

Jardin connecté

Introduction :

Ce projet a pour objectif de concevoir un jardin connecté avec possibilité d'arroser les plantes en mode manuel ou automatique. Le système permet de mesurer la température et l'humidité ambiante, d'afficher ces informations sur un écran LCD et de gérer plusieurs plantes possédant leurs propres caractéristiques.

L'utilisateur peut interagir avec le système via un bouton poussoir qui va permettre de naviguer entre les différents écrans du LCD. Il peut aussi, via un capteur tactile, choisir le mode d'arrosage des plantes, ce mode sera représenté par une led.

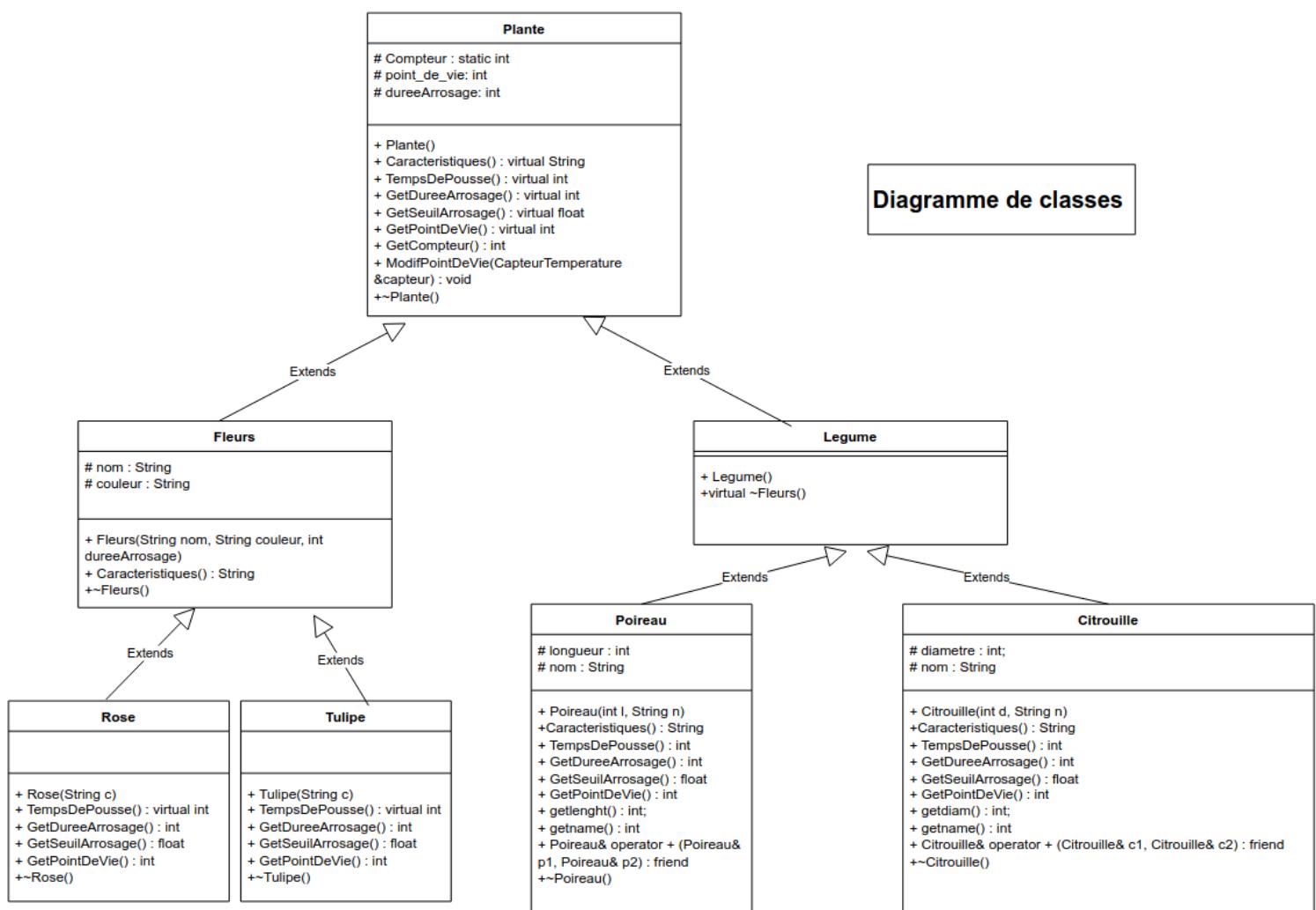
Ce projet s'inscrit dans le contexte actuel du développement des objets connectés et de la gestion des ressources.

