

Développement d'un Driver logiciel pour piloter un équipement de test

Stagiaire : ANKA SOUBAAI Abdelmajid – 4AE SE
Tuteur : LEIB Anthony
Période : 12/06/2023 – 15/09/2023

HENSOLDT NEXEYA FRANCE
Golf Park - Bâtiment F - 1 rond-point du Général
Eisenhower - 31100 Toulouse

CONTEXTE :

Nexeya Hensoldt est le résultat de la fusion de Nexeya et de Hensoldt en 2020. Elle se spécialise dans l'électronique de défense, offrant des systèmes radar avancés, des communications sécurisées, des solutions de guerre électronique et des avioniques critiques. L'entreprise joue un rôle clé dans la sécurité nationale et internationale grâce à ses technologies de pointe.

MA MISSION :

La mission consistait à développer un driver d'un cube UEIDAQ sur le logiciel Kallisté afin de permettre la lecture et l'écriture de données. Parallèlement, le travail incluait la création d'interfaces homme-machine (IHM) pour simplifier l'utilisation du driver. L'objectif global était d'améliorer l'environnement de Kallisté en rendant le driver plus fonctionnel, facilitant ainsi la manipulation des données.

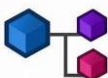
OUTILS UTILISES :



KALLISTÉ, un logiciel propre à Nexeya Hensoldt, m'a servi à tester mon driver et à concevoir des IHM pour en faciliter l'utilisation. Adapté à divers secteurs industriels, il offre une flexibilité essentielle pour répondre aux besoins actuels et futurs.

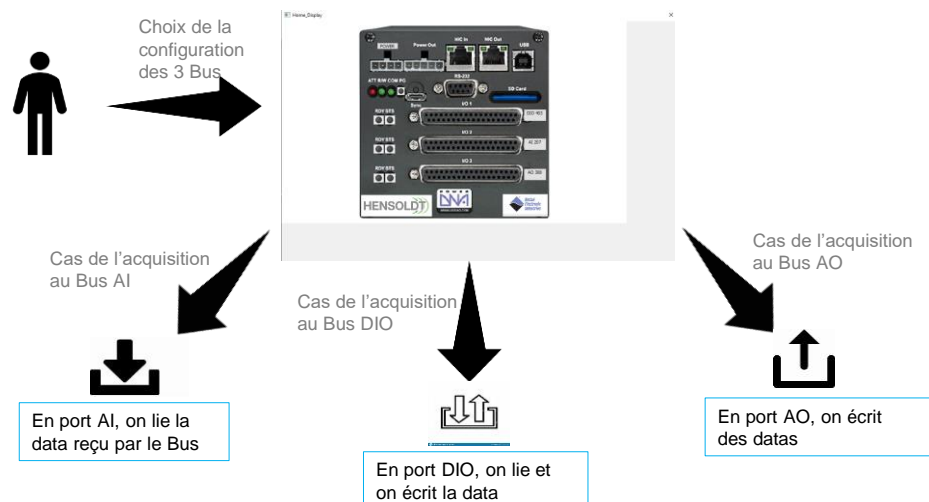


QT a été employé pour la programmation du driver et l'intégration du SDK Kallisté, tandis que Dependency Walker a servi à gérer les dépendances des DLL du driver lors des tests. Le choix de Kallisté a démontré sa compatibilité avec les API d'UEIDAQ.



Le développement du driver a impliqué l'utilisation des langages C et C++, avec des fichiers C++ contenant des méthodes interagissant avec des fonctions en C. Afin d'assurer l'interopérabilité entre l'API UEI en C et le SDK Kallisté en C++, un Wrapper a été créé pour définir ces fonctions en C et les invoquer en C++.

REPRESENTATION DU SYSTÈME :



Mots clé:

- AI** : Analog input
- AO** : Analog Output
- DIO** : Digital Input Output
- BUS** : Canal de communication qui permet le transfert de données entre les composants électroniques

On démarre par une page d'accueil (Home Display) où l'utilisateur peut mettre en marche et éteindre le cube d'acquisition. L'objectif principal de cette étape est de fournir un point de contrôle centralisé pour l'activation du système. De plus, l'utilisateur peut spécifier les adresses IP pour définir la destination des données.

L'IHM permet aux utilisateurs de spécifier les paramètres de chaque canal, notamment en configurant les adresses IP pour diriger les données vers des destinations spécifiques. De plus, l'interface offre la possibilité de définir les ports digitaux en entrée ou en sortie pour une gestion précise des flux de données.

Une fois les paramètres configurés, l'utilisateur peut passer à l'acquisition de données analogiques (AO,AI) et Digitales (DIO). Les utilisateurs peuvent lire en temps réel et éventuellement écrire des données. Cette étape vise à permettre une acquisition précise et souple des données, adaptée aux besoins de l'utilisateur.

L'IHM propose une interface intuitive avec des boutons, cadrans et indicateurs pour régler les paramètres, configurer les adresses IP, et gérer les ports en entrée/sortie. Son objectif est de simplifier l'utilisation tout en permettant une gestion précise des données.

RESSOURCES

Ce guide d'utilisation vous permettra d'accéder à toutes les interfaces et de trouver des réponses à vos questions concernant le système. Il offre une vue complète et détaillée des fonctionnalités expliquées auparavant :



RESULTATS

L'objectif du stage a été atteint : notre système a été testé avec succès en effectuant des bouclages sur les trois bus du cube. Cela signifie que notre système est opérationnel et capable d'envoyer et de recevoir des signaux de manière fiable.



Une amélioration possible serait de permettre la modification en temps réel de la configuration des ports numériques pendant l'acquisition.

Connaissances théoriques mises en pratique :

- Programation C/C++
- Communications réseaux
- Electronique

APPORTS PERSONNELS

Nouvelles compétences acquises :

- Conception d'IHM
- Développement d'application
- Utilisation des DLL
- Prise en main de QT et de Dependency Walker

Ainsi, ce stage fut très enrichissant puisqu'il m'a permis de gérer en autonomie un projet dans sa globalité (conception, développement, tests, déploiement), et aussi de me familiariser au domaine du développement applicatif.