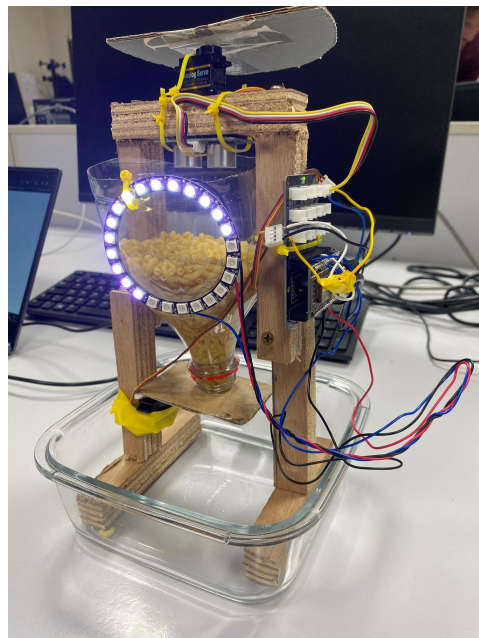


# Compte Rendu Projet C++

## Introduction

Dans le cadre de ce BE, nous avons décidé de réaliser un distributeur de croquettes automatique et programmable. Notre distributeur de croquettes connecté est conçu pour répondre aux besoins des propriétaires d'animaux de compagnie en simplifiant la distribution alimentaire et en offrant un contrôle personnalisé via une interface web. L'objectif principal est de permettre aux propriétaires de simplifier la distribution quotidienne des repas de leurs animaux domestiques en définissant des horaires spécifiques pour distribuer automatiquement les croquettes et cela même lors de leur absence. L'interface Web permet aux propriétaires d'ajuster les paramètres de distributions (heures programmables) et vérifier le niveau de croquettes restant.



## Conception

Pour démarrer la conception de notre distributeur de croquettes, nous avons tout d'abord fait une liste des fonctionnalités essentielles que notre distributeur doit avoir:

- Un réservoir pour faire une réserve de croquettes.
- Une trappe pour permettre l'ouverture du réservoir de croquette
- Un couvercle pour fermer le réservoir

Nous sommes donc partie dans l'idée de réaliser le distributeur le plus simple possible et ensuite nous avons ajouté des fonctionnalités:

- Une indication sur le niveau de croquette
- Système pilotable via une interface
- Programmation d'une dose journalière

