

Première Partie : Programmation et communication I2C

I. Installation et prise en main du logiciel

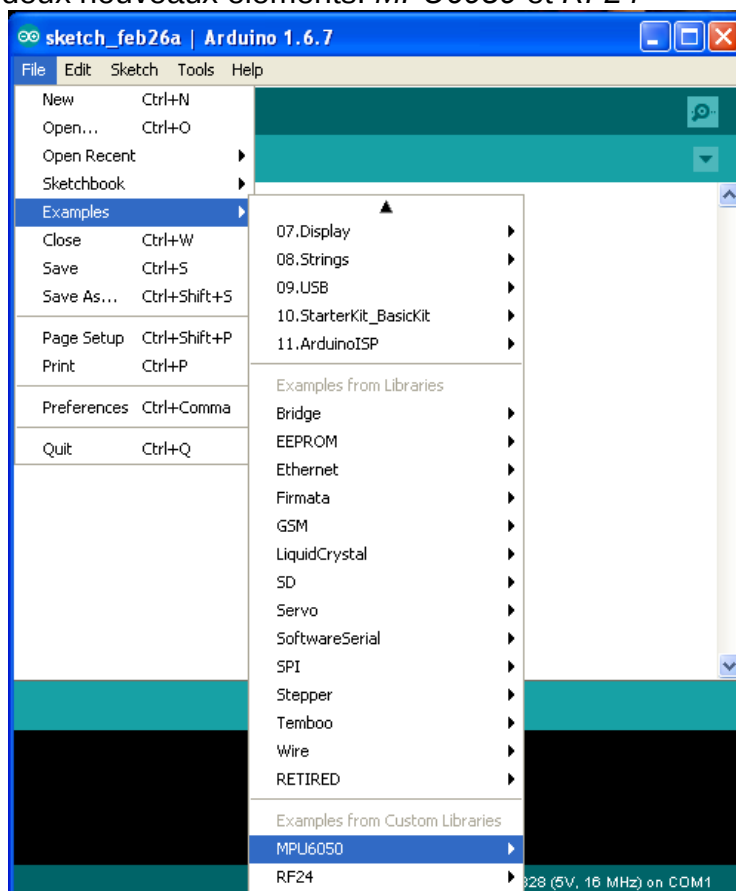
Pour pouvoir programmer le micro-contrôleur (Atmega328) qui est sur la carte IMU, on utilise l'IDE (environnement de développement intégré) et les bibliothèques Arduino, ainsi que deux autres bibliothèques open-source pour pouvoir communiquer avec l'accéléromètre/gyroscope (I2C) et le transmetteur sans fil (SPI).

On va commencer par télécharger et installer (si ce n'est pas encore fait) l'Arduino IDE :

- Aller sur <http://arduino.cc/> et cliquer sur l'onglet *Download* puis sur *Windows Installer*, puis sur *Just download*.
- Installer puis lancer le logiciel, confirmer toutes les boîtes de dialogue qui s'ouvrent
- Une fois lancé, fermer l'Arduino IDE, un dossier *Arduino* devrait avoir apparu dans le dossier *Mes Documents*

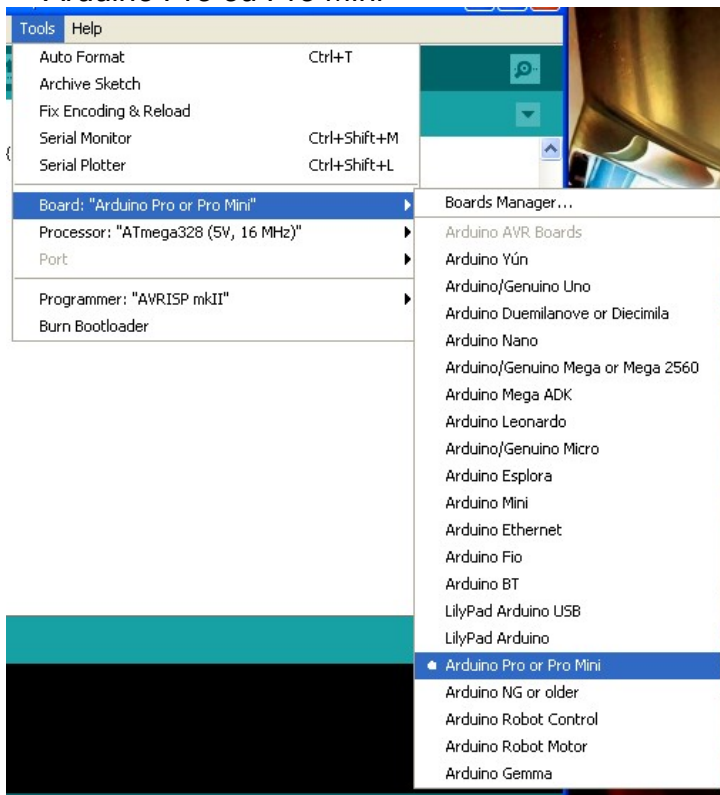
On va ensuite installer les deux bibliothèques nécessaires :

