LICINI Valentin
XU Andy

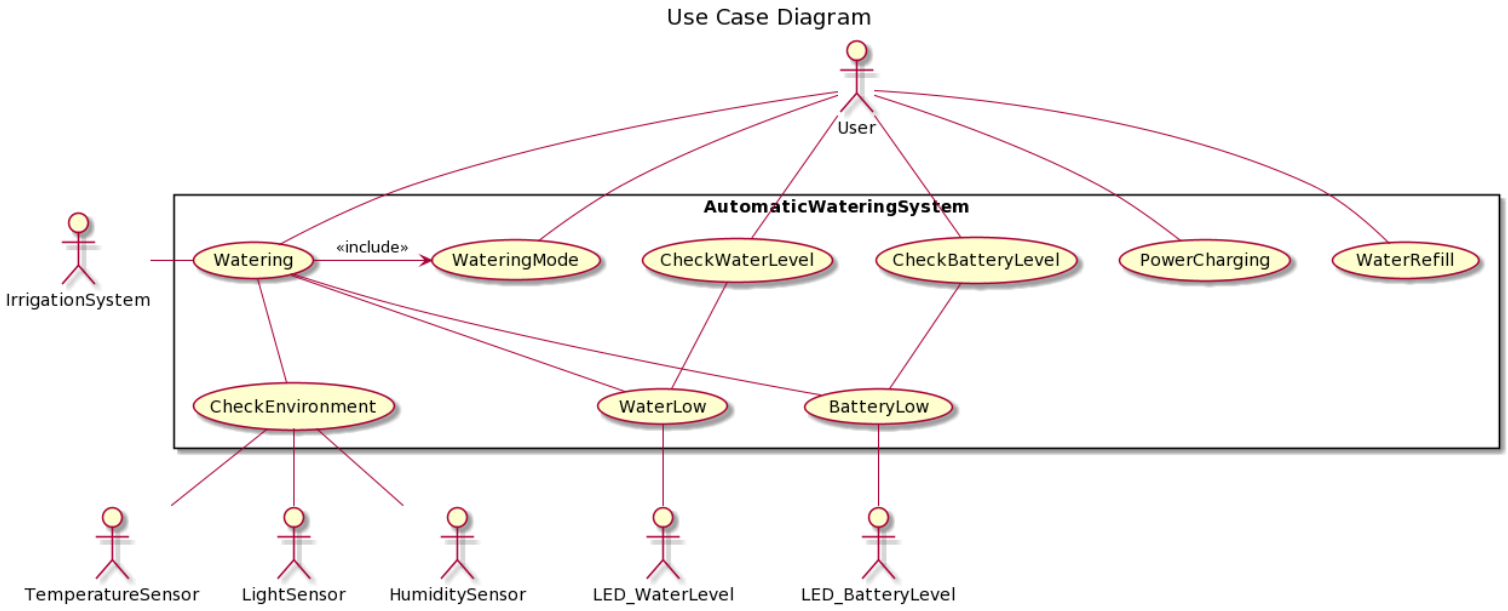
4 AE-SE 3

Rapport de BE COO en C++

Conception :

Nous avons choisi de concevoir un système permettant un arrosage automatique de plantes. Divers capteurs sont ainsi utilisés pour choisir à quel moment de la journée arroser la plante suivant ce que souhaiterait l'utilisateur.
Notre système est ici très simple par rapport à un objet qui pourrait être bien plus "smart".

