Notice

I. Arduino

A. Détail des bibliothèques

- *PinChangeInt* :
  Permet d'utiliser une interruption sur une Pin.

- *eHealthClass* :
  Regroupe les méthodes relatives à la plateforme de capteurs : initialisation des capteurs, accès à la valeur d'un capteur.

- *MsTimer2 :*
  Regroupe les méthodes relatives au fonctionnement du timer2 ; notamment pour l'utilisation du timer2 en interruption.

- *Monitor :*
  Regroupe les méthodes relatives à la surveillance des données des capteurs : contrôle de la mesure afin de savoir si celle-ci est dans sa plage normale de valeurs, détecter une anormalité, lui associer un code d'alerte, contrôler le bon fonctionnement des capteurs (capteurs non débranchés).

- *SoftwareSerial*
  Permet d'utiliser la liaison série du module Bluetooth : initialiser, envoyer, recevoir des données.

B. Fonctionnement global

Toutes les secondes, les mesures sont mises à jour : bpm, spo2, airflow. On regarde alors si ces mesures sont « normales ».

- Si oui, on envoie ces données vers la tablette.
- Si non : on augmente la fréquence d'échantillonnage des mesures afin d'avoir un aperçu plus précis de ce qu'il se passe. On contrôle le bon fonctionnement des capteurs et on envoie. Si au bout de 10 secondes les mesures ne sont toujours pas bonnes, on déclenche l'alerte et on envoie le code d'alerte vers la tablette.

C. Format des trames

Afin d'envoyer toutes les secondes les données, un format de trame a été créé afin d'envoyer les données dans le même ordre. Le format est le suivant :

- 0xFFFF : début de la trame
- « \n » : caractère séparateur de données dans la trame
- 16 bits (int Arduino) : Code Alerte représentant le problème détecté
- « \n »
- 16 bits (int Arduino) : correspondant à la valeur du pouls
- « \n »
- 16 bits (int Arduino) correspondant à la valeur du SPO2
- « \n »
- 16 bits (int Arduino) correspondant à la valeur du airflow.
- « \n »
- 16 bits correspondant à la valeur de la période d'échantillonnage.
- « \n »

Cette trame sera décodée dans l'application mobile, à la réception de la trame.

II. Application tablette

A. Fonctionnement de l'application

L'application a été créée avec Qt Creator en langage C++ associé au langage Qml et javascript. Le C++ est utilisé pour les calculs dans le programme principal alors que le qml est utilisée seulement pour l'interface graphique et l'affichage.

L'Arduino envoie des données par Bluetooth sous forme de trame comme vue précédemment. A la réception d'une nouvelle trame, celle-ci va être décodé afin d'extraire les valeurs des capteurs, le type de l'alerte si un problème a été détecté. Lorsque les données ont été extraites, elles vont être transférées du côté du qml pour affichageSi un code alerte est décodé, le détail de l'alerte est enregistré. Une alerte est composée du type de l'alerte, de son numéro d'alerte (alerte 1, 2…), des valeurs de pouls, SPO2 et airflow au moment de l'alerte, de trois buffers contenant les valeurs de pouls, SPO2 et airflow avant et après l'alerte afin de retracer l'historique de celle-ci. Elle se compose également d'une variable qui permet de savoir si l'acquisition de tous les points des buffers a été effectuée. Après que l'alerte ait été enregistrée, elle va être affichée en qml avec la courbe des trois grandeurs

B. Détail des classes

Ci-dessous se trouve le détail des classes C++:

- *RemoteSelector* :
  Permet de rechercher les services Bluetooth environnant afin de trouver le service procuré par l'Arduino (service de « Serial Port Profile »).

- *Client* :
  Permet de se connecter à un service.

- Bluetooth :
  Regroupe les deux précédentes classes afin de se connecter à l'Arduino, initialise le Bluetooth de la tablette, permet de recevoir et envoyer des données.

