**Mise en place du capteur MPU9250**

La capteur MPU9250 est un capteur de suivi de mouvement 9 axes. Il est composé d’un gyroscope 3 axes, d’un accéléromètre 3 axes, d’un magnétomètre 3 axes, et d’un capteur de température. Le magnétomètre sera utilisé pour l’expérience sur les résonnances de Schumann. L’accéléromètre et le gyroscope sera utilisé pour la localisation de la fusée.

# Initialisation :

Dans un premier temps, il faudra développer une fonction d’initialisation du capteur pour les paramètres suivants :

|  |  |
| --- | --- |
| Paramètre | Valeur |
| Communication | I2C |
| Gyroscope angular rate | +-500 °/s |
| Gyroscope low-pass filter | A déterminer |
| Gyroscope offsets (X, Y, Z) | A déterminer lors de la calibration |
| Accelerometer scale range | +-16g |
| Accelerometer offsets (X, Y, Z) | A déterminer lors de la calibration |
| Magnetometer offsets (X, Y, Z) | A déterminer lors de la calibration |
| Interruption | Sensor registers |

# Acquisition :

Plusieurs solutions sont possibles pour l’acquisition des valeurs. On peut directement lire les données dans le **Sensor Data Register** qui contient les dernières valeurs des capteurs, ou alors utiliser la pile **FIFO** (First In First Out). Le registre FIFO doit être configuré auparavant. Cette méthode permet de récupérer les données par burst. Cependant, les données du magnétomètre ne peuvent pas être acquise via cette méthode. Le Sensor Data Register sera alors privilégié. Une interruption sera envoyée a chaque fois qu’une acquisition sera prête.

L’acquisition des capteurs ne doit commencer qu’après 100ms.

L’adresse I2C devrait être 0x68 ou 0x69 (0x = hexadécimal)

Il faudra donc récupérer les valeurs de l’accéléromètre, gyro, magnéto et capteur de température à chaque interruption. Et vérifier que les valeurs sont correctement calibrées.

# Traitement :

Un algorithme de fusion de données devra permettre de détermine la position de la fusée dans l’espace en fonction de l’angle et de l’intégration de l’accélération. Une fusion sera ensuite proposé avec le GPS et l’altimètre afin d’affiner les valeurs.

**Références :**

Datasheet : <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132035/TDK/MPU-9250.html>

Arduino library : <https://github.com/hideakitai/MPU9250>