

Rapport du projet bureau d'Etude en C++

I. Description du projet

Pour ce projet, nous avons choisi de réaliser un instrument de musique à l'aide des capteurs mis à notre disposition. Nous avons ainsi choisi de réaliser des instruments avec deux capteurs différents : un capteur ultrason et un potentiomètre. Pour le capteur ultrason, la note jouée dépend de la distance de notre main au capteur et pour le potentiomètre, c'est la variation de l'angle qui va permettre de changer de note. Les notes sont jouées à l'aide du buzzer. Un bouton va finalement permettre à l'utilisateur de sélectionner le mode souhaité : c'est-à-dire sélectionner un des capteurs pour jouer les notes.

La configuration de nos capteurs est définie comme présentée dans l'illustration ci-dessous :

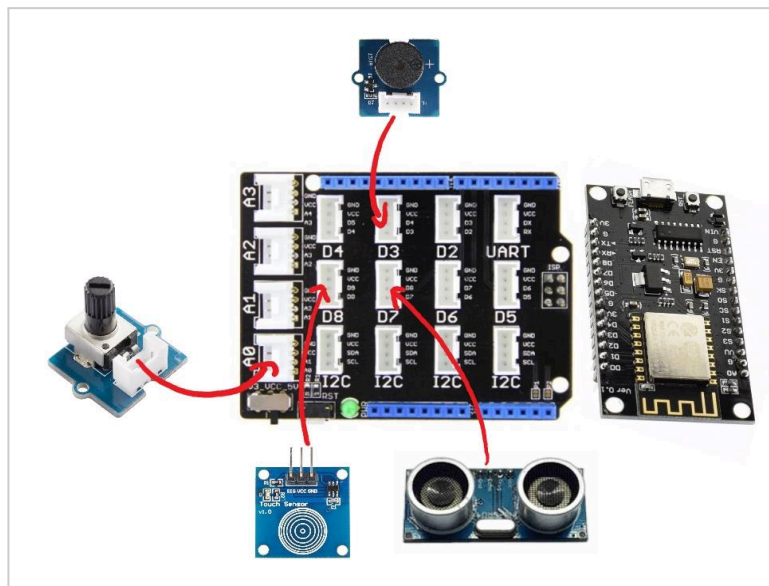


Schéma de branchement des capteurs

Fonctionnement du projet :

L'utilisateur peut sélectionner l'instrument qu'il souhaite en utilisant le bouton touch sensor. En appuyant sur le bouton, il choisit d'utiliser le potentiomètre, sinon le capteur ultrason est sélectionné par défaut.

Une fois l'instrument sélectionné, l'utilisateur peut jouer en direct une séquence de 10 notes avec le capteur. Les intervalles ont été définis pour permettre de jouer une octave de notes. Cette séquence de 10 notes est enregistrée en parallèle puis est rejouée par le buzzer.

À noter qu'il y a un délai de 2s entre chaque note pour permettre à l'utilisateur de changer la distance ou l'angle du capteur. Une fois la séquence rejouée, l'utilisateur peut à nouveau sélectionner un instrument et rejouer une séquence de notes.

