

Bureau d'étude: Application du C++ au domaine des objets connectés

Duc Minh Muon - Quang Phu Le

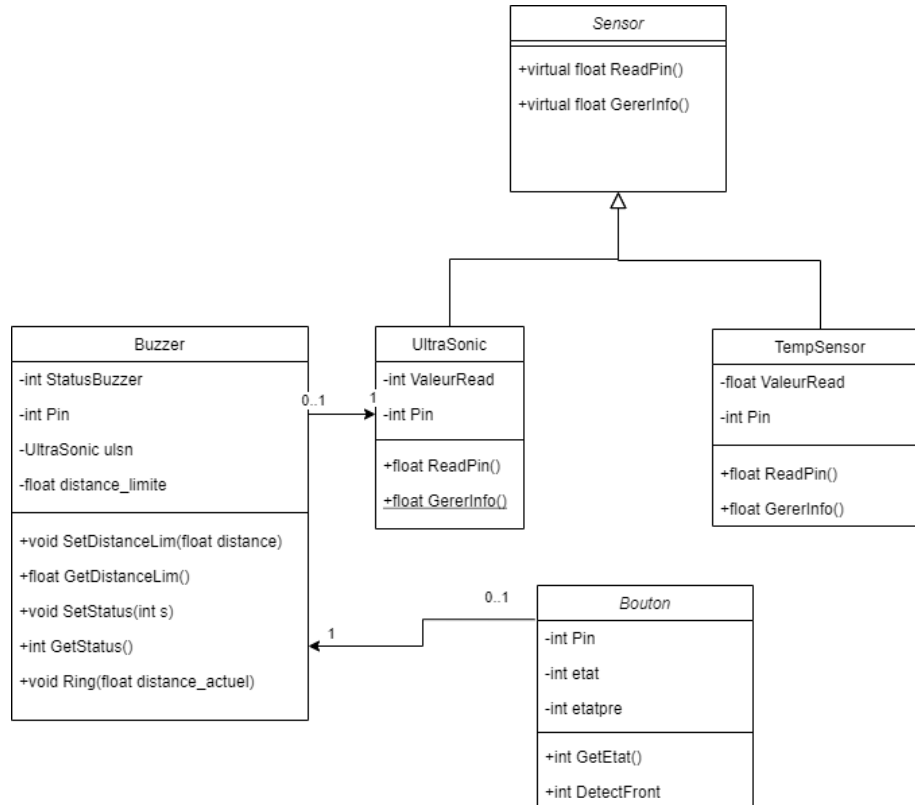
Mai 2021

1 Introduction

Pour ce BE, on a été fourni un microcontrôleur ESP8266 programmable sur Arduino, un écran Grove-LCD RGB Backlight, un capteur de distance Ultrasonic Ranger, un capteur de température, un buzzer et un bouton. L'objectif de ce bureau d'étude est de créer une bibliothèque modulaire, extensible pour s'interfacer avec des capteurs et des actionneurs et de développer une application pour utiliser notre bibliothèque.

