

# POA Projet C++ : CoffeePotController

*Informatique de Confiance : Cybersécurité du logiciel*

## Préambule

Dans le cadre du cours de « Programmation objet avancée » nous avons réalisé un projet en C++. Nous avons décidé d'implémenter un système pour allumer une cafetière à distance.

## Sujet

Le sujet est libre, il présente cependant un certain nombre de contraintes :

- Le projet doit être réalisé en C++
- Le projet doit contenir au moins 3 classes dont au moins une héritée (pas d'une bibliothèque)
- Un opérateur de classe doit être surdéfini
- Des exceptions doivent être gérés
- Le projet doit utiliser un type de la STL

## Présentation du projet

Nous avons décidé de réaliser un système de gestion de cafetière en réseau. Nous nous sommes inspiré du RFC 2324 (<https://www.ietf.org/rfc/rfc2324.txt>) qui définit le protocole HTCPCP.

Nous avons donc implémenté un serveur ainsi qu'un client qui s'échangent des « pseudo » requêtes HTTP interprétables par un navigateur (nous avons seulement testé Chrome et Firefox). Le client peut actionner la cafetière en ligne de commande.

Nous pouvons aussi accéder à une adresse définie sur un navigateur web et accéder à une page HTML qui nous permet de démarrer et éteindre une cafetière.

Un driver est implémenté contrôlant des servomoteurs qui allument et éteignent la cafetière. Les servomoteurs sont actionnés par un RaspberryPi, nous avons donc compilé le projet directement sur celui-ci (nous n'avons pas fait de cross-compiling).

Au niveau des contraintes qui étaient données, nous les avons toutes respectées, sauf la surdéfinition d'un opérateur de classe. Étant donné le nombre de bibliothèques que nous avons utilisé, cela nous a paru d'un intérêt très limité dans le cadre de notre projet.

## Problèmes rencontrés

Le principal problème a été de définir l'architecture de notre projet. Ayant seulement une idée vague du projet lors de son lancement, et peu de connaissance des outils disponible (bibliothèques), il nous a par exemple été difficile de définir un diagramme de classe et de le respecter.

Nous avons aussi dû utiliser un certain nombre de bibliothèques (boost, wiringPi, etc.) pour simplifier nos développements : le langage C étant maintenant ancien et disposant donc de nombreuses bibliothèques, il nous a paru pertinent de les utiliser et éviter de devoir ré-implémenter des classes qui existaient déjà.

Les derniers problèmes n'ont rien à voir avec la programmation mais sont d'ordre mécanique : nous avons dû faire un peu de bricolage avec les servomoteurs, certains ne fonctionnaient pas, ou juste à moitié et il nous a été difficile d'actionner l'interrupteur de notre cafetière avec ces derniers.