- DataMemory :
  Regroupe les méthodes permettant de sauvegarder les alertes, de lire la mémoire des alertes ou d'aller chercher une information en mémoire

- *QmlData :*
  Classe contenant les informations relatives à une alerte. Une alerte est composée du type de l'alerte, de son numéro, des valeurs de pouls, SPO2 et airflow au moment de l'alerte, de trois buffers contenant les valeurs de pouls, SPO2 et airflow avant et après l'alerte afin de retracer l'historique de l'alerte.

Elle se compose également d'une variable qui permet de savoir si l'acquisition de tous les points des buffers a été effectuée.

- *BtData* :
  Permet de trier les données reçues par le Bluetooth, de décoder la trame reçue. Elle fait le lien entre les données reçues, les données à sauvegarder et celle à envoyer vers le qml pour l'affichage comme par exemple la valeur du pouls, du SPO2 et d'Airflow en temps réel.

- *ScreenDetails* :
  Contient les informations relatives à l'écran : sa taille (hauteur et largeur) ou son orientation.

- *Display :*
  Cette classe contient les éléments visuels de l'application comme la taille de l'écran ou un modèle pour l'affichage des alertes.

C. Détail du Qml

La partie Qml se compose de trois gros fichiers, qui correspondent aux trois fenêtres de l'application :

- Accueil.qml :
  Correspond à la page d'accueil de l'application.

- UserInterface.qml
  Correspond à la page où les données ainsi que les courbes temps réelles sont affichées.

- UserAlertes :
  Correspond à la page où l'historique des alertes est affiché. Contient également, lorsque l'on clique sur une alerte, son détail ; c'est à ce qu'il s'est passé avant et après l'alerte.