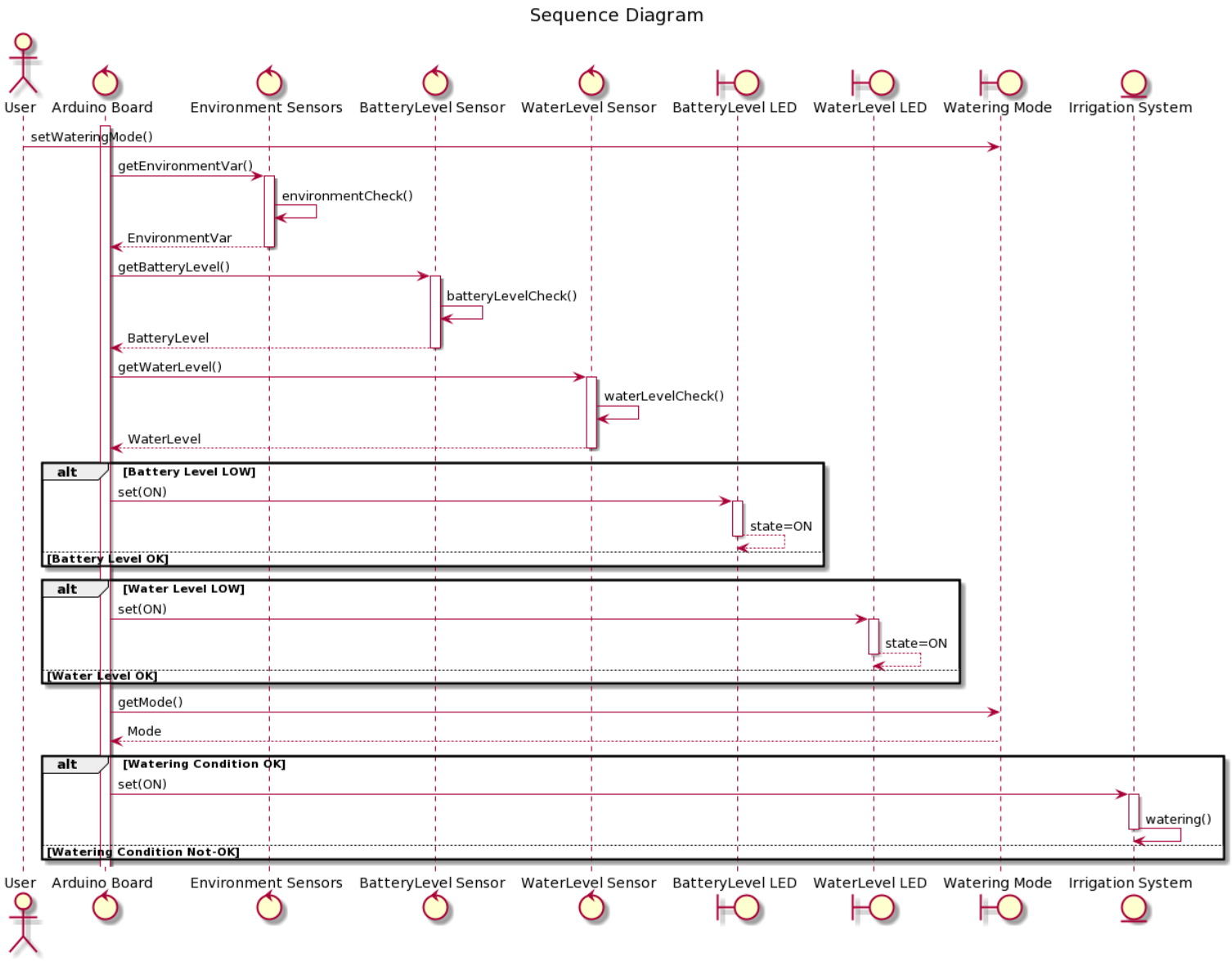
Diagramme d'utilisation :



L'utilisateur peut ainsi choisir différent mode de fonctionnement du système pour paramétrer l'arrosage automatique. Nous n'avons donoc que coder une version très simplifiée de ce que pourrait être le projet, notamment au niveau des algorithmes choisis pour l'automatisation de l'arrosage. L'utilisateur peut aussi activer manuellement l'arrosage et devra recharger la batterie ainsi que remettre de l'eau pour ce prototype. Un faible niveau dans l'une de ces deux ressources est indiqué par une led qui s'allume.

