Développé par [redacted],
[redacted] STI2D Option SIN.
Lycée Vauban Brest

Projet Fusée - BAC

Partie : Informatique

Cette partie informatique est composée de plusieurs parties qui seront expliquées et détaillées.
Résumé : Cette partie consistera en la conception d'un programme en **C++**, qui enregistrera et convertira toutes les données du vol dans un périphérique de stockage au format .csv (format de tableau partiel), et la création du câblage des différents composants à la carte ESP8266 que nous allons utiliser comme boîte noire lors du vol de la fusée. Les données du vol pourront être ensuite exporté dans un logiciel de tableur de haut niveau tel que Libre Office ou Microsoft Excel.

Sommaire

Liste des composants :

- ESP8266
- Gyroscope 3 axes/Accéléromètre GY-521 MPU-6050
- 2 Baromètres GY-BMP280
- Micro SD/SDHC Card SPI
- Active Buzzer

Pourquoi cette carte ?

Au départ la carte Arduino Uno devait être utilisé car très peu cher et notre établissement en fourni plein. Mais au fur et à mesure de la conception du programme il s'est avéré que la carte n'était pas suffisamment puissante pour effectuer les nombreux calculs très rapidement et qu'elle ne possède pas assez de mémoire vive pour faire tourner le programme correctement. Nous avons donc opté pour l'utilisation de la carte ESP8266 qui a, d'après ce site qui en a fait la comparaison, 64x plus de mémoire vive, qui est 5x à 10x plus rapide pour effectuer des calculs, et qui est 3x plus petite.

De plus le lycée en possède également pour l'emprunt, ce qui nous facilitera la tache.

Description du Programme :

Le programme doit être capable de récupérer les informations des différents capteurs et d'écrire les données dans une carte Micro SD tout en restant rapide et ergonomique.

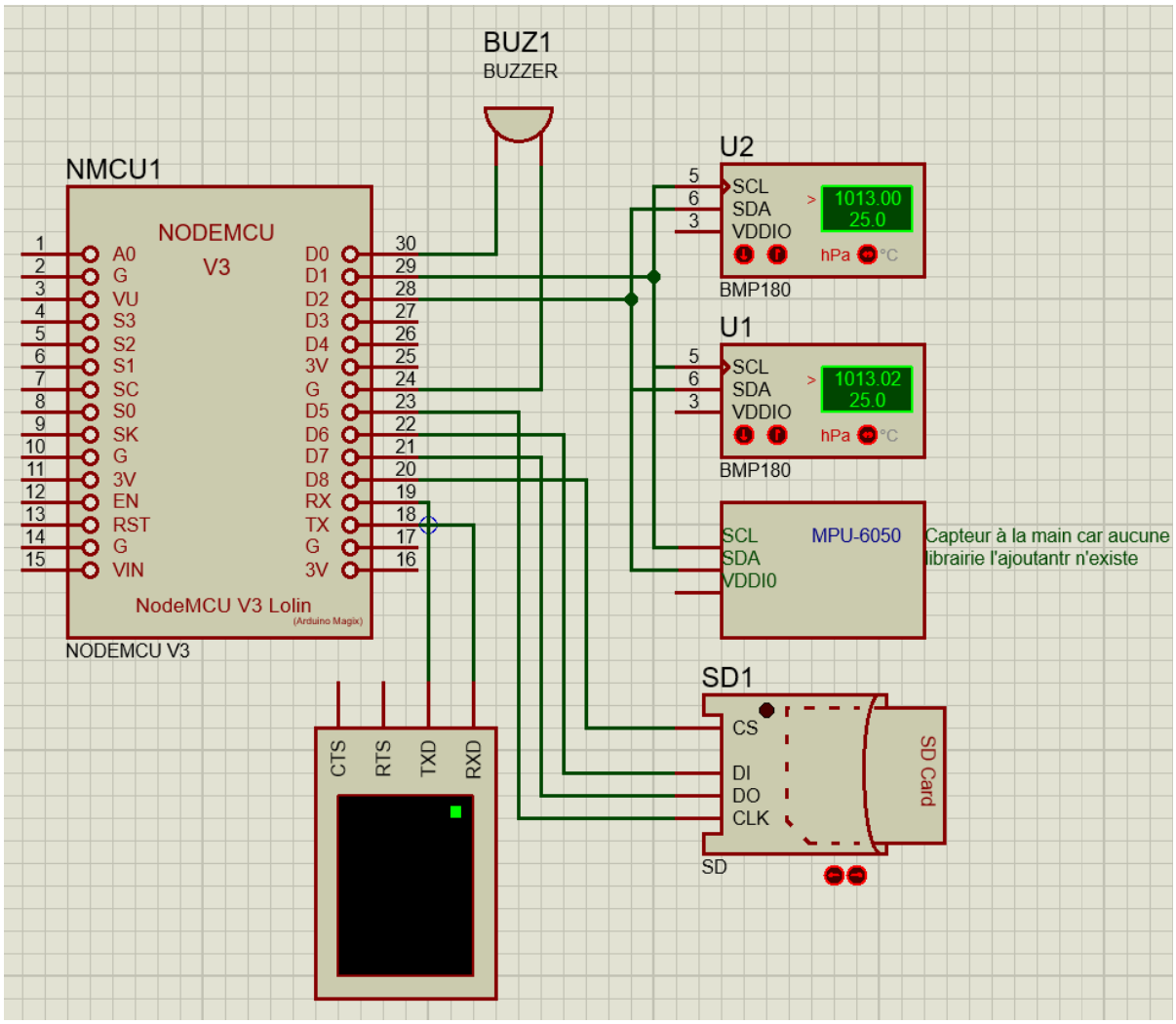
La fréquence de l'opération, imposée par le maître de projet, doit être d'environ 100 ms, et l'écriture sur la carde SD doit être le plus rapide possible, un format de tableau partiel sera donc utilisé pour minimiser l'écriture.

