VIALA Luc

RADJOU Omni

3DN ITI ENSEA

2016

Compte-Rendu de Mini-Projet

Gestion d’une centrale inertielle sous Linux

M. PEPIN Frédéric & M. KESSAL Lounis

Table des matières

[1 La carte électronique 4](#_Toc453854503)

[2 La centrale inertielle ALT-IMU10 5](#_Toc453854504)

[2.1 Le capteur LSM303D 6](#_Toc453854505)

[2.1.1 L’accéléromètre 6](#_Toc453854506)

[2.1.2 Le magnétomètre 8](#_Toc453854507)

[2.2 Le gyromètre L3GD20H 9](#_Toc453854508)

[2.3 Le baromètre LPS25H 10](#_Toc453854509)

[3 L’Architecture du projet 11](#_Toc453854510)

[3.1 La configuration de la carte i.mx6Q SABRELite 11](#_Toc453854511)

[3.2 Le programme de la carte i.MX6Q SABRELite 14](#_Toc453854512)

[3.2.1 Architecture de l’application 15](#_Toc453854513)

[3.2.2 Fichier ENSEA\_i2c.c 17](#_Toc453854514)

[3.2.3 Fichiers acc.H & acc.c 18](#_Toc453854515)

[3.2.4 Fichiers gyro.h et gyro.c 20](#_Toc453854516)

[3.2.5 FICHIERS baro.h & baro.c 22](#_Toc453854517)

[3.2.6 Fichier main.c 24](#_Toc453854518)

[3.3 Le programme Labview 25](#_Toc453854519)

Introduction

Dans le cadre de notre 3ème année en formation Ingénieur ITI en alternance à l’ENSEA, dans l’option Systèmes Numérique & Intégration, nous avions à réaliser un mini-projet qui nous permettrait de mettre en application toutes les connaissances acquises durant les 3 années de formation.

Parmi les sujets proposés, nous avons choisi le projet de gestion de centrale inertielle sous Linux, c’est un sujet qui nous tenait à cœur, premièrement pour nous permettre d’appréhender l’environnement Linux, qui est le système d’exploitation le plus utilisé dans les systèmes électroniques embarqués, et deuxièmement pour avoir la possibilité de mettre en place une communication sur bus CAN, qui est le bus de terrain le plus utilisé dans le monde de l’électronique embarquée à l’automobile. En rejoignant ces deux points, nous étions certains de réaliser un projet intéressant et qui nous permettrait d’apprendre énormément sur les systèmes électroniques embarqués.

Le but de ce mini-projet est de mettre en place une application fonctionnant sous le système d’exploitation Linux exploitant le bus I2C pour communiquer avec la centrale inertielle, et renvoyer les informations de cette dernière vers une interface LabView grâce au bus de communication CAN.

# La carte électronique

La carte sur laquelle se base notre projet est une i.MX 6Quad SABRELite développée par la société Freescale.

La particularité de cette carte est tout d’abord son processeur intégré, un ARM Cortex A9 MPCore possédant 4 cœur processeurs cadencés à 1GHz et permettant l’installation du système d’exploitation Linux, les périphériques disponibles de cette carte SABRELite sont également très nombreux et intègre la gestion des bus de communication CAN (Controller Area Network), i2c (Inter-Integrated Circuit) et une interface Ethernet est également disponible.

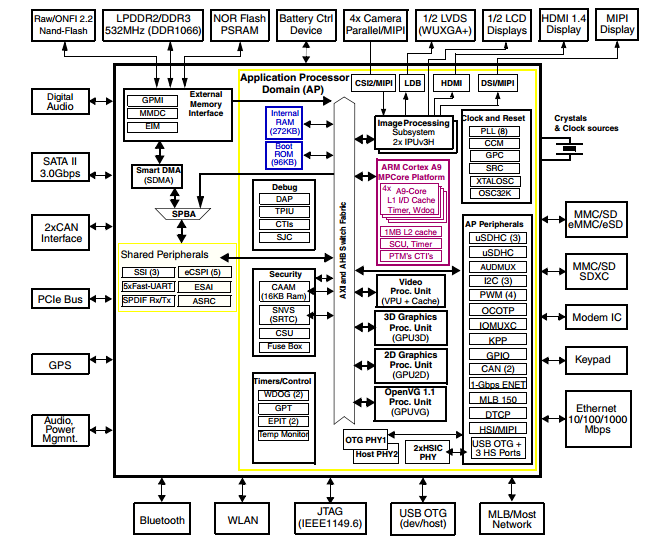


Figure - Architecture de la carte i.MX6Q SABRELite

Pour communiquer avec la carte i.MX6Q SABRELite et le PC de développement, nous utiliserons le port série ou UART également disponible.