- Copier le contenu du dossier *bibliothèques* fourni avec le TP dans le dossier *libraries* qui est dans le dossier *Arduino*.
- Lancer l'Arduino IDE, regarder dans l'onglet *Fichier* → *Exemples*, il devrait y avoir deux nouveaux éléments: *MPU6050* et *RF24*

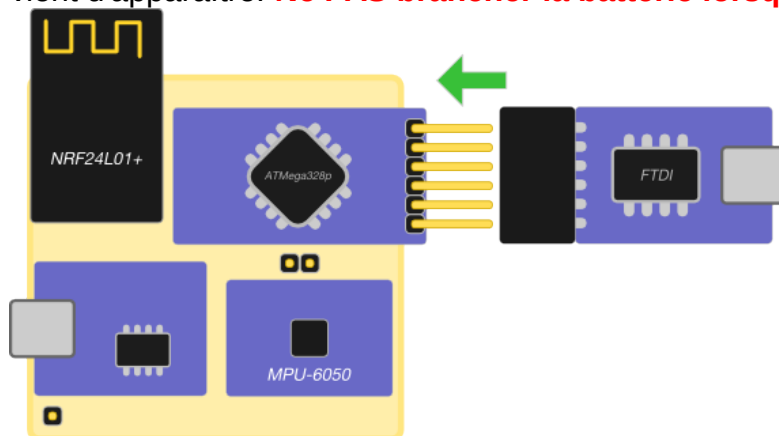


Une fois les bibliothèques installées, on programme le micro-contrôleur avec des instructions simples : allumer une LED 1 fois par seconde :

- Ouvrir *Fichier* → *Exemples* → *Basiques* → *Blink*
- Sélectionner la bonne configuration du micro-contrôleur en faisant *Outils* → *Carte* → *Arduino Pro ou Pro Mini*



- Noter les ports série présents sur le PC en regardant dans *Outils* → *Port*, puis brancher la carte IMU au PC (via la carte FTDI) et sélectionner le port série qui vient d'apparaître. **Ne PAS brancher la batterie lorsque la carte est reliée au PC**



- Flasher (ou programmer) le programme sur le micro-contrôleur en cliquant sur le bouton Téléverser (la flèche vers la droite)



- La puce de programmation devrait clignoter un moment et l'Arduino IDE devrait afficher Téléversement *Terminé* dans la console (le carré noir en bas)
- Le micro-contrôleur devrait allumer la LED pendant 1s puis l'éteindre pendant 1s en boucle
- Fermer le logiciel