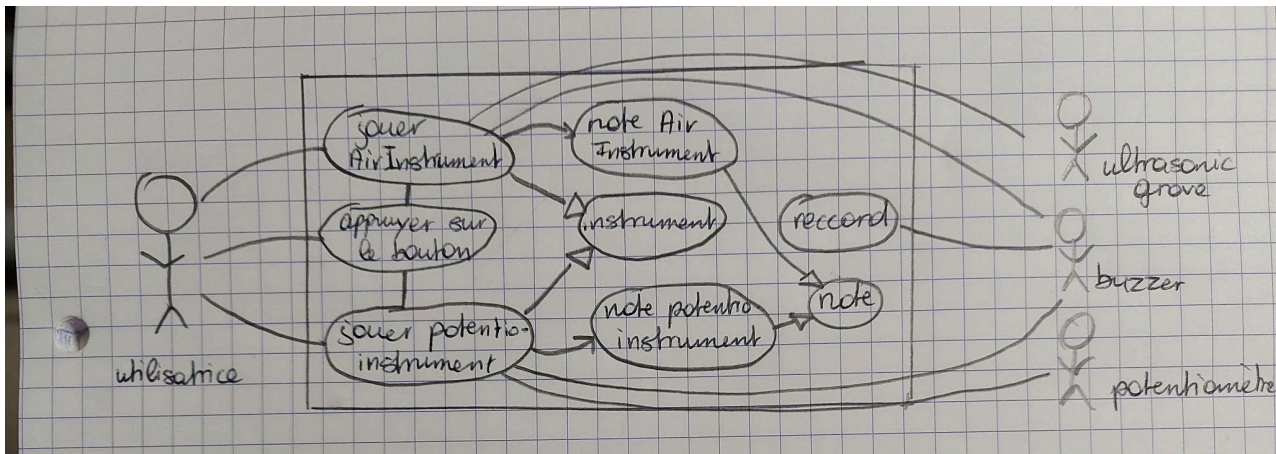


Diagramme de cas d'utilisation

II. Structuration du code

Afin de mener à bien ce projet, nous avons défini diverses classes permettant chacune de réaliser une fonction précise.

Ainsi, nous avons défini une classe **Instrument** puis une classe **PotentiInstrument** et **AirInstrument** héritant toutes les deux des attributs de la classe **Instrument**. Ces deux classes permettent de configurer les capteurs, de réaliser la mesure de distance ou d'angle et de la stocker dans une variable appelée *mesure*.

Nous avons ensuite réalisé une classe **Note** pour définir la gamme des notes utilisées et la fréquence à envoyer au buzzer pour chaque note. Nous avons ainsi défini 8 notes différentes, soit une octave. De la même manière qu'avec **Instrument**, deux classes **Note_air_instrument** et **Note_potentio_instrument**, héritant de **Note**, ont été implémentées pour permettre d'associer à chaque intervalle d'angle ou de distance une note et de la jouer dans le buzzer.

Une classe **Record** permet à l'aide d'une queue d'enregistrer la séquence de note puis de la rejouer.

Nous utilisons également des exceptions afin de détecter un potentiel problème au niveau de la mesure des capteurs. Ainsi, si la valeur retournée par le capteur n'est pas dans les intervalles définis dans nos classes, une exception sera lancée.

Point sur la classe Application :

Tous nos capteurs sont initialisés dans la méthode `init` de **Application**. La liaison série avec le terminal de l'interface arduino est également initialisée pour permettre de visualiser les notes jouées.

Dans la méthode `run` de **Application**, la valeur du bouton est lue afin de dissocier les cas d'utilisation : capteur ultrason ou potentiomètre. Dans chacun des cas, on crée un objet **PotentiInstrument** ou **AirInstrument**. On crée ensuite un objet **Record** qui prend en paramètre une référence de notre objet **PotentiInstrument** ou **AirInstrument**. On appelle ensuite la méthode `enregistrer()` de la classe **Record** qui va appeler la méthode `LierNoteMesure()`, qui, elle, va jouer la note et retourner la note. La méthode `jouer()` de la classe **Record** permet ensuite de rejouer la séquence.

