



Département de génie informatique et génie logiciel

INF1900
Projet initial de système embarqué

Rapport final de projet

Projet dans SimulIDE

Équipe No < **0712** >

Section de laboratoire < **01** >

Paul Marie Akoffodji
Mike Davy Folepe Dikko
Jean-Marc Ndong .É
Etienne Chevalier

8 décembre 2020

1. Description de la structure du code et de son fonctionnement

Afin de réaliser le système de ce projet , nous avons adopté une méthodologie qui nous a permis de définir la structure de notre code et son fonctionnement c'est-à-dire la façon dont nous l'avons organisé et les interactions qui existe entre ses différentes composantes et qui régissent son comportement .

Le but de cette section de notre rapport sera donc d'expliquer cette méthodologie qui nous a permis de réaliser le système.

Notre code pour la réalisation de ce projet est structuré en différents fichiers permettant de contrôler les composantes et les actions qui sont dans notre librairie et un fichier principal qui est dans le répertoire projet et qui utilise les fichiers de la librairie pour faire fonctionner le système. Pour obtenir cette structure , nous avons procédé de la sorte :

Premièrement , nous avons commencé par écrire le code qui permet de faire fonctionner les composantes du système qui sont le sonar , l'horloge , le servomoteur , les barres de DEL , le clavier et la LCD et ces parties de codes ont été réparties en fichier . h (fichier qui contiennent la signature des fonctions nécessaires pour le fonctionnement des composantes) et des fichiers .cpp (qui contiennent l'implémentation des dites fonctions) que nous avons ajouté à notre librairie pour faciliter leurs utilisations partout dans le code et conserver la propreté dans notre code principale .

Deuxièmement , nous avons écrit du code pour effectuer les différentes actions du mode configuration du système . Et pour ce faire nous avons utilisé les parties de code que nous avons écrites pour le fonctionnement des composantes du système dans notre librairie . Ainsi nous avons écrit des parties de code qui utilisent les composantes du système pour faire fonctionner le mode configuration du système et ces parties de code , nous les avons ajoutées à notre librairie pour pouvoir utiliser les fonctions des options dans notre code principale pour éviter de surcharger ce dernier et garder cette structure de fonctionnement qui utilisent les fonctions réimplémentés dans la librairie. On retrouve donc dans la librairie des fichiers qui contiennent la signature et l'implémentation des fonctions qui utilisent les fichiers de la même librairie qui implémente le fonctionnement des composantes nécessaires pour la réalisation de chaque action .

Les fichiers de la librairie qui implémentent le comportement des actions et leurs relations avec les composantes sont les suivants :

- Les fichiers Horloge.h et Horloge.cpp implémentent en même temps les fonctions pour le fonctionnement de l'horloge et les fonctions pour l'exécution de l'option 1 qui est de définir l'heure de départ de la simulation en utilisant le fonctionnement du clavier implémenté dans clavier_lib.h et .cpp .
- Les fichiers Option2.h et Option2.cpp implémentent les fonctions pour l'exécution de l'option 2 qui est de définir l'état d'un dispositif en utilisant les fonctions des fichiers servoMoteur.h et .cpp , barresDEL.h et .cpp et clavier_lib.h et .cpp qui implémentent les comportements du servomoteur et des barres de DEL et du clavier nécessaires pour la réalisation de cette option.

- Les fichiers `action_lib.h` et `.cpp` implémentent les fonctions pour la réalisation des options 3 , 4 et 5 qui sont respectivement : afficher les actions préprogrammées, programmer les actions et supprimer les actions préprogrammées en créant une classe `Action` qui permet donc de définir les différentes opérations qu'on applique à un objet de cette classe.

Troisièmement , nous avons créé deux fichiers `Projet .h` et `Projet.cpp` que nous avons ajouté à notre librairie et qui incluent tous les fichiers de la librairie qui sont nécessaires pour le fonctionnement des composantes et des actions et en plus implémentent les fonctions qui permettent le fonctionnement du mode simulation du système que l'on pourra utiliser dans notre main afin de réduire la quantité de code , de faciliter la compréhension , le garder propre et bien organisé.