# La centrale inertielle ALT-IMU10

La centrale inertielle que nous avons à notre disposition pour réaliser ce projet est la carte AltIMU dans sa version 10 DOF, pour Degrees Of Freedom (degrés de liberté pour les francophones), c’est-à-dire que cette centrale est capable de mesurer 10 grandeurs physiques grâce à la présence de ces différents capteurs.

En effet, la centrale inertielleAltIMU-10 possède trois circuits intégrés, ayant chacun une fonction bien précise.

Les trois circuits intégrés, tous développés par la société STMicroelectronics, à cette centrale inertielle sont les suivants :

* L’accéléromètre/magnétomètre, un LSM303DLHC.
* Le gyromètre, un capteur gyroscopique ayant la dénomination L3GD20H.
* Le baromètre, LPS25H.

Ces trois capteurs intègres la gestion de la communication par bus i2c.



Figure - Centrale inertielle AltIMU-10

## Le capteur LSM303D

L’accéléromètre LSM303D est un capteur 3 axes qui permet de récupérer les valeurs des accélérations linéaires sur ses axes X, Y et Z. Ce circuit intégré possède également un magnétomètre 3 axes, qui permet de mesurer la direction d’un champ magnétique sur ses axes X, Y et Z. Un capteur de température est également intégré dans le circuit.

### L’accéléromètre

Comme son nom l’indique l’accéléromètre permet de récupérer les valeurs des accélérations linéaires sur les axes X, Y et Z.

La technique qui permet de mesurer l’accélération fonctionne grâce au principe de la loi fondamentale de la dynamique :

, avec F : force(N), m : masse (kg) et a : accélération (m/s²)

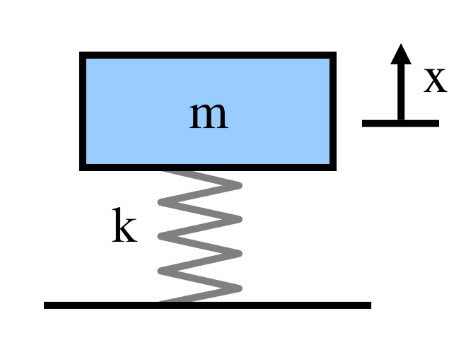


Figure - Principe d'une masse sismique sur l’axe X avec ressort de rappel

En effet, ce capteur utilise ce que l’on appelle une masse sismique liée à un ressort enfermée à l’intérieur d’un carré métallique qui permet de créer un effet capacitif, la contrainte d’une accélération dans un des trois axes a pour effet de modifier la position de cette masse sismique à l’intérieur de son emplacement. Cette modification de position induit une différence dans l’effet capacitif du montage, de cette manière on peut déterminer les forces appliquées à cette masse sismique et par le calcul précédent en déduire l’accélération.

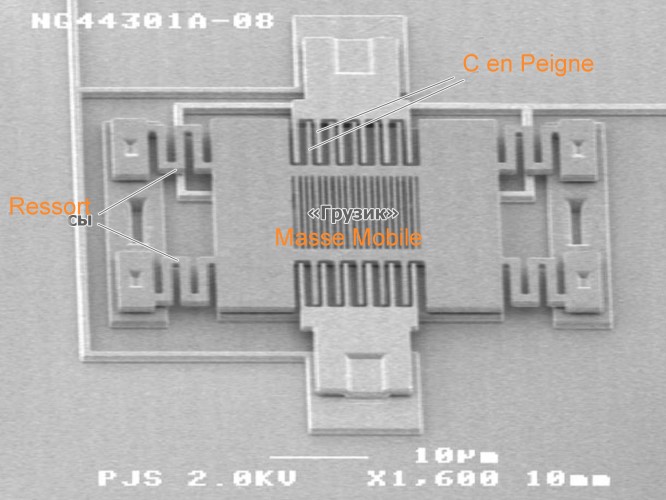


Figure - Masse sismique intégrée à l'intérieur d'un accéléromètre électronique

L’accéléromètre permet de savoir dans quelle direction l’appareil se déplace sur ses axes X, Y et Z.

Comme tout corps sur la surface de la Terre, notre accéléromètre sera soumis en permanence à la gravité terrestre qui correspond à une accélération de 9,81 m/s².

### Le magnétomètre

Le magnétomètre permet de récupérer la direction du capteur sur chacun de ses axes, X, Y et Z par rapport à un flux de champ magnétique. Ici le champ magnétique de référence sera le champ magnétique terrestre.