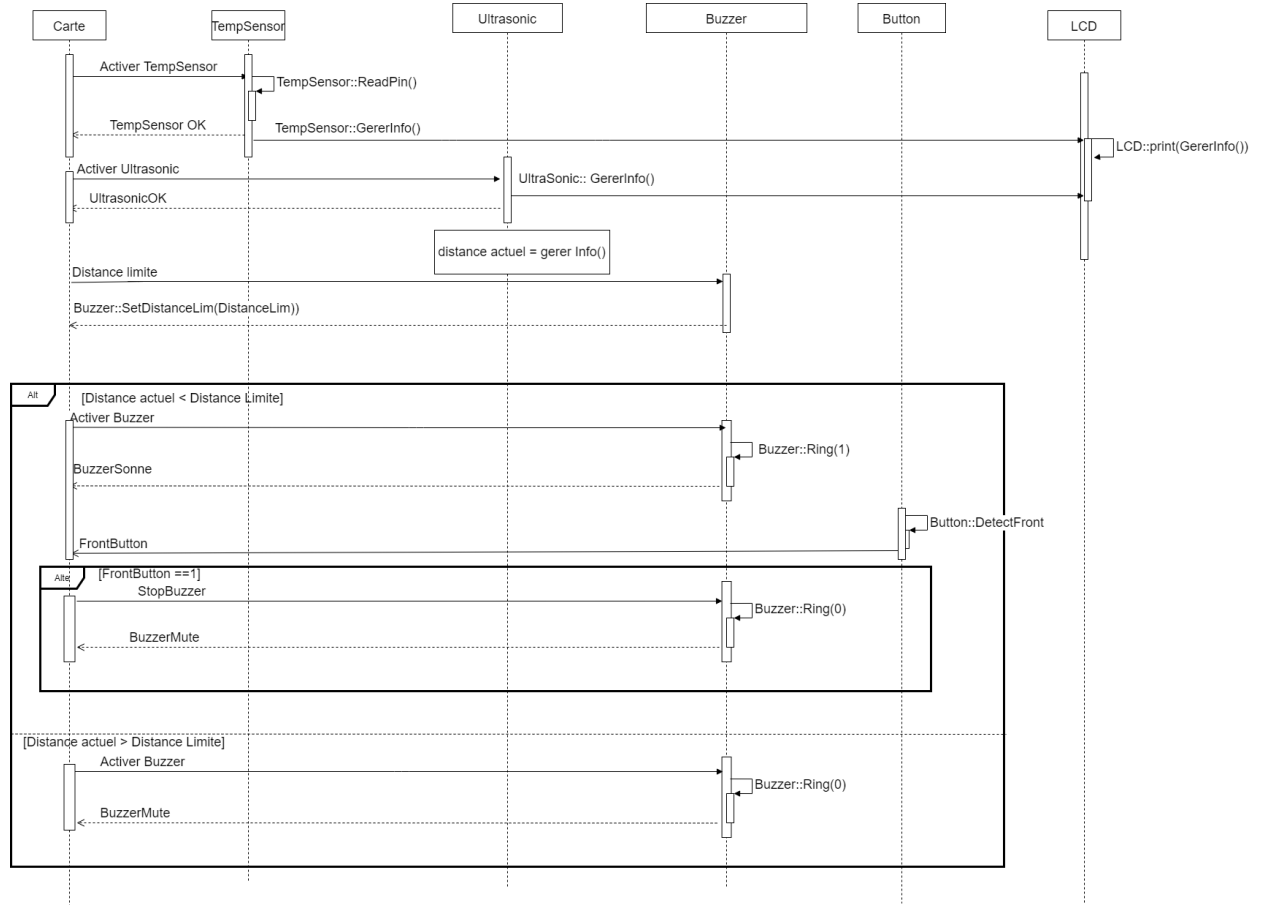
Sur l'écran on verra la température et la distance qui sont détectées par les capteurs fournis et il se rafraichit chaque 500ms pour afficher des nouvelles données. Au début, l'écran est en rouge et ça veut dire que le buzzer est activé. Donc si la distance est inférieure qu'une limite (on a défini 10cm par défaut et on peut le changer), le buzzer sera déclenché. On peut désactiver le buzzer en pressant le bouton. L'écran deviendra vert et on pourra même toucher le capteur de distance et le buzzer ne bippera pas

2 Diagramme de classe



Les deux capteurs partagent des fonctionnements similaires donc on a décidé de créer une classe mère qui contient ces fonctionnements. Pourtant, l'initialisation d'un objet de ces deux classes avec leur numéro de pin sont dans les constructeurs qu'on a créés et donc n'apparaissent pas dans le diagramme. On a aussi créé une classe pour le buzzer et une pour le bouton. La classe 'bouton' est en association avec la classe 'buzzer' qui est elle-même en association avec la classe du capteur de distance

3 Diagramme de séquence



4 Problèmes rencontrés et perspective d'amélioration

Comme c'est la première fois qu'on travaille avec une carte Arduino, on a rencontré des difficultés au début, notamment avec l'installation et la configuration de son IDE ou trouver un datasheet pour faire le lien entre le numéro de pin dans le programme et celui sur la vraie carte. On a aussi eu des difficultés quand on devait chercher des bibliothèques nécessaires et les implémenter dans notre projet. Notre projet subit aussi des problèmes de fonctionnalité quand on met un délai important après chaque itération de la boucle principale

Pour améliorer ce projet, on aurait pu intégrer des concepts de fonctionnement en temps réel comme mutex ou sémaphore et par conséquent, tous nos composants auraient pu marcher parallèlement.