Finalement, nous avons créé un répertoire `Projet` qui contient le fichier `main.cpp` et qui est donc le fichier qui contient notre code principal pour le fonctionnement du système complet. Il inclut le fichier `Projet.h` , ce qui lui permet d'avoir accès à toutes les fonctions permettant de faire fonctionner les composantes, les actions du mode configuration et aussi le mode simulation du système. Ce dernier se structure comme suit :

- Il comporte dans les premières lignes , l'Initialisation des variables volatiles utilisés dans les routines d'interruption.
- Routine d'interruption lié au sonar (`INT0_vect`) fonctionne de façon à enclencher l'interruption quand l'onde d'écho passe à 0 pour pouvoir calculer la distance par rapport à un objet
- Routine d'interruption lié à l'horloge (`TIMER1_COMPA_vect`) fonctionne de façon à arrêter l'horloge après 24h
- Routine d'interruption lié au bouton poussoir D3 (`INT1_vect`) fonctionne de façon à arrêter la simulation prématurément.
- Initialisation des ports et des interruptions
- Initialisation des variables utilisés tout au long du programme , pour la conversion analogique numérique et pour le réglage de la vitesse de l'horloge.
- Structure de switch case permettant de changer l'état actuel du système entre configuration et simulation
- État configuration comporte le code affichant le menu ainsi que le code pour l'exécution des options
- État simulation comporte le code permettant d'ajuster la vitesse d'avancement de l'horloge , d'exécuter les actions préprogrammées , les interruptions sur le sonar , le bouton poussoir et l'horloge

Le diagramme suivant démontre la structure et les relations entre les parties de notre code ayant permis de réaliser le système

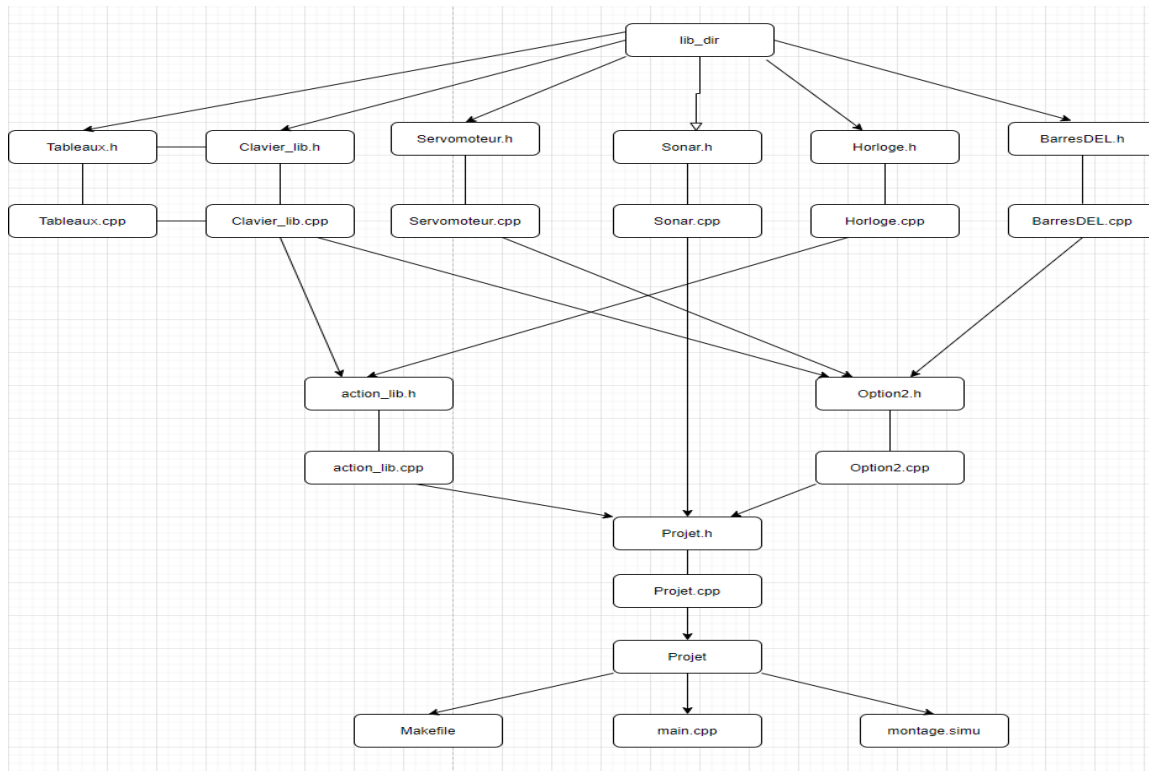


Figure 1 : Diagramme de structure de code

En analysant le diagramme ci-dessus, on est capable de bien comprendre la structure hiérarchique et comportementale de notre code. On retrouve donc à la racine de notre entrepôt git les fichiers `lib_dir` (fichier contenant notre librairie) et `Projet` (fichier contenant le main de notre projet, le makefile et le `.simu`).