Diagramme de séquence :

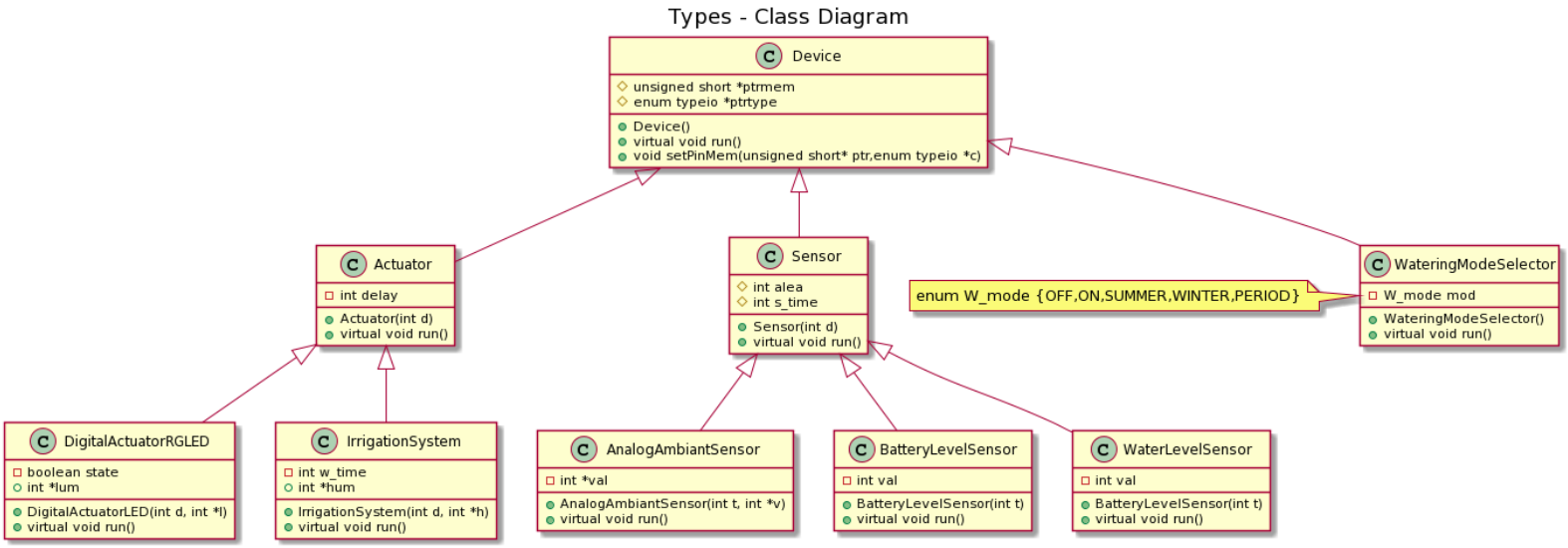
Le système que nous avons codé sur la carte arduino fonctionne comme suit



Les leds de niveau faible s'allument donc lorsque le système le détecte. Le système récupère les différentes données fournies par les capteurs environnementaux et suivant le mode sélectionné par l'utilisateur, il active l'arrosage automatique si les conditions sont remplies (niveau d'eau et de batterie suffisant outre les conditions météorologiques). L'utilisateur peut aussi choisir le mode arrosage manuel et le système activera alors l'arrosage.

Diagramme de classe :

Nous avons donc choisi de définir les différentes classes de périphériques comme ceci



Nous avons alors deux types de périphérique principaux, dérivés de la classe standard, les capteurs et les actionneurs.
Nous utilisons ici 3 capteurs environnementaux, qui étant utilisés et fonctionnant de la même façon, sont regroupés sous la même catégorie. Le capteur du niveau de batterie et celui du niveau de l'eau sont différenciés car ils détectent directement des valeurs internes au système.
Les actionneurs ont pour similarité un délai lors de l'activation, mais dans notre cas, ils sont très peu semblables finalement car l'un est une LED avec une forte vitesse de réaction, contrairement au système d'arrosage qui est bien plus lent dans la réalité que ce que nous avons codé.
Il reste ensuite l'interface avec l'utilisateur, qui permet de définir le mode d'arrosage du système (il pourrait s'apparenter à beaucoup de forme dans une conception concrète).

Fonctionnement du système :

Nous avons implémenté une simulation qui fonctionne bien plus rapidement qu'un même système réel. Beaucoup de valeurs, comme la consommation par le système sont très éloignés de la réalité, mais permettent de faire évoluer rapidement le système pour observer les différents comportements attendus.
Le système utilise des variables globales pour simuler l'environnement. Nous n'avons pas mis en place une modification en temps réel de ces derniers, il faut changer la situation de départ lors de l'exécution du programme dans *board.cpp*.
L'utilisateur intéragit avec le système par l'ajout de fichiers dans le dossier *src*.