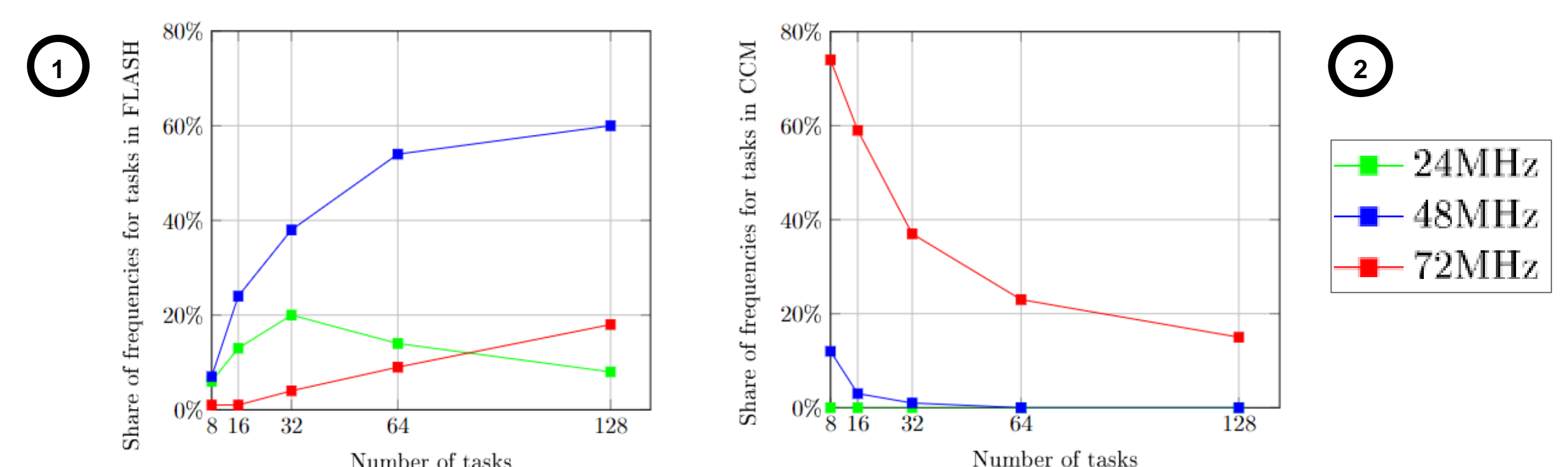
- **Temps réel** : les deadlines des tâches doivent être respectées.

$$\sum_{i=1}^n \sum_{c \in C} U_i^c \cdot x_i^c \leq 1$$

- **Mémoire** : les capacités de chacune des mémoires ne doivent pas être dépassées (ci-dessous, la ligne pour le CCM).

$$\sum_{i=1}^n \left( \sum_{c \in C} (x_i^c \cdot m_i^P) + \sum_{c \in C} (x_i^c \cdot m_i^{Ro}) \right) \leq M^C$$

Avec  $i$  l'indice de la tâche et  $c$  l'indice de la configuration utilisée :  $x_i^c = 1$  si la tâche  $i$  ( $\tau_i$ ) est à la configuration  $c$ .  $m_i^e$  l'espace utilisé par la section  $e$  (instructions, read only, input) de  $\tau_i$ .  $U_i^c$  le taux d'utilisation de  $\tau_i$  configuré en  $c$ .  $M^S$  l'espace de stockage de la mémoire  $S$  (CCM, FLASH, SRAM) et  $E_i^c$  l'énergie de  $\tau_i$  avec la configuration  $c$ .



Choix des fréquences par le modèle quand les instructions sont dans le FLASH (1) ou le CCM (2) (ex : avec 8 tâches, 75% des tâches sont dans le CCM et tournent à 72MHz)

### Conclusion / Compétences acquises :

- Meilleure compréhension du temps réel grâce à la lecture de plusieurs articles.
- Amélioration des compétences de programmation en C et ASM sur des microcontrôleurs.
- Introduction au monde de la recherche grâce à la rédaction d'un article scientifique.

**Loïc THOMAS - 4IR SI**  
Stage du 5 juin au 25 août  
LAAS-CNRS  
Tuteur : Tomasz KLODA