2. Expérience de travail à distance

Cette expérience fut très intéressante, en effet cette session que nous venons de passer était comme un saut dans l'inconnu pour tout le monde autant pour les étudiants que pour les membres du personnel éducatif. De nombreuses contraintes et défis ont été au rendez-vous. En premier lieu la motivation fut l'un des atouts les plus durs à développer au cours de cette session. Le contexte pandémique, la monotonie des journées ont beaucoup joué sur le moral. Il était parfois dur de se lever très tôt le matin même si on n'avait pas à se déplacer pour se rendre à Polytechnique c'est comme si le fait de se réveiller juste de son lit puis de prendre son ordinateur portable était encore plus éprouvant. Nombreuses sont les fois où nous nous sommes réveillés après l'heure parce que on avait du mal au début. Mais il ne fallait pas prendre de retard dès le début de la session connaissant la cadence des cours et l'accumulation rapide de la charge de travail à Polytechnique en plus que nous suivons les cours à distance, fallait redoubler d'ardeur afin de cerner correctement les notions assez rapidement et ne pas être à la traîne. Ceci étant nous avons fait des chargés de laboratoire

nos alliés dans cette quête, durant et en dehors des séances de laboratoire nous posons des questions de compréhension, faisons des rencontres dans les salons virtuels très fréquemment. Ceci a aussi dans un certain sens développé une certaine dynamique dans la gestion des activités liés à la matière en général. Plus la session progresse plus la matière s'accumule le contrôle périodique et les remises s'approchaient à grand pas fallait s'organiser en conséquence pour les révisions. Mais nous nous en sommes sortis pas mal vis-à-vis de cela. Après un certain temps, rendu à la semaine 7, nous avons eu à nous mettre en équipe de quatre pour les derniers travaux pratiques encore là nous faisons comme un deuxième saut dans l'inconnu car le choix de nos partenaires nous le savions tous, est déterminant et irréversible jusqu'à la fin du projet. Fallait donc s'allier avec des personnes ayant l'envie de travailler et de performer au maximum malgré le contexte de travail à distance. Fort de cela nous nous sommes engagés dans ce marathon à quatre nous avons réussi les travaux, développer une bonne dynamique de groupe, ainsi qu'une bonne cohésion. Les membres de l'équipe avaient trouvé leurs places et tout marchait pour le mieux. Nous eûmes l'occasion de nous rencontrer aussi à Polytechnique avec Jérôme Collin afin de discuter un peu nos impressions sur la matière, l'avancement de groupe et d'autres choses. Le projet final aussi était en cours demandant particulièrement une décentralisation des tâches dans un premier temps nous avons eu un peu de retard sur la stratégie à adopter afin de faire le travail demandé. Nos rencontres ont dû devenir plus fréquentes car le temps pressait. Lors des séances de laboratoire qui passait tellement vite on avançait pas mal mais ce n'était pas assez mais néanmoins on a pu sortir des prototypes de code itératif qui fonctionnait plutôt bien afin de faire fonctionner notre circuit. Partant de cela nous devons les combiner cela a été une autre paire de manche car certains bugs apparaissaient fallait s'entendre sur les ports qui serait attribuer à chacun afin de pas créer des interférences. De nombreuses heures de débogage furent allouer aussi afin de savoir dans notre code qu'est-ce qui provoquait parfois des comportements inattendus. C'était un gros travail mais l'équipe fut solide et nous avons pu avancer dans les modules et leurs fusions. Aussi le refactoring du code fut une étape cruciale fallait écrire le code l'optimiser, changer nos variables de fonctions, la portée de nos variables ainsi que beaucoup d'autres petits détails assez importants. Le code a donc été optimisé en veillant à ne pas introduire de nouveaux bogs afin de respecter au maximum les exigences du projet ainsi que les directives émises par le document d'exigence dans l'énoncé de départ. Notre code devrait reproduire dans l'absolu ce qui était spécifié et devrait être assez lisible et facile à interpréter. Bien que l'évaluation du fonctionnement de notre code se fera par des tests en boîte noire dans un premier temps la qualité du code sur lequel repose le fonctionnement dudit système demeurerait aussi un grand enjeu dans l'évaluation du projet. Faire fonctionner le système ne suffirait donc pas à obtenir une très bonne note il faut des critères additionnels. Nous nous sommes donc attelés à cette tâche. Les derniers détails ont été peaufinés en conséquence. Nous avons consacré beaucoup d'heures à ce projet malgré son caractère « virtuel » ceci montre notre engagement dans l'élaboration d'un travail de qualité malgré le fait qu'on soit à distance. Nous nous sommes données à deux cents pour cent. Même à l'heure où j'écris nous sommes encore en train de travailler sur le projet afin d'avoir un bon rendu. L'expérience à distance n'est pas évidente on s'accorderait tous à le dire mais elle a été un challenge pour nous car nous avons pu, même dans l'adversité nous en sortir et offrir le meilleur de nous-mêmes. Et nous sommes très fiers de cela. Nous avons beaucoup appris au cours de cette session et nous gardons de ce projet de très bons souvenirs.