Le magnétomètre fonctionne grâce au principe du capteur par effet Hall, lorsqu’un champ magnétique B traverse, avec un angle d’incidence, un conducteur parcouru par un courant électrique E, une différence de potentiel apparaît selon l’axe perpendiculaire au plan B-E, en mesurant cette différence de potentiel il est possible de déterminer la direction du champ magnétique.

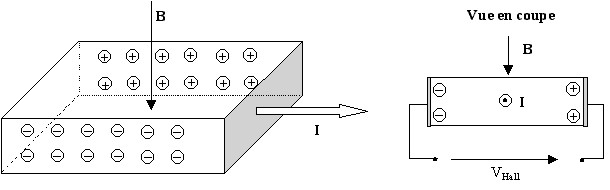


Figure - Principe du magnétomètre par effet Hall

## Le gyromètre L3GD20H

Le gyromètre L3GD20H est un capteur gyroscopique qui permet de mesurer les valeurs des accélérations angulaire autour de ses 3 axes X, Y et Z.

Le principe de fonctionnement du capteur gyroscopique utilise le principe du moment inertiel induit par la force de Coriolis. Deux masses mobiles sont disposées face à face et reliées entre-elles par un ressort fixé en son centre à l’armature. L’application d’une force de rotation sur l’axe perpendiculaire au plan masse-ressort-masse induit une déformation du ressort du à l’inertie des masses, et en raison de l’effet capacitif entre les masses et l’armature il est possible de déterminer l’accélération angulaire de l’axe en rotation.

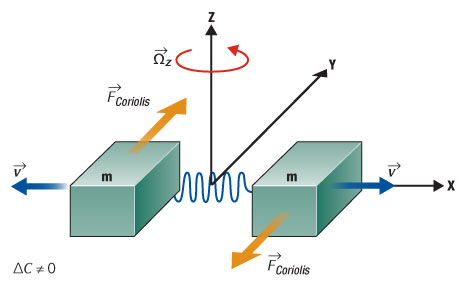


Figure - Principe de fonctionnement du capteur gyroscopique

## Le baromètre LPS25H

Le capteur LPS25H permet de mesurer la pression atmosphérique.

Le principe de fonctionnement du capteur barométrique repose sur le principe de différence de pression entre deux environnements, dont un fermé, un diaphragme placé entre ces deux environnements va subir la pression de l’environnement extérieur et se déformer en fonction de la différence de pression.

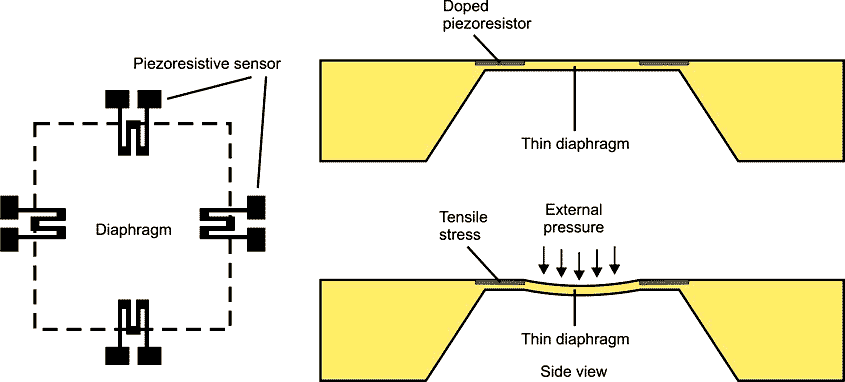


Figure - Principe de fonctionnement d'un capteur de pression

# L’Architecture du projet

L’ordinateur LabView nous permettra de construire une application pour visualiser les informations de la centrale inertielle AltIMU-10 fournies par le biais de la carte SABRELite.

Le programme de la carte i.MX6Q SABRELite devra être capable d’implémenter les communications i2c et CAN, pour lire les informations de la centrale inertielle et les renvoyées vers le PC LabView.