D. Diagramme de classe

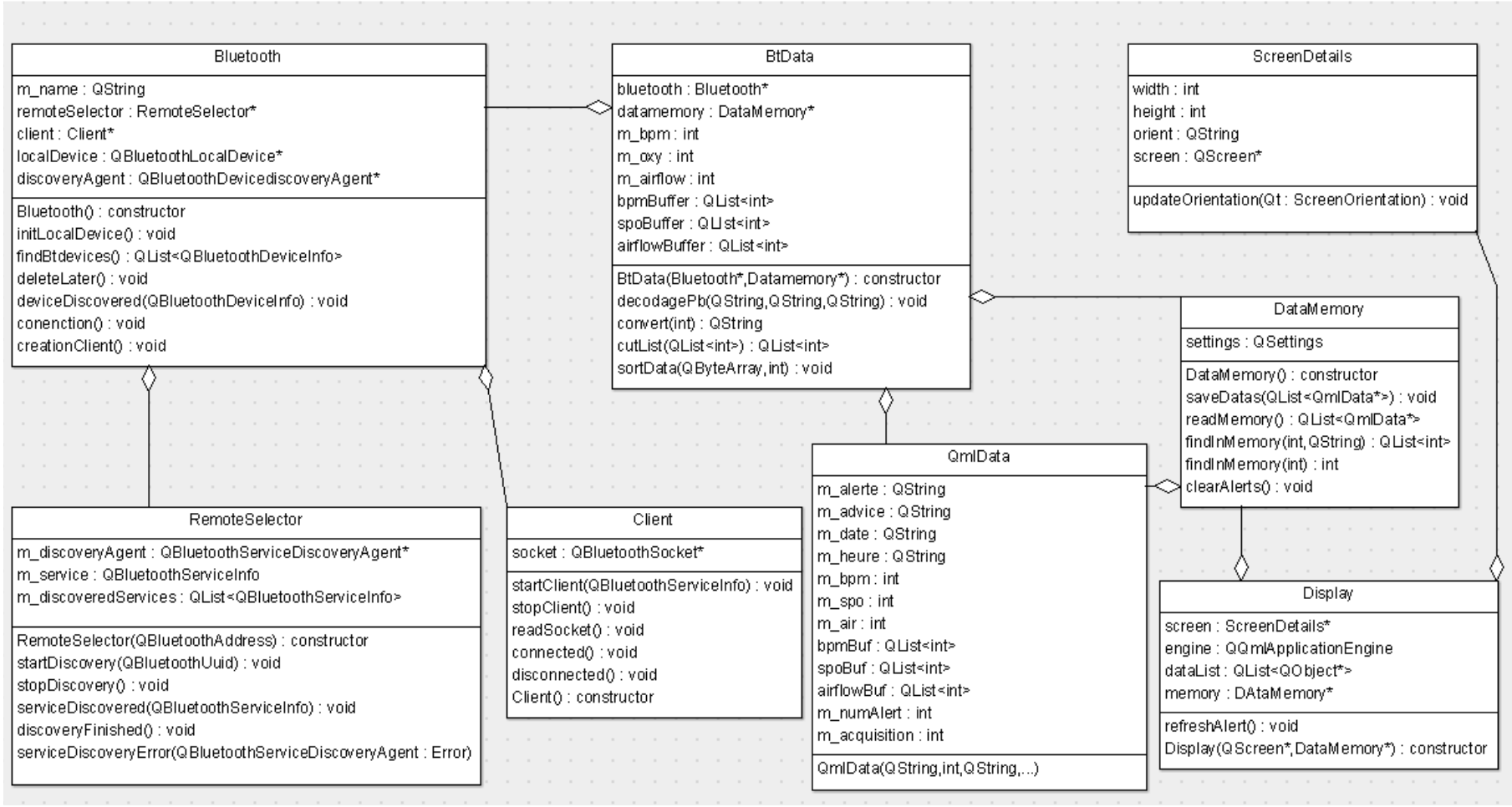

Figure 1: Diagramme de classes de l'application

E. Diagramme de séquence

a. Cas où il n'y a pas de problèmes

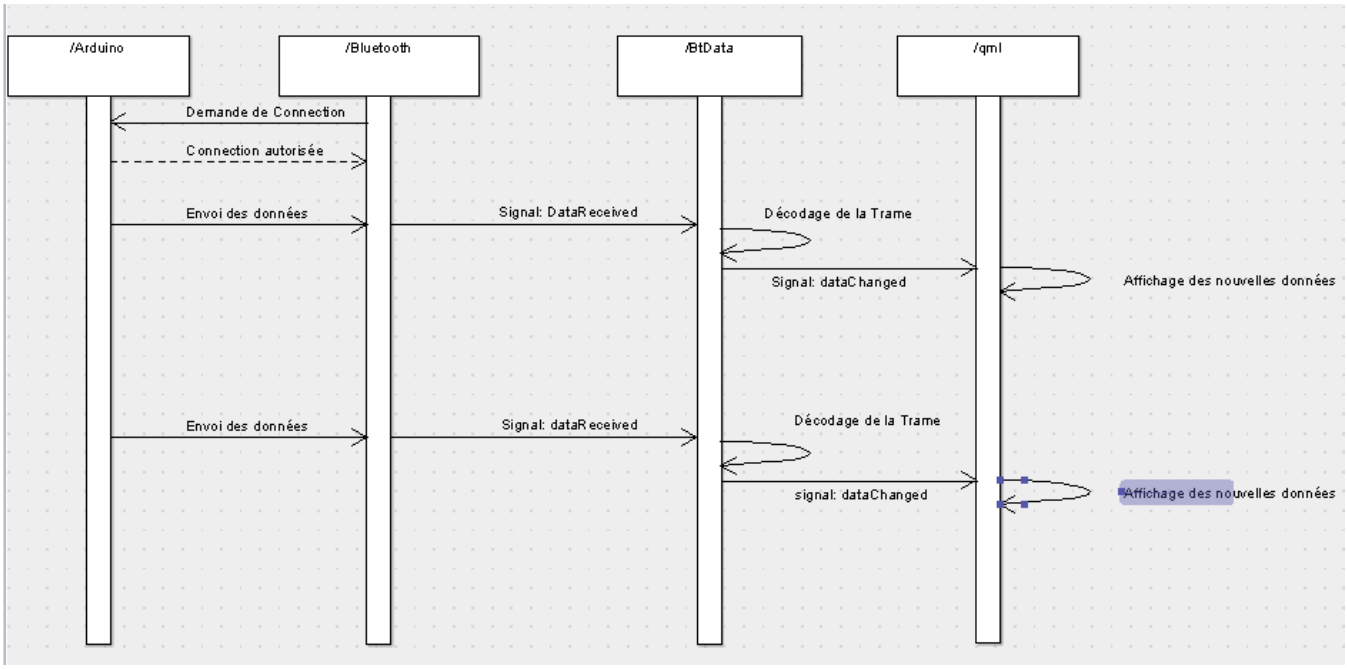

Figure 2: Diagramme de séquence lorsqu'aucun problème n'est détecté

b. Cas où un problème a été détecté

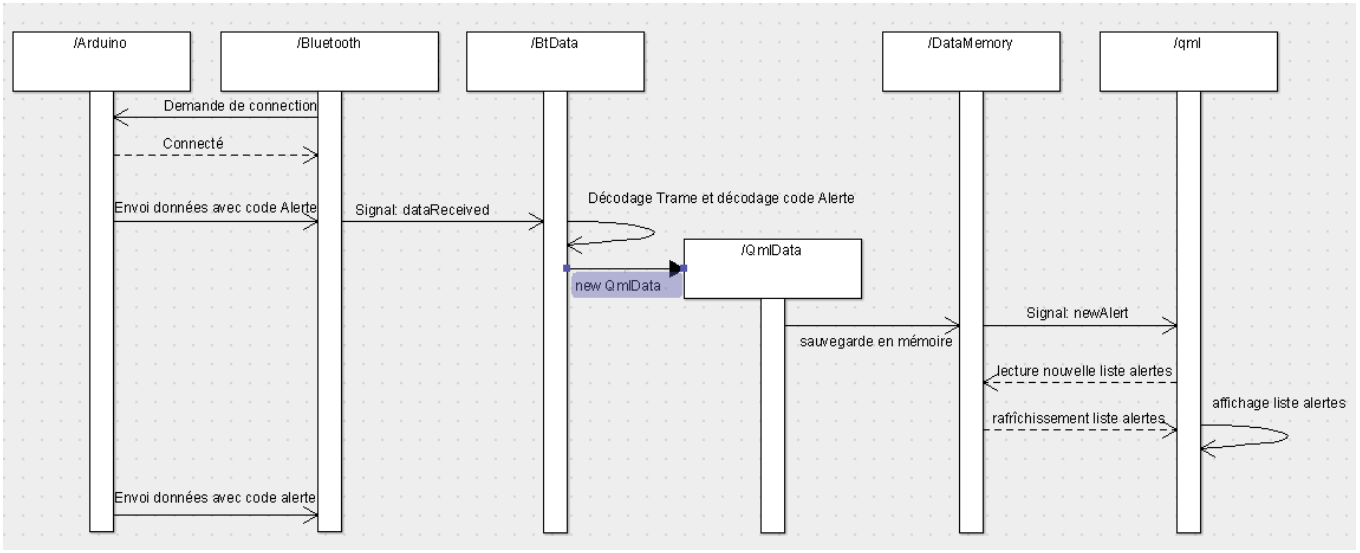

Figure 3: Diagramme de séquence lorsqu'un problème a été détecté

III. FAQ

1. Une minute après l'ouverture de l'application, l'application ne se connecte pas à l'Arduino.

Si la LED rouge du module Bluetooth de l'Arduino clignote toutes les secondes, vous n'êtes pas connecté, réessayer de vous connecter.
Si la LED rouge du module Bluetooth de l'Arduino clignote deux coups rapides puis s'éteint, puis reclignote deux coups rapides… appuyer sur le bouton RESET de l'Arduino et reconnectez-vous depuis l'application.

2. Je suis connecté à l'Arduino mais je ne reçois aucune donnée : les valeurs temps réelles restent à 0 ou sont incohérentes.

Appuyez sur le bouton RESET de l'Arduino.