MARQUES Raphaël
CHAMPAGNE Julie

Compte Rendu de BE de C++

La météo nous réveille

Introduction :

Le réveil forcé est l'un des moments les moins agréables d'une journée. Pourquoi ne rendrait-on donc pas ce réveil plus plaisant en lui ajoutant une utilité ? Un réveil qui donne la météo ! Le projet est alors de récupérer des données de météo grâce aux capteurs d'humidité et de température, d'infrasons et de lumière, ce qui permettrait ensuite de choisir une musique propice à la météo. Une musique joyeuse et énergique pour une journée ensoleillée ou une musique plus calme pour une journée pluvieuse. Ainsi plus besoin de se précipiter pour regarder la météo et choisir ses vêtements. Dès les premières secondes du réveil, nous savons déjà à quoi nous attendre ! Sans oublier qu'il peut être adapté à une alarme thermique ou un outil de surveillance des plantes !

Le Projet :

· Etat du projet :

Actuellement, le projet n'a pas atteint le stade final prévu. En effet, nous arrivons à récupérer les données de température, d'humidité et de luminosité ainsi que d'agir en fonction sur le buzzer pour que le son soit différent. L'alarme, cependant, n'est pas réglable, de façon mélodieuse. Elle se déclenche dans trois cas : si l'un des seuils de température, d'humidité ou de luminosité est dépassé. L'alarme s'arrête lorsqu'on touche le capteur toucher. En outre, la Led s'allume à partir de ces seuils et un message est écrit sur l'écran.

· Difficultés :

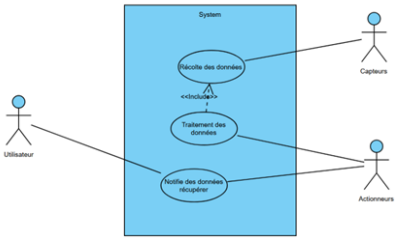
Nous avons rapidement supprimé le capteur infrason qui devait recueillir les informations pluviométriques car la précision du capteur étant faible, il aurait fallu ajouter un réceptacle qui récupère beaucoup d'eau dans une petite surface (utiliser une sorte d'entonnoir par exemple). Pour des raisons d'optimisation de temps, nous avons vite délaissé cette idée, d'autant plus que le capteur d'humidité peut nous donner des indices quant à la présence de pluie.L'utilisation des deques s'est avérée difficile afin d'éliminer les mesures trop séparées des mesures alentour car les itérateurs restaient peu facilement lisibles ou compris. Par ailleurs, nous avons eu du mal à jouer nos mélodies, nous avons donc décidé de n'utiliser qu'un bip qui sort à une fréquence différente selon la météo. Par manque de temps, le projet n'a pas pu implémenter la gestion du temps, que cela soit à l'affichage ou en tant qu'autre alarme. L'une des raisons est que la fonction init() de la classe UTCTime était bloquante, et la synchronisation avec NTCTime n'a pas pu être testée.

· Points à améliorer :

Un des points à améliorer serait donc de réussir à jouer nos mélodies sur le buzzer pour avoir une sonnerie plus agréable. Il faudrait aussi trouver un moyen de programmer l'alarme pour qu'elle se déclenche à une heure donnée. La capture de l'humidité n'est pas

très pertinente pour des réveils très matinaux ou en hiver, lorsqu'il fait encore nuit ou que la rosée matinale augmente considérablement le taux d'humidité. De la même manière, la capture de la luminosité n'est réellement viable que lorsque la personne se lève en même temps que le sommeil. La possibilité de régler les seuils de ces capteurs, par exemple avec un potentiomètre, permettrait de pallier ces problèmes.