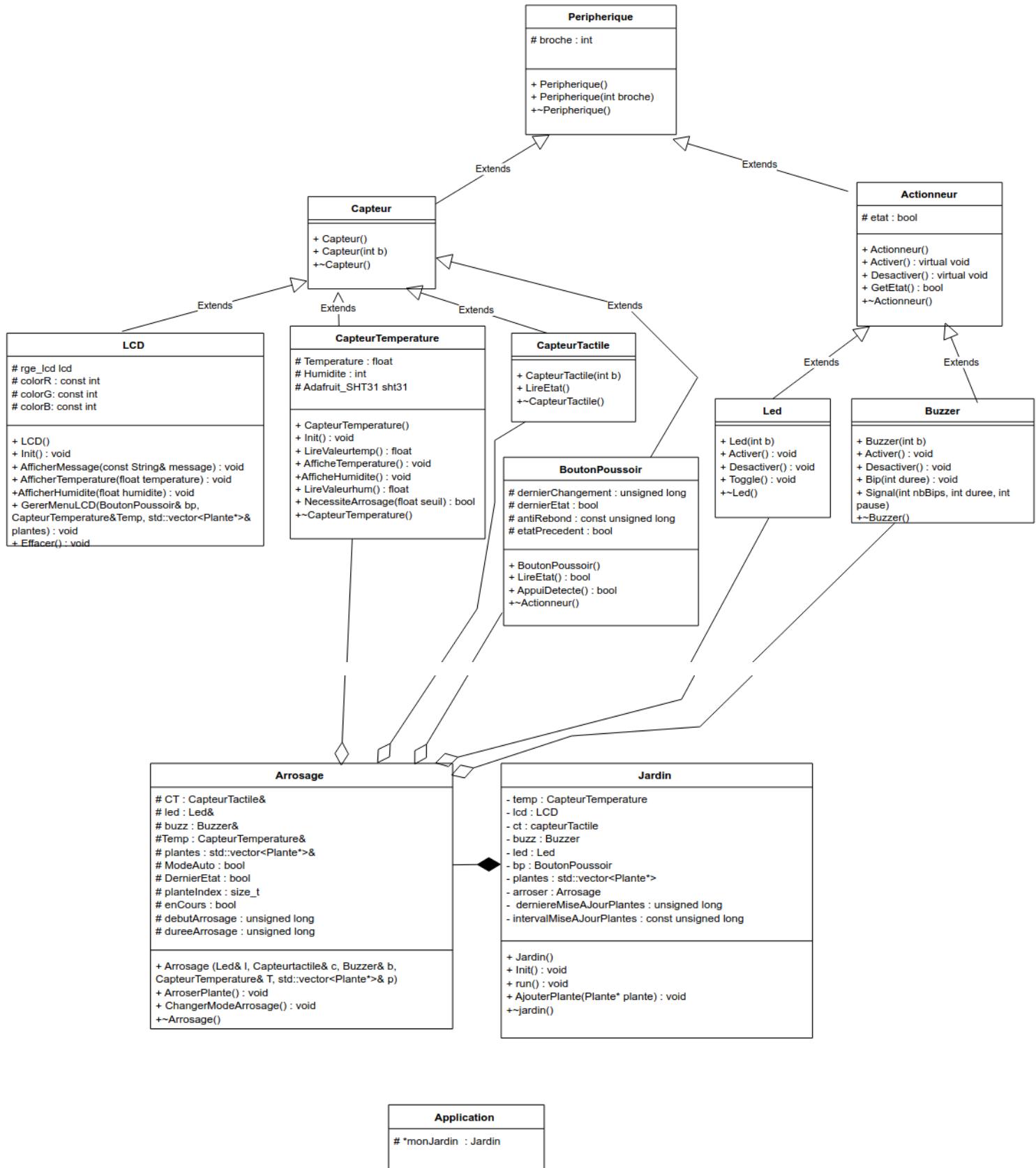
L'arrosage des plantes est une tâche répétitive qui dépend des conditions environnementales comme la température et l'humidité.