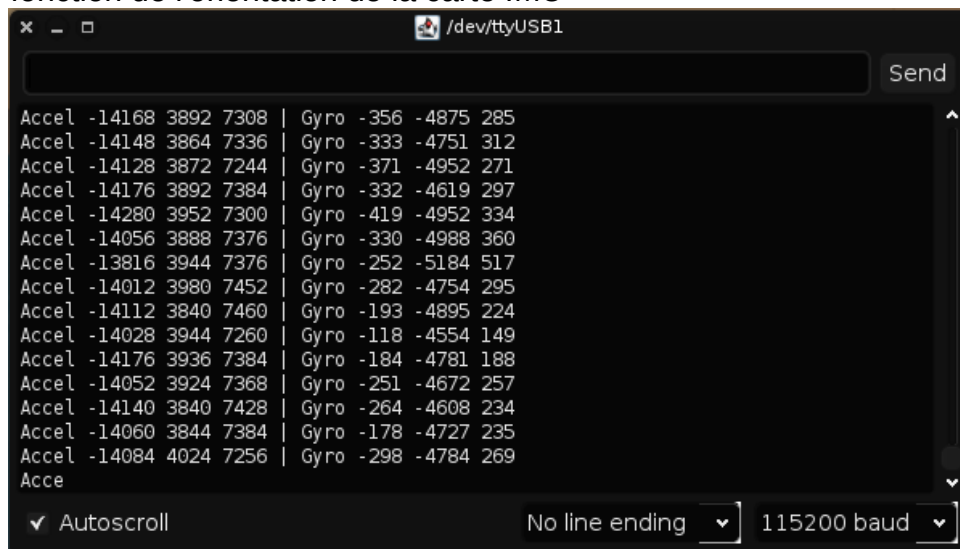
II. Adressage I2C

La puce utilisée est le *MPU-6050* de chez *Invensense*. Elle contient un accéléromètre 3 axes, un gyroscope 3 axes et une unité de calcul (DMP). Elle communique avec l'extérieur via une interface série I2C et peut avoir deux adresses au choix. Cette fonctionnalité est décrite à la page 33 de la datasheet (*MPU-6000.pdf*).

- Déterminer l'adresse en hexadécimal de la puce, sachant que la broche 9 (*AD0*) est connectée à la masse
- Copier le contenu du dossier *Code* fourni avec le TP dans le dossier *Arduino*
- Ouvrir Arduino IDE, puis ouvrir *imu_base* dans le menu *Fichier* → *Croquis*
- Changer l'adresse (*#define...*) à la valeur trouvée précédemment
- Cliquer sur le bouton *Téléverser* et attendre le message *Téléversement Terminé*
- Ouvrir la console série : (petite loupe a droite)



- Changer la vitesse à 115200 baud
- Les mesures de l'accéléromètre et du gyroscope devraient s'afficher et changer en fonction de l'orientation de la carte IMU



III. Calcul des valeurs réelles

Les valeurs affichées sont des valeurs dites "Plaine Echelle", elle n'ont pas d'unité et sont juste le résultat de la conversion de la tension de sortie des capteurs. Elles sont sur 15 bits (valeur max = 32768).

La datasheet du composant indique la valeur du bit de poids faible (*LSB*) : c'est la plus petite variation possible.

- Utiliser la datasheet (pages 12 et 13) pour compléter les fonctions *calcAngle* et *calcAccel* (attention, en anglais la virgule est juste un séparateur pour mieux lire les grands nombres, le séparateur décimal est le point)
- Une fois les fonctions complétées, activer ces fonctions en dé-commentant les lignes 30 et 31 du programme

```
// Calcul de l'acceleration lineaire et angulaire en fonction de
// calcAccel(&ax, &ay, &az, imu.getFullScaleAccelRange());
// calcAngle(&rx, &ry, &rz, imu.getFullScaleGyroRange());
```

- Les valeurs de l'accéléromètre devraient être d'environ 1g pour l'axe Z et -1g lorsqu'on met à l'envers la carte IMU
- Les valeurs du gyroscope devraient atteindre à peu près les ± 250 degrés/s lorsqu'on fait tourner rapidement la carte IMU
- Si les valeurs ne semblent pas correspondre, relire les commentaires des fonctions `calcAccel` et `calcAngle` et vérifier les unités

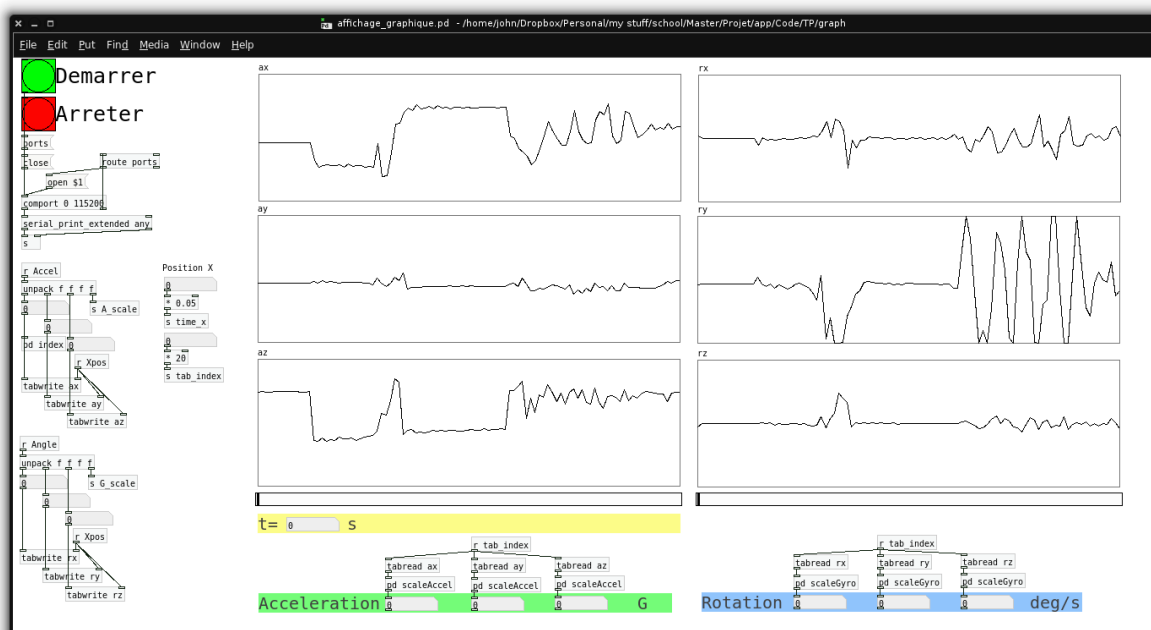
IV. Transmission sans fil et Interface graphique