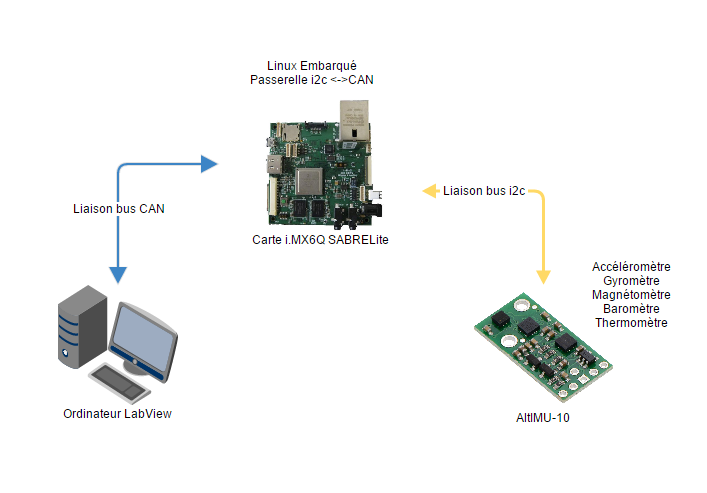


Figure - Schéma d'architecture du projet

## La configuration de la carte i.mx6Q SABRELite

Avant de pouvoir développer notre application, il nous a fallu configurer correctement le noyau Linux de la carte i.MX6Q.

Grâce au TP de Linux embarqué, nous avons été capables de mettre en place dans la configuration du noyau les drivers du bus i2c et du bus CAN. Pour ce faire nous avons modifié le fichier de configuration du noyau Linux avec la commande :

Cette commande nous ouvre une interface graphique qui nous permet de choisir les drivers à intégrer dans le noyau.

Nous avons donc activé l’intégration du bus CAN.

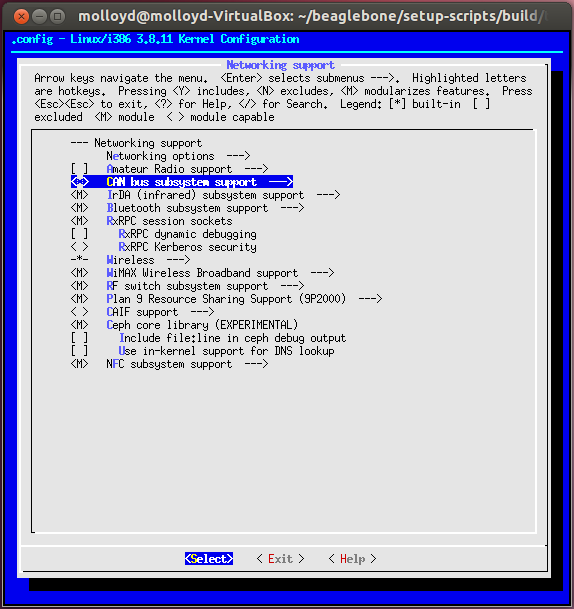


Figure - Ajout du driver CAN bus

Puis l’intégration du bus i2c.



Figure - Ajout du support de communication i2c

Nous avons également réussi à mettre en place le système de fichier en réseau grâce au protocole NFS (Network File System), ce qui nous a permis de développer notre application beaucoup plus rapidement que si nous avions développé directement depuis la carte.



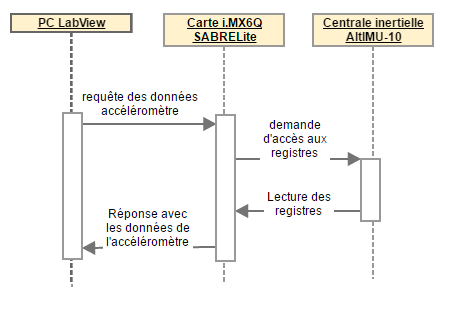
Figure - Ajout du support du protocole NFS

Une fois que la configuration du noyau était terminée nous l’avons compilé afin de créer une nouvelle image Linux intégrant les nouveaux drivers grâce à la commande suivante :

## Le programme de la carte i.MX6Q SABRELite

Pour permettre le transfert des informations de la centrale inertielle AltIMU-10 vers le PC LabView, nous avons dû développer un logiciel qui permet de lire les informations demandées par le PC via le bus de communication CAN afin de demander les informations à la centrale et enfin répondre au PC avec les informations de la centrale.

Pour ce faire, nous avons utilisé les drivers Linux i2c et socketCAN et avons développé notre application grâce au langage de programmation C.



CAN I2C

Figure - Protocole de communication entre les différents éléments

Nous avons développé une surcouche du driver i2c afin de communiquer avec les différents capteurs de la centrale inertielle AltIMU-10 et permettre une utilisation plus aisée du driver. Nous devions à l’origine développer nos propres drivers i2c afin de permettre la communication directe avec l’accéléromètre, le gyroscope, le magnétomètre et le baromètre, cependant les contraintes de temps nous ont poussés à réduire les exigences du projet.