Les fichiers créés par le programme seront dotés de l'extension .csv pour « Comma-Separated Values » qui est un format type tableur dont les valeurs sont séparées par des ‘,’ ou des ‘;’. Ce type de fichier est très utilisé pour listes de client, de fournisseurs, de relevés bancaires, et bien d'autres et est importable par tout logiciel de tableur.

Un buzzer sera aussi utilisé pour fournir un signal auditif à l'utilisateur un bon fonctionnement ou une erreur du programme lorsque le port série, pour le débogage, n'est pas branché. Pendant l'initialisation, 50 tests des différents capteurs seront effectués, pour les calibrer et ainsi diminuer la marge d'erreur, puis écrit sur la mémoire externe, qui servira ensuite de base à l'exploitation.

Câblage électrique :

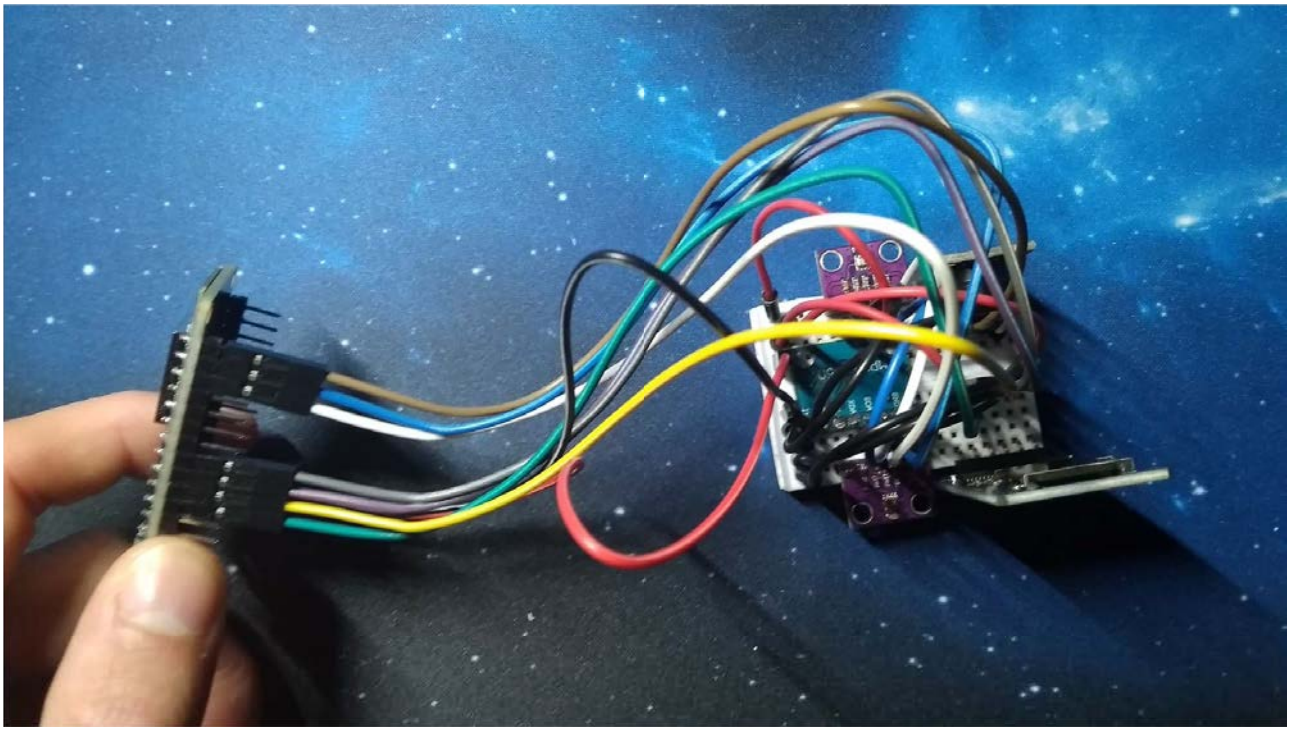
Le câblage des différents composants doit être le plus optimisé possible en prenant le moins de place possible. Les différents capteurs seront branchés selon le protocole I2C qui est pratique et facile à mettre en place pour la communication entre plusieurs capteurs en utilisant le moins de fil possible, cependant pour le module de carte SD seulement le protocole SPI est accepté, il sera donc câblé selon ce protocole. Et pour répondre à la question : Pourquoi ne pas tout brancher en SPI ?, et bien j'expliquerai que j'aurais voulu faire comme ça mais le Gyroscope/Accéléromètre n'accepte que le I2C. J'ai donc mis tous les capteurs en I2C.