Avant d'utiliser l'affichage graphique des valeurs, on doit activer le transmetteur 2.4GHz intégré (nRF24L01+ de Nordic) et programmer la carte de réception :

- Ouvrir le code `imu_TX` : *Fichier* → *Croquis* → `imu_TX`
- Bien veiller à débrancher la batterie de la carte IMU – TX
- Modifier les adresses d'émission/réception (ligne 17) `1Node` et `2Node` par 5 lettres/chiffres au hasard (pour éviter d'envoyer des données chez le voisin)
- Flasher le code sur la carte IMU – TX
- Ouvrir le code `imu_RX` : *Fichier* → *Croquis* → `imu_RX`
- Débrancher la carte IMU – TX de la puce FTDI et brancher la carte IMU – RX
- Modifier les adresses (ligne 5) et mettre les mêmes que celles du code `imu_TX`
- Flasher le code sur la carte

Nous allons maintenant installer le logiciel de programmation/visualisation graphique (si ce n'est pas déjà fait) : *PureData*

- Télécharger le logiciel : <http://sourceforge.net/projects/pure-data/files/pd-extended/0.43.4/Pd-0.43.4-extended-windowsxp-i386.exe/download>
- Installer le logiciel, confirmer toutes les boîtes de dialogue
- Ouvrir le fichier `affichage_graphique.pd` dans le dossier `graph`



L'interface/code est séparée en 3 parties :

- A gauche : réception et le décodage des trames série envoyées par la carte RX
- Au milieu l'affichage graphique : à gauche l'accélération X, Y et Z et à droite la vitesse de rotation
- En bas des graphes : les curseurs, le temps t et les valeurs précises au temps t

Utilisation :

- Brancher la carte IMU – RX
- Brancher la batterie de la carte IMU – TX
- Cliquer sur *Démarrer* (Bouton vert)
- Les graphes devraient commencer à enregistrer les mouvements de la carte TX
- Faire tourner rapidement la carte TX sur l'axe X (mouvement du poignet analogue à utiliser un tournevis) : les axes sont indiqués sur la carte *MPU-6050*
- Cliquer sur *Arrêter*
- Utiliser les curseurs pour vérifier que la vitesse de rotation max. enregistrée sur l'axe X est bien de 250°/s et sature facilement

Réduction de la sensibilité du gyroscope :

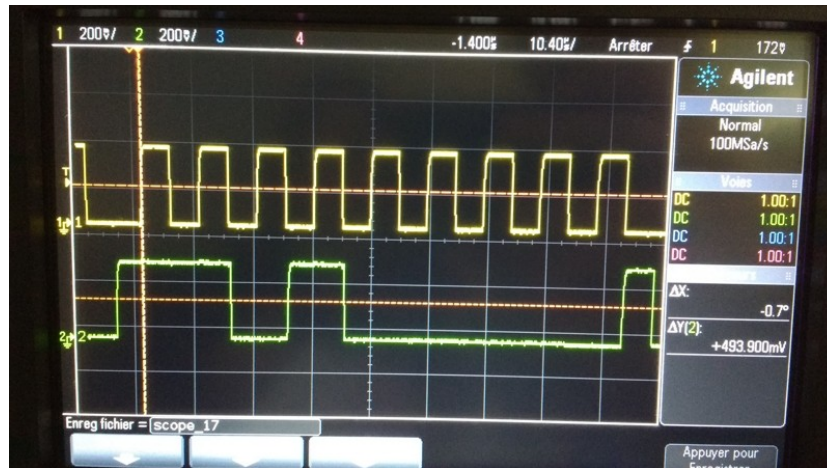
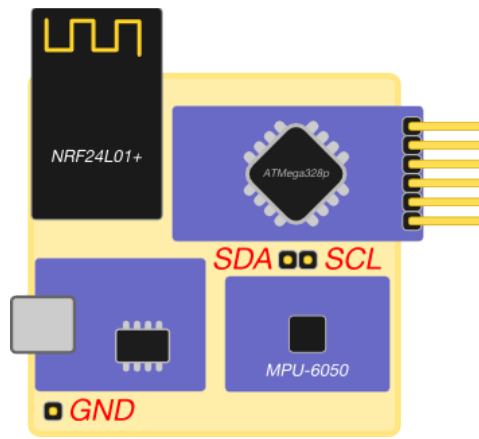
- Débrancher la carte RX de la puce FTDI
- Modifier la ligne 39 du programme *imu_TX* et mettre un 3 à la place du 0
- Débrancher la batterie de la carte TX
- Brancher la carte TX à la puce FTDI et flasher le code modifié
- Débrancher la carte TX et rebrancher la carte RX
- Brancher la batterie de la carte TX
- Cliquer sur *Démarrer* sur l'interface graphique
- Faire le même mouvement de poignet que précédemment avec la même force
- Cliquer sur *Arrêter*
- Mesurer la vitesse de rotation max. , consulter les pages 12/13 de la datasheet et conclure sur l'opération effectuée