### Architecture de l’application

L’architecture de l’application est définit comme suit :

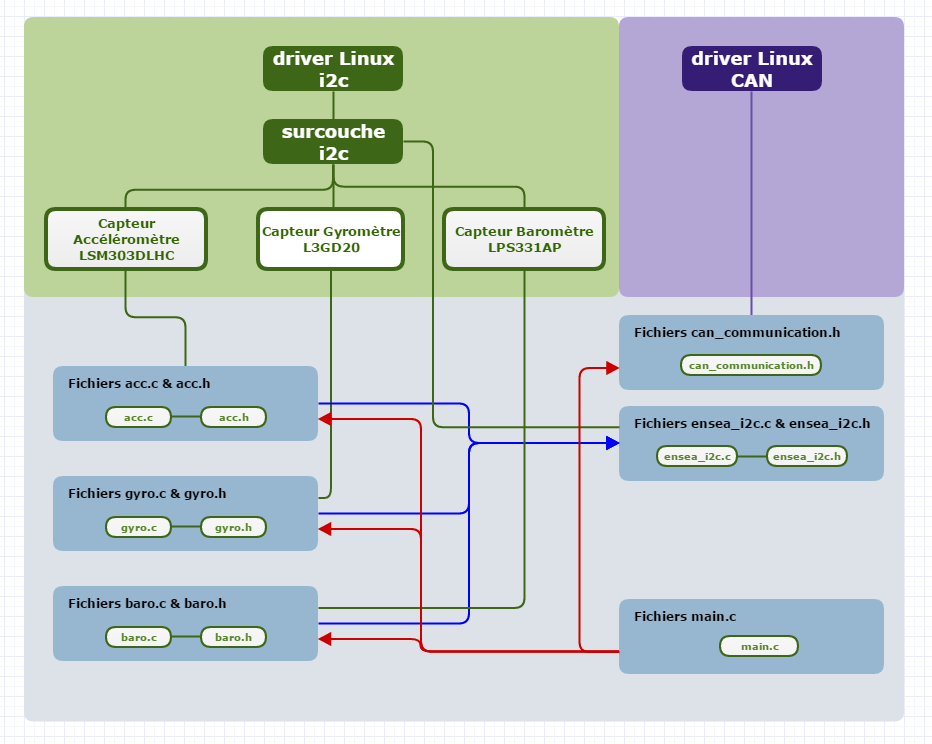


Figure - Schéma d'architecture fonctionnelle de l'application

Le WorkFlow de l’application est déterminé comme suit :