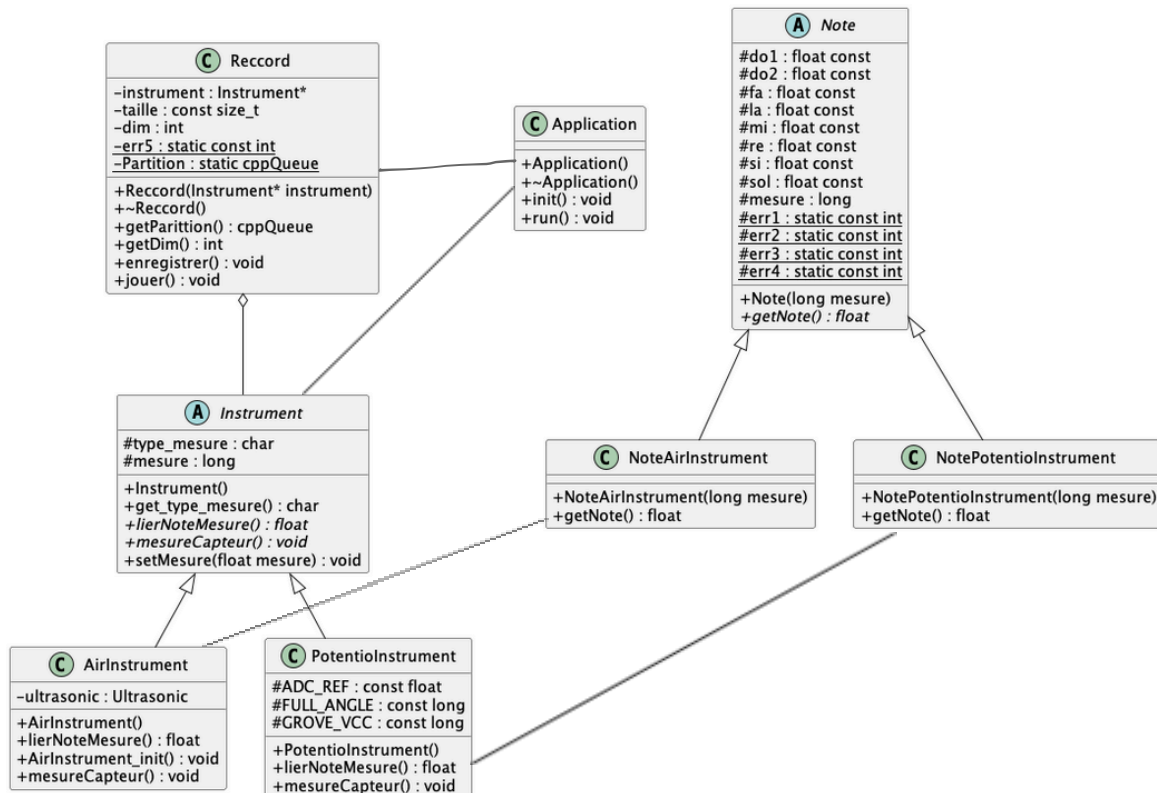


Diagramme de classe du projet

III. Problèmes rencontrés

Le passage à l'IDE arduino pour tester notre code nous a posé divers problèmes. Nous avons tout d'abord eu des erreurs dues aux librairies arduino et aux librairies des capteurs utilisés (notamment pour le capteur ultrason). Le compilateur nous indiquait des problèmes de redéfinitions non autorisées dans des librairies auxquelles nous n'avions normalement pas accès. Nous en avons conclu que ce conflit pouvait venir d'une différence entre les versions des librairies utilisées et la version de l'interface utilisée. Nous avons donc choisi de revenir à une version plus ancienne de l'IDE arduino (version 1.8.19).

Nous avons également eu des problèmes de compilation sous l'IDE arduino, car certaines librairies C++ n'étaient pas reconnues pour arduino. Nous avons en effet défini une queue pour l'enregistrement des notes. Cependant, la librairie queue n'existe pas sur arduino. De fait, nous avons dû chercher une librairie équivalente à celle de la queue mais compatible avec arduino. Cependant, les méthodes et même les constructeurs ne sont pas les mêmes entre la librairie queue de base et celle trouvée sur internet. Ainsi, nous avons dû tout convertir pour que le code compile avec la nouvelle librairie .

IV. Conclusion

Ce projet nous a permis de bien mettre en application les notions de programmation objet apprises durant le cours. Cependant, à cause des problèmes de bog avec l'interface arduino, nous n'avons pas pu prendre le temps d'améliorer les fonctionnalités de notre projet, comme par exemple proposer d'autres modes pour jouer les notes ou utiliser d'autres capteurs comme un écran pour afficher les notes jouées. Le buzzer étant assez limité pour jouer des beaux sons, un autre périphérique aurait donc été plus approprié pour ce genre de projet.