Schémas du câblage :

Voici un image du schéma du câblage des différents composents réalisé avec Proteus 8 Professional.

Et voici une image du câblage réel, effectué sur platine d’essais.



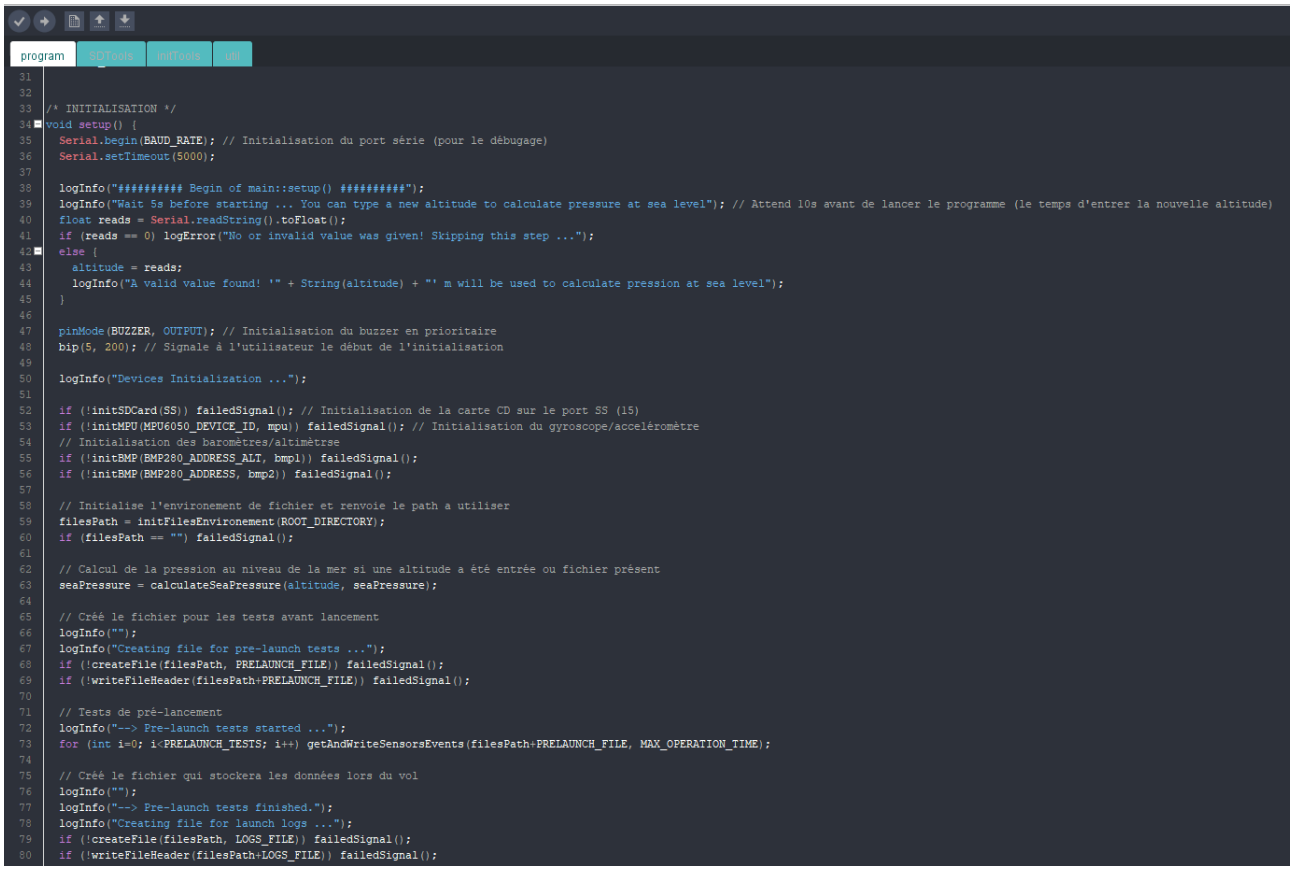
Conception du programme :

Le programme sera écrit en C++, car c’est un des langages comprit par le compilateur et aussi car je le maitrise, et sera écrit de manière optimisée et divisée en fonctions en essayant de prendre le moins de ligne de code possible pour qu’il soit le plus lisible possible.

Les fonctions sont réparties dans plusieurs fichiers pour plus de lisibilité et sont documentées.

PS : Ma syntaxe personnelle sera utilisée pour écrire le programme.

Aperçu du programme :



Disponible [sur GitHub](#).