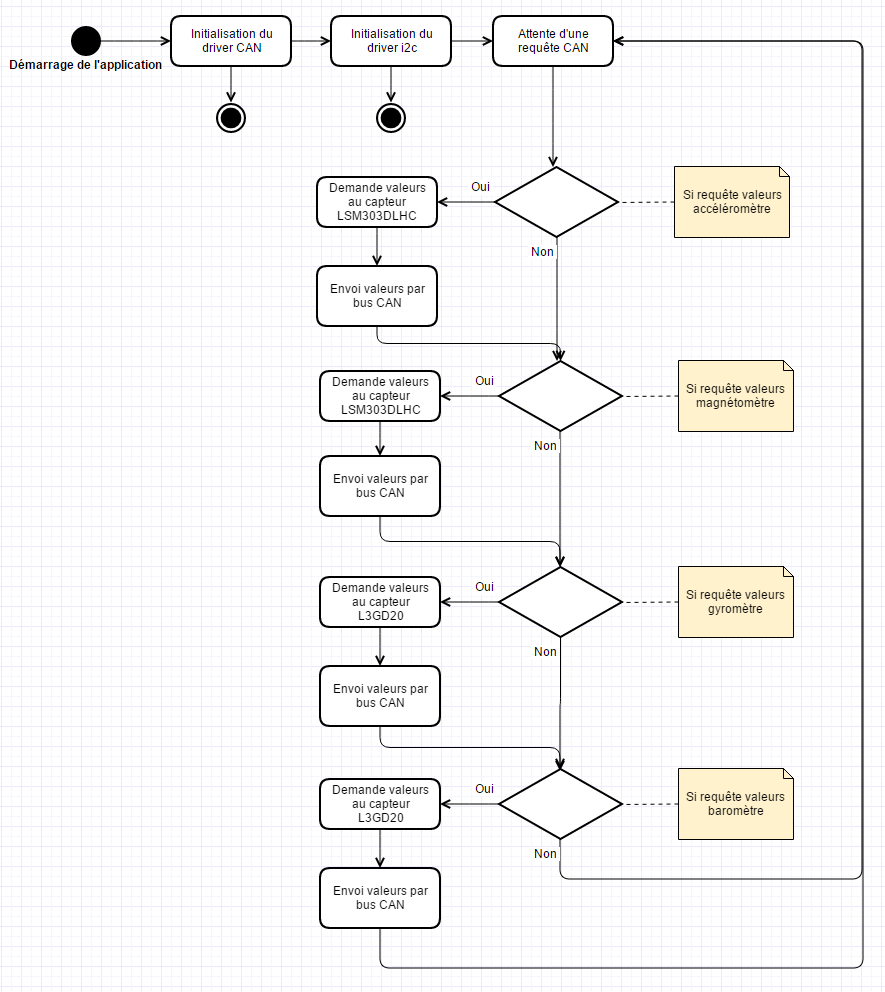


Figure - WorkFlow de l'application

### Fichier ENSEA\_i2c.c

Pour simplifier l’utilisation du bus i2c, nous avons fait le choix de réaliser une surcouche du driver i2c afin de simplifier les différents accès aux équipements i2c. Toutes les fonctions nécessaires à élément se trouvent dans ce fichier ensea\_i2c.c, et chacune des surcouches d’équipement se basera sur les fonctions de ce fichier afin d’uniformiser la communication.



Figure - Surcouche du driver i2c permettant de créer des fonctions adaptées à notre projet

### Fichiers acc.H & acc.c

Le fichier acc.h est une librairie contenant les adresses de tous les registres du capteur accéléromètre et magnétomètre LSM303DLHC ainsi que les prototypes des fonctions et une structure de l’équipement.

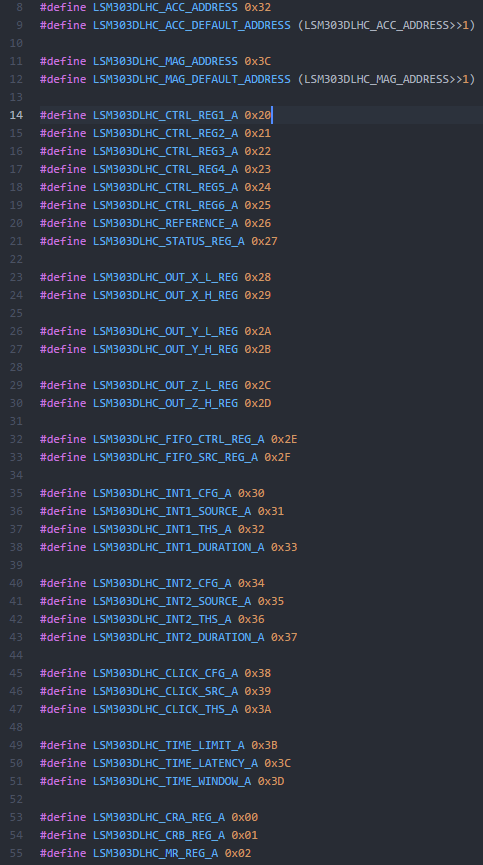


Figure - Extrait du fichier acc.h

Le fichier acc.c contient toutes les fonctions permettant de lire et d’accéder aux différents registres du capteur accéléromètre et magnétomètre de la centrale inertielle.



Figure - FIchier acc.c contenant les fonctions d'accès aux registres de l'accéléromètre et du magnétomètre

### Fichiers gyro.h et gyro.c

Le fichier gyro.h est une librairie contenant les adresses de tous les registres du capteur gyromètre L3GD20 ainsi que les prototypes des fonctions et une structure de l’équipement.

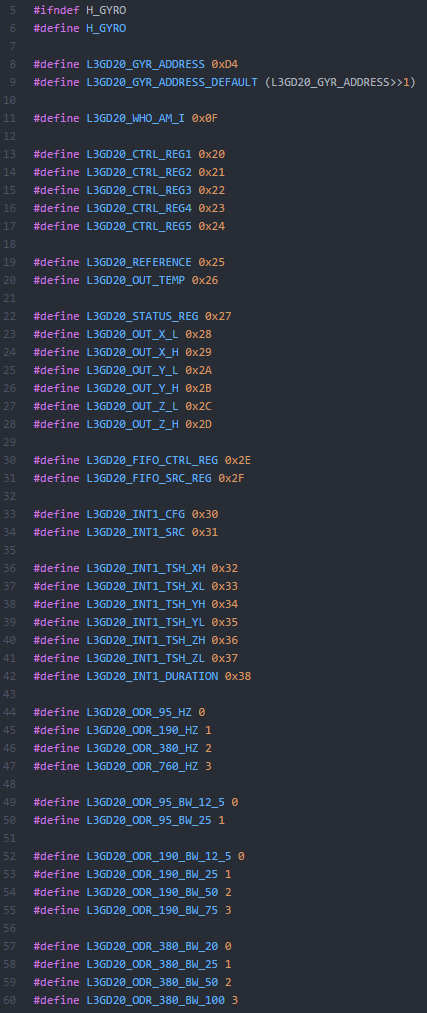


Figure - Extrait du fichier gyro.h

Le fichier gyro.c contient toutes les fonctions permettant de lire et d’accéder aux différents registres du capteur gyromètre de la centrale inertielle.