Automatiser cette tâche permet d'améliorer l'efficacité tout en limitant le gaspillage de l'eau.

Diagramme de classes :

Nous avons décidé de créer plusieurs classes avec deux arbres un pour les plantes et un pour les capteurs :





Fonctionnement du jardin connecté :

Création de plantes :

Premièrement pour avoir un jardin il faut créer des plantes dans Jardin.cpp. Il est possible de créer des roses, des tulipes (fleurs), des citrouilles et des poireaux (légumes). En utilisant l'opérateur +, le nouveau légume créé aura alors un taille étant la moyenne de ses parents et un nom composé à partir de celui de ses parents.

Utilisation du bouton poussoir :

Chaque pression sur le bouton poussoir va afficher sur le lcd la plante suivante du jardin, ses points de vie et ses caractéristiques. Les plantes sont ajoutées dynamiquement via un tableau grâce à la librairie <vector>. Un des éléments de la liste affichera la température et le taux d'humidité obtenus grâce aux capteurs correspondant.

Point de vie et humidité :

Un nombre de points de vie (modifiable dans la classe plante) et un seuil d'humidité critique (modifiable dans la fonction GetSeuilArrosage) sont attribués à chaque plante. Si le taux d'humidité capté par le capteur associé est inférieur au seuil d'une plante, celle-ci va perdre des points de vie de manière périodique. Une fois qu'une plante n'a plus de point de vie, elle est considérée morte et un message s'affiche sur le LCD.

Mode arrosage automatique

Pour passer en mode arrosage automatique, il suffit de presser le capteur tactile. La LED s'allume pendant que ce mode est actif. Comme son nom l'indique ce mode permettra d'arroser les plantes automatiquement si leur manque des points de vie. L'arrosage est représenté dans le monde réel par une activation du buzzer. Le buzzer émet un son plus ou moins long selon le type de plante qu'il est en train d'arroser. Il est possible de modifier le temps d'arrosage dans la fonction GetDureeArrosage. Pour sortir de ce mode, il faut appuyer une nouvelle fois sur le capteur tactile, la LED s'éteint alors.

Difficultés :

Nous avons rencontré un problème avec l'exception. Notre but était d'afficher un message sur le serial monitor si on essayait de créer une citrouille avec un trop gros diamètre. Nous observons que le programme n'affiche rien et fonctionne normalement même en créant une telle citrouille.

Nous avons également eu du mal à nous familiariser avec l'utilisation de git et cela nous a fait perdre beaucoup de temps lors des heures de cours.

Pour finir, nous avons adoré le fait de pouvoir choisir notre sujet mais avons été confrontés aux difficultés d'organisation, réflexion et conception du projet et nous avions tendance à vouloir faire beaucoup de choses au lieu de se concentrer sur l'objectif principal.

Améliorations possibles :

Il aurait été possible de créer une interface complète sur le lcd qui permettrait de créer de nouvelles plantes de manière classique ou par ajout de deux légumes. On aurait pu aussi imaginer pouvoir les renommer ou les arroser indépendamment. Ou encore un système de pousse et de récolte si on s'est bien occupé de ses plantes.

Conclusion :

Nous avons réalisé un jardin connecté fonctionnel en essayant de répondre à certaines problématiques de la vie quotidienne.

Durant ce projet, nous avons acquis des compétences en programmation orientée objet C++.

Nous nous sommes familiarisés avec l'utilisation de capteurs et actionneurs, de la communication I2C, de l'IDE arduino et de git.

Ce projet nous a aussi permis de comprendre les enjeux de l'électronique embarquée et des situations en temps réel.