AER8500 - Informatique embarquée de l'avionique

Mini-projet :

Plan de développement

Arthur Van Betsbrugge 1955439

Katherine Zamudio-Turcotte 1951929

Travail présenté à:

Habacuc Honvo

Polytechnique Montréal

Le 20 février 2023

1. Description de la solution

1.1 Architecture logicielle générale

Le mini-projet consiste à implémenter les protocoles ARINC429 et AFDX au sein d'un système avionique simplifié. Notre solution propose alors de séparer les composantes principales, soit l'agrégateur, le maître AFDX et le calculateur, dans des applications logicielles indépendantes dans le but de simuler le fait que ces composantes seraient des boîtes indépendantes dans un avion réel. La figure 1 illustre notre architecture, dans laquelle l'agrégateur s'occupe des interactions avec l'utilisateur et le calculateur s'occupe des calculs des paramètres de vol.

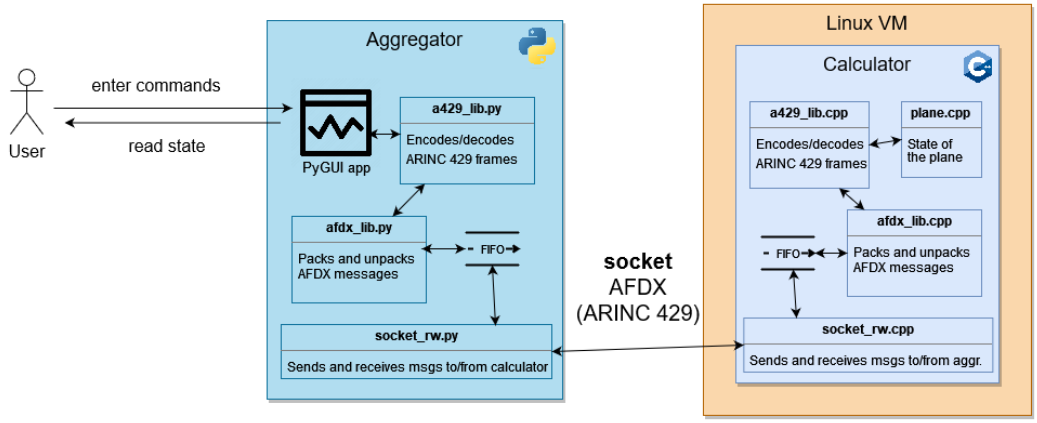

Figure 1. Architecture générale du mini-projet

Pour la transmission des paquets AFDX, nous allons tirer profit d'une connexion réseau réelle entre le calculateur (qui se trouve dans une machine virtuelle Linux sur l'hôte) et l'agrégateur (qui est l'hôte) pour attribuer des adresses IP aux 2 machines et les faire communiquer dans un réseau fermé avec des paquets IP. Le réseau fermé sera le réseau de VirtualBox et le commutateur Ethernet sera le commutateur virtuel de VirtualBox ou la carte réseau de l'hôte. La communication réseau sera effectuée à l'aide d'un *socket* réseau bidirectionnel. Un *socket* additionnel peut être ajouté pour ajouter de la redondance. Un tel *socket* peut être créé et utilisé avec la librairie *socket* de Python et la librairie POSIX *socket* en C. Les paquets IP (AFDX) encapsuleront des trames ARINC429 qui doivent être encodées par le composant source et décodées par le composant destination (afdx_lib.py/cpp). Le codage et le décodage des trames ARINC429 se fera aussi dans l'agrégateur et dans le calculateur (a429_lib.py/cpp).

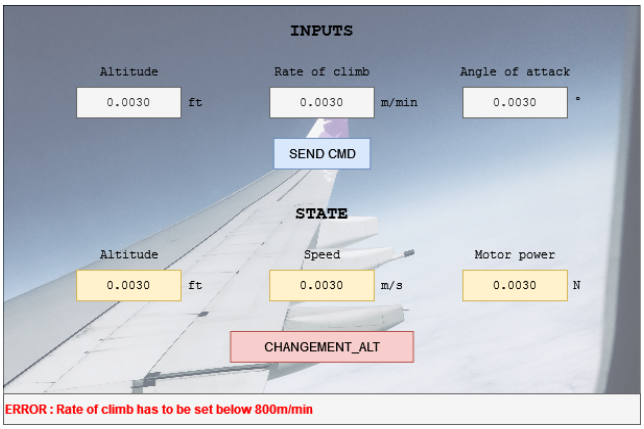

Figure 2. Allure générale attendue pour l'application graphique

L'agrégateur comportera une interface visuelle faite avec l'API PyGUI qui permettra à l'utilisateur de voir et interagir avec le système. La figure 2 démontre l'allure générale de cet affichage.

1.2 Liste des signaux

Les points suivants listent les signaux que nous prendrons en compte dans le développement de notre application. Par signaux, nous voulons dire les données qui seront transmises par messages ARINC429.

• État (chaîne de caractères) • Altitude (pi) • Vitesse ou taux de montée (m/min)	• Puissance des moteurs (%) • Angle d'attaque (°)

Alors que l'altitude, le taux de montée et l'angle d'attaque sont à la fois des entrées et des sorties de l'agrégateur, la puissance des moteurs est seulement une sortie de l'agrégateur et l'état (au sol, en vol ou en changement d'altitude) est seulement une sortie du calculateur.

1.3 Justification des choix de langages

En ce qui concerne l'agrégateur et son interface visuelle, nous avons choisi le langage Python en raison de sa facilité d'utilisation pour encoder des messages ARINC429 et pour utiliser des sockets, en plus de son API PyGUI facile à utiliser pour les applications graphiques. De plus, c'est un langage que nous avons beaucoup utilisé jusqu'à présent. Pour le calculateur, nous avons choisi le langage C++, puisque c'est le langage de prédilection pour plusieurs systèmes embarqués, en plus du fait qu'il permet d'utiliser la librairie POSIX en C pour des appels-système. C'est aussi un langage que nous avons beaucoup utilisé jusqu'à présent, facilitant ainsi le développement.

2. Tâches planifiées et échéances

Tableau 1. Tâches planifiées et échéances du mini-projet

Tâche	Échéance
Implémentation de l'extraction et de l'empaquetage de trames ARINC 429 pour l'agrégateur et le calculateur	6 mars 2023
Implémentation des *sockets* pour la communication inter-composants	20 mars 2023
Intégration du maître AFDX pour l'arbitrage des paquets sur le réseau	27 mars 2023
Ajout de la logique de calcul dans le calculateur	3 avril 2023
Ajout de l'interface utilisateur	3 avril 2023

Peaufinage de la robustesse	17 avril 2023
Rédaction du rapport	17 avril 2023