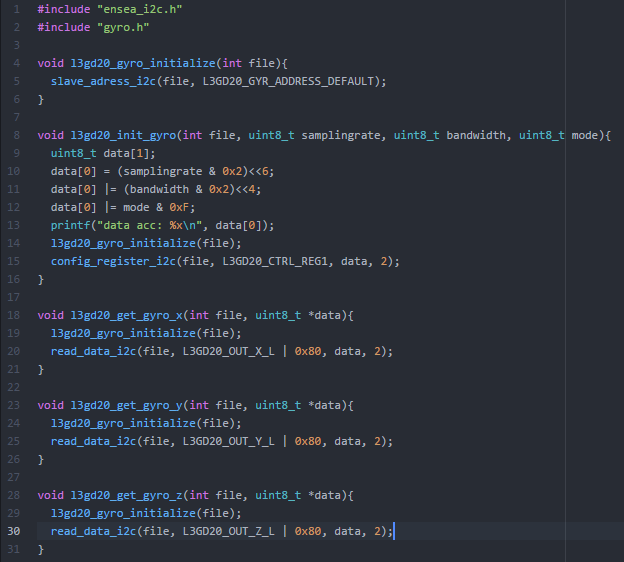


Figure - Fichier gyro.c contenant les fonctions d’accès aux registres du gyromètre

### FICHIERS baro.h & baro.c

Le fichier baro.h est une librairie contenant les adresses de tous les registres du capteur baromètre LPS331AP ainsi que les prototypes des fonctions et une structure de l’équipement.

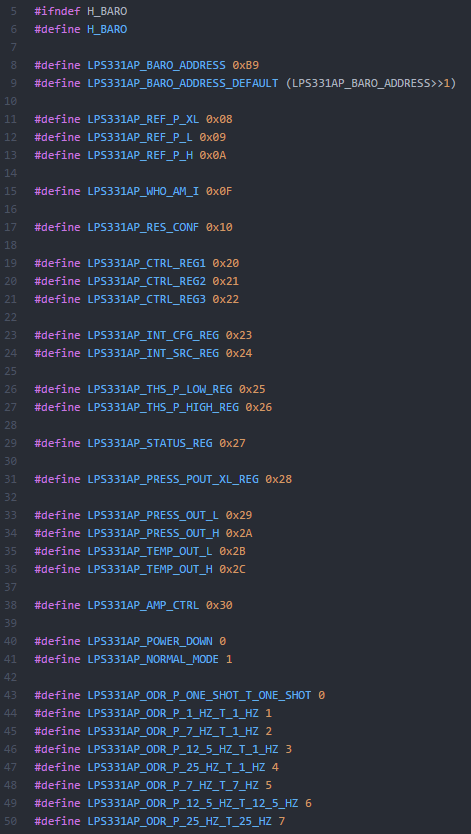


Figure - Extrait du fichier baro.h

Le fichier baro.c contient toutes les fonctions permettant de lire et d’accéder aux différents registres du capteur baromètre de la centrale inertielle.

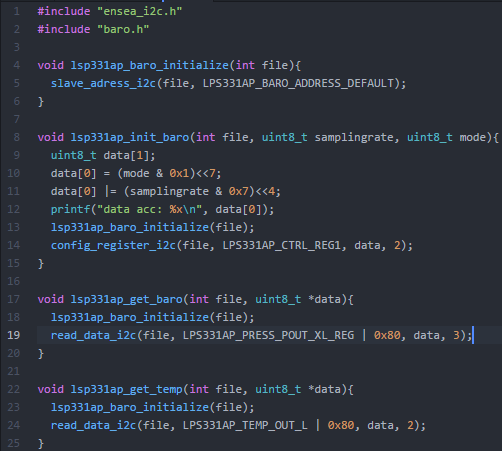


Figure - Fichier baro.c contenant les fonctions d'accès aux registres du capteur baromètre LPS331AP

### Fichier main.c

Le fichier main.c est le fichier principal de l’application, c’est par lui que se lance l’application.