V. Protocole de communication (I2C)

Le gyroscope/accéléromètre communique via un protocole série appelé I2C Bus : Inter-Integrated Circuit Bus. Comme son nom l'indique, il est principalement utilisé pour communiquer entre différents circuits intégrés sur une même carte.

Nous allons d'abord visualiser et décoder un début de trame I2C à l'oscilloscope, puis analyser un relevé fait à l'analyseur logique (décodage plus facile) :

- Brancher les 2 voies de l'oscilloscope aux signaux *SDA* (Données) et *SCL* (Horloge), puis brancher la masse au pin *GND* sur la carte IMU – TX
- Allumer l'oscilloscope, appuyer sur le bouton *Auto* ou *Autoscale* pour réinitialiser les paramètres
- Modifier les paramètres (Temps et Tension), puis utiliser le bouton *Single* pour effectuer des acquisitions individuelles jusqu'à avoir une trame qui ressemble à peu près à la photo



Recopier (ou imprimer) la trame sur feuille et identifier les différents éléments : voir les pages 33 à 35 de la datasheet *MPU-6000.pdf*

- Conditions *START* et *STOP*
- Réponse de l'esclave (MPU-6050) au maître (ATMega328) : *ACK*
- 7 bits d'adresse
- Bit *Read/Write*
- Décoder les 7 bits d'adresse, les passer en hexadécimal et comparer avec l'adresse calculée précédemment

Décodage des mesures inertielles :

Un analyseur logique est un appareil digital, il ne mesure pas la tension mais seulement l'état logique des lignes à mesurer. Il est doté d'une grande mémoire et est utilisé pour debugger les systèmes numériques.

Il est aussi fourni la plupart du temps avec des analyseurs de protocoles (I2C, SPI, CAN etc..) qui décodent automatiquement les trames enregistrées et fournissent un fichier texte/csv contenant les données décodées en hexadécimal.

Ici, nous avons placé la carte IMU – TX dans une certaine position et effectué un relevé à l'analyseur logique, il va falloir retrouver cette position en décodant/convertissant les données relevées :

- Ouvrir le fichier *TX_i2c_decode.ods* dans le dossier *capture* fourni avec le TP
- Recopier sur feuille la première trame (entre le premier *Read* et le 2^e *Write*)
- A l'aide des pages 29 à 31 de la datasheet *MPU-6000-Register-Map.pdf*, identifier les données appartenant à chaque capteur et axe : accélération (X, Y, Z),

température, et rotation (X, Y, Z), sachant que les adresses des registres s'incrémentent automatiquement

- Assembler les MSB et LSB de chaque axe de l'accéléromètre, et sachant que la sensibilité est de 2g, calculer la valeur réelle en g
- Dédire la position de la carte dans l'espace au moment de la mesure

Bonus :

- En utilisant la formule page 30, calculer la température au moment de l'acquisition
- Attention : la valeur binaire est en complément à 2

Pour ceux qui veulent voir à quoi ressemble une interface d'analyseur logique, télécharger et installer *Logic* de Saleae ici : <https://www.saleae.com/downloads>

Il suffit de déplacer le fichier *TX_i2c.logicdata* dans la fenêtre du logiciel pour ouvrir les trames enregistrées. (zoom avec la molette et déplacement avec clic gauche)