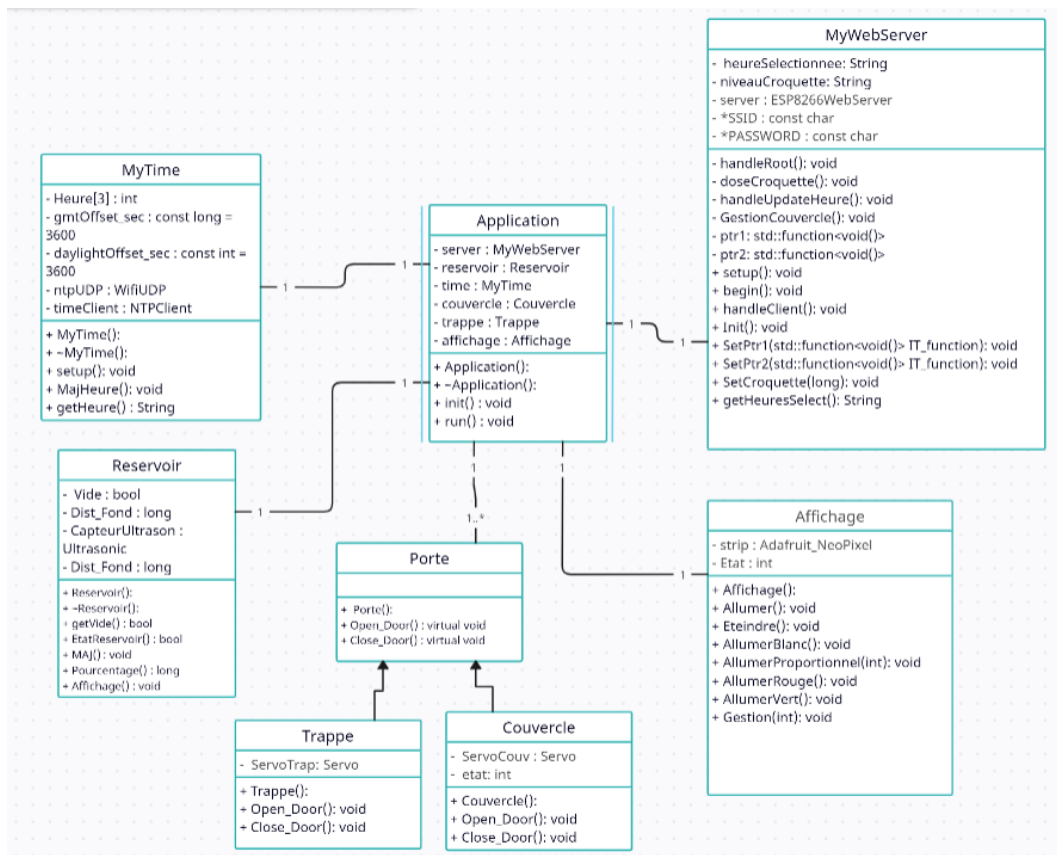
-Indication visuelle

Et d'autres encore que nous n'avons pas eu le temps d'implémenter mais que nous aborderons dans la liste des améliorations.

Au final voici le matériel en plus de la carte ESP8266:

- Capteurs Ultrason
- 2 Servomoteurs
- 1 Cercle de LED adressable

## Diagramme de classe du projet:



## Détail du fonctionnement du distributeur :

Lorsque le distributeur est mis en marche, l'on crée une instance de toutes les classes nécessaire : Trappe, Porte, Réservoir, Affichage, My Time, My Web Server et les initialise correctement.

Ensuite, le serveur se connecte au réseau wifi du domicile. Du point de vue de l'utilisateur, le cercle de led est bleu.

Lorsque la connection est établie, le cercle change de couleur pour indiquer le niveau de croquette du réservoir.(Rouge <20%, Vert >80%, un nombre de led correspondant entre). L'interface web est alors accessible.

Ensuite dans une boucle infinie, le distributeur va mettre à jour l'heure, le niveau du réservoir, l'affichage du cercle de led. Et déclencher une distribution automatique si l'heure correspond.

De plus, le système répond au demande envoyé depuis la page web pour effectuer les actions associées.

## Interface utilisateur distributeur

The screenshot shows a web interface for a distributor. At the top, there are two buttons: 'Dose croquette' and 'Couvercle'. Below them is the text 'Heure de distribution programmée'. Further down, there is a section for changing the distribution time, labeled 'Changer l'heure de distribution', which includes a time input field (showing '-- : --') and an 'Appliquer' button. At the bottom, it displays 'Niveau de croquette : 0'.

Depuis l'interface l'on peut déclencher une distribution de croquettes, ouvrir le couvercle et le refermer. Voir le niveau de croquettes en pourcentage et redéfinir une heure de distribution automatique.

## Organisation et problèmes:

Nous avons utilisé les 3 séances de BE pour avancer au maximum sur la structure de notre code, créer un git pour faciliter le partage des documents, le test des capteurs et actionneurs à part ainsi que la création des classes avec la déclaration des fichiers d'en tête etc. Nous nous sommes alors partagé les classes pour gagner du temps.

Les premiers problèmes sont survenus lorsque qu'il a fallu faire une première maquette de notre projet. Nous avons décidé de la réaliser en carton et morceaux de bois afin d'obtenir un prototype fonctionnelle rapidement. Mais la fixation de ces matériaux à nos composants n'a pas été aisée. En particulier le réglage des moteurs et surtout la fixation au servomoteur qui a été assez longue et fastidieuse.

Un autre point qui nous a retardé à était la réalisation de l'interface web. Ce fut la première que nous avons eu à créer. Nous l'avons donc d'abord fait dans un projet à part pour la développer. Une fois fonctionnelle nous avons voulu l'intégrer dans la classe "MyWebServer". Nous nous sommes alors confrontés à des problèmes pour réutiliser les mêmes fonctions étant donné que nous manipulons maintenant l'instance d'une classe. Après plusieurs heures de recherche, le problème a été résolu nous permettant de l'introduire dans notre projet final.

## Conclusion

Nous avons trouvé que le BE est très intéressant et nous a permis de travailler les notions liées à la programmation orientée objet. Mais aussi la gestion du travail de groupe notamment avec la plateforme github.

Cependant, le nombre de séances alloué au projet est trop faible. La réception des composants nécessaire est trop tardive ne nous permettant pas de prendre le temps de correctement s'impliquer dans le projet. Nous avons au moins dû passer le double d'heure normalement prévue au projet.

Le BE donne envie de s'impliquer et d'y passer du temps pour rendre un travail abouti, mais le temps ne le permet pas. Nous obligeant à présenter une maquette dont nous ne sommes pas totalement satisfait. Surtout que différentes fonctionnalités auraient pu être ajoutées comme la possibilité de mettre plus d'heure de distribution, ajouter d'autres actionneurs comme un haut-parleur pour enregistrer des sons à diffuser, et aussi perfectionner le code en gérant les cas critiques afin de rendre le distributeur vraiment abouti.