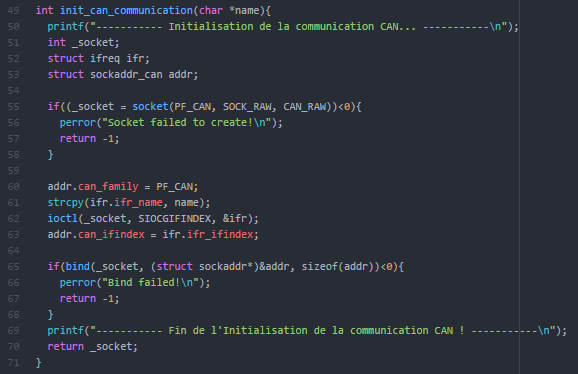


Figure - Fonction d'initialisation du Bus CAN

Le fichier main.c est disponible à l’annexe 1.

## Le programme Labview

Le programme LabView nous permet de demander à la carte i.MX6Q SABRELite les informations des différents capteurs de la centrale inertielle et ensuite de les afficher dans l’interface LabView.

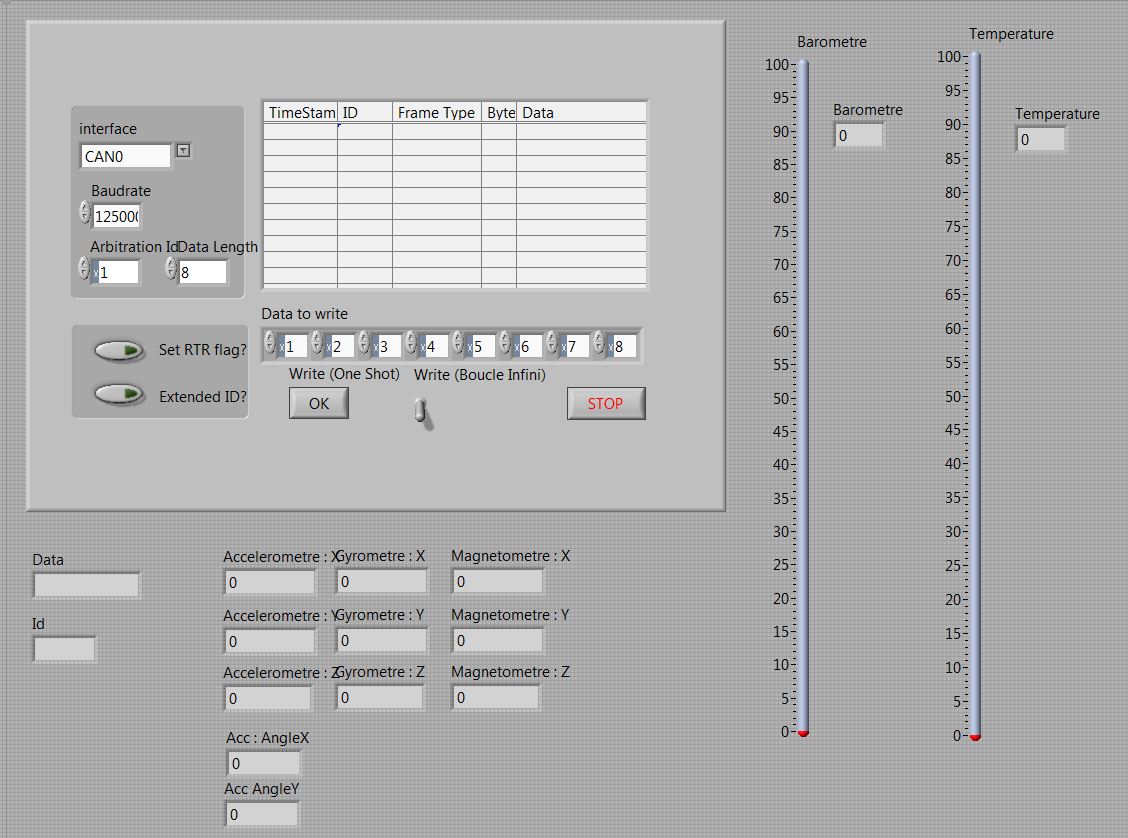


Figure - IHM de l'application LabView

CONCLUSION

Ce projet fut très intéressant pour comprendre le fonctionnement des protocoles de communication i2c et CAN, et nous as permis d’apprendre énormément sur la programmation dans un environnement Linux et LabView.

Nous tirons de cette expérience une grande richesse tant en terme technique qu’en terme de gestion de projet.

Annexes

**Annexe 1 : Code du fichier main.c**