Diagramme d'utilisation et de classe :



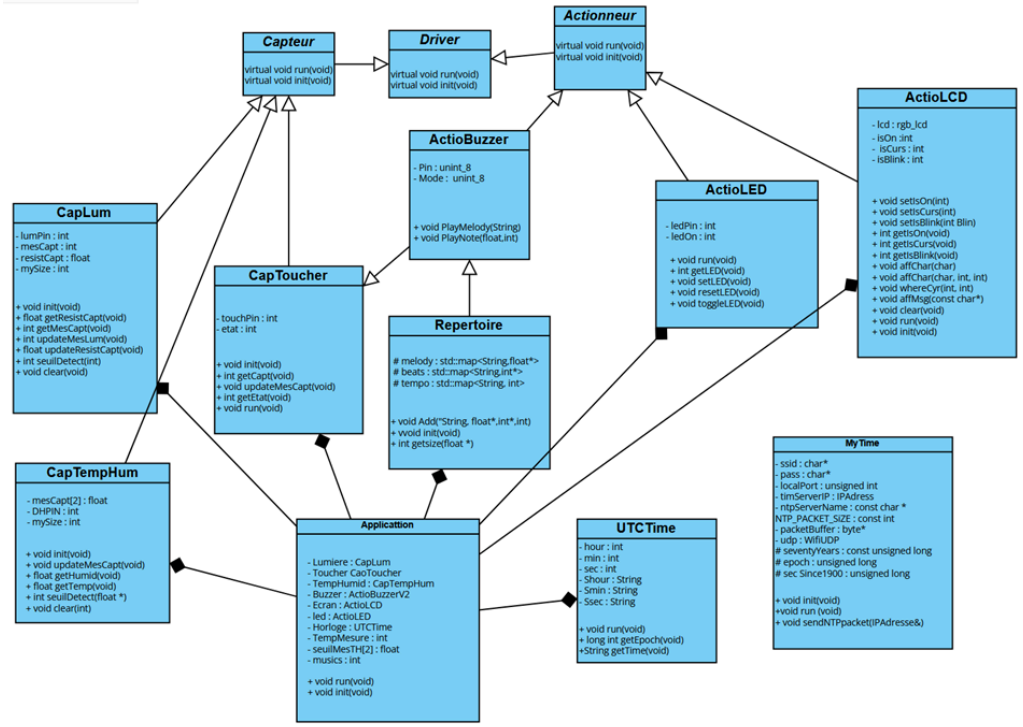
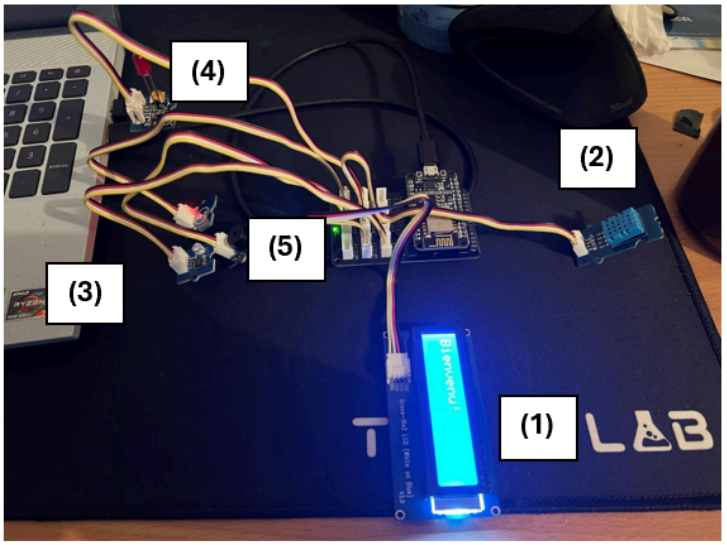


Schéma de fonctionnement :

- Matériel



- (1) Ecran LCD 16x2

(2) Capteur de température et d'humidité

(3) Capteur de lumière

(4) LED

(5) Buzzer

(6) Capteur tactile (centre de 3, 4 et 5)

MARQUES Raphaël
CHAMPAGNE Julie

. Logiciel