

# RAZOR IMU

## AHRS

INSTALLATION  
CALIBRATION

Alice Hiron

## A. INSTALLATION LOGICIELLE ET MATERIELLE

- Télécharger et dé-zipper les fichiers sources depuis le dépôt GitHub à l'adresse suivante :  
<https://github.com/Razor-AHRS/razor-9dof-ahrs/archive/refs/heads/master.zip>
- Télécharger et installer **IDE Arduino** :  
<https://www.arduino.cc/en/software>
- Télécharger et installer **Processing** :  
<https://processing.org/download/>
- Relier le module headtracker via le câble USB associé et brancher le à votre ordinateur.

## B. PARAMETRAGE

- Lancer l'interface de l'IDE Arduino
- Aller dans le menu « Outils », puis « Type de carte » et sélectionnez « Arduino Pro or Pro Mini »
- Vérifier que « Processeur » correspond bien à « ATmega328P » 3,3V et 8Mhz

Type de carte: "Arduino Pro or Pro Mini"	>
Processeur: "ATmega328P (3.3V, 8 MHz)"	>

- Sélectionner votre port série selon MAC ou Windows.

Par exemple, le module headtracker d'origine utilise l'*USB série ADO25JT3* sous Mac, et par exemple « *COM12* » sous Windows.

- Depuis votre dossier téléchargé, ouvrir le fichier « *Razor\_AHRS.ino* » :

*Arduino/Razor\_AHRS/Razor\_AHRS.ino*

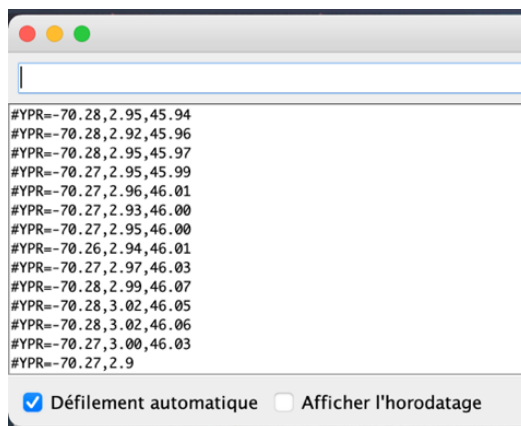
- Dans la partie « *HARDWARE OPTIONS* » du script, décommenter la bonne valeur de la variable globale « *HW VERSION CODE* » :  
Pour le module 9DOF Razor IMU contenant un magnétomètre HMC5883L la valeur est 10736.

```
// HARDWARE OPTIONS
//*****
// Select your hardware here by uncommenting one line!
// #define HW_VERSION_CODE 10125 // SparkFun "9DOF Razor IMU" version "SEN-10125" (HMC5843 magnetometer)
// #define HW_VERSION_CODE 10736 // SparkFun "9DOF Razor IMU" version "SEN-10736" (HMC5883L magnetometer)
// #define HW_VERSION_CODE 10183 // SparkFun "9DOF Sensor Stick" version "SEN-10183" (HMC5843 magnetometer)
// #define HW_VERSION_CODE 10321 // SparkFun "9DOF Sensor Stick" version "SEN-10321" (HMC5843 magnetometer)
// #define HW_VERSION_CODE 10724 // SparkFun "9DOF Sensor Stick" version "SEN-10724" (HMC5883L magnetometer)
```

- Téléverser le fichier en cliquant sur la flèche située en haut à gauche



- Vous pouvez tester le module headtracker en vérifiant la bonne réception des données ; aller dans le menu « *Outils* » puis « *Moniteur de Série* ».
- Sélectionner la vitesse « baud rate » à 57600. Le résultat visible devrait correspondre à l'image ci-dessous :



## C. CALIBRATION

Patientez quelques minutes afin que celui-ci se fasse dans des conditions de chaleur habituelles pour les capteurs.

### Calibrage de l'accéléromètre

Lors des manipulations suivantes, veiller à bouger le headtracker très lentement afin que l'accélération appliquée soit la plus petite possible.

Ce qui nous intéresse, ce sont les valeurs minimales et maximales des 3 axes : x, y et z.

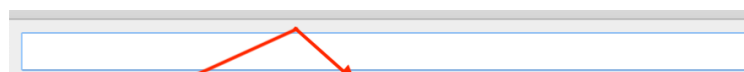
- Toujours depuis le « *Moniteur de Série* », rentrer la commande suivante : `#oc` suivi de la touche entrée.

Les données devraient correspondre à celles-ci :

accel x,y,z (min/max) = -5.00/-1.00 25.00/29.00 225.00/232.00

- Calibration de l'axe X :

Pointer délicatement le gyroscope externe vers le haut (connecteur vers le haut) et observer la valeur maximale augmenter.



accel x,y,z (min,max)	= 236.00/256.00	35.00/55.00	-81.00/-62.00
accel x,y,z (min,max)	= 236.00/256.00	35.00/55.00	-81.00/-62.00
accel x,y,z (min,max)	= 236.00/256.00	35.00/55.00	-81.00/-62.00
accel x,y,z (min,max)	= 236.00/256.00	35.00/55.00	-81.00/-62.00

- Tenez le boîtier très immobile et réinitialiser la mesure en envoyant à nouveau `#oc`
- Faites la même chose pour le côté opposé (connecteur vers le bas), et noter cette fois-ci la valeur minimale.
- Procéder de la même façon pour l'axe z (bas et haut) et pour l'axe y (droite et gauche).

- Depuis le fichier « *Arduino/Razor\_AHRS/Razor\_AHRS.ino* », repérer la ligne "USER SETUP AREA" puis "SENSOR CALIBRATION" et saisir les valeurs obtenues.

### Calibrage du gyroscope

- Pour passer à la calibration du gyroscope, saisir deux fois *#on*  
Vous devriez visualiser les informations suivantes :

gyro x,y,z (current/average) = -29.00/-27.98 102.00/100.51 -5.00/-5.85

- Poser le headtracker à plat sans le déplacer puis patienter 10 secondes le temps que le gyroscope collecte et fait la moyenne du bruit obtenue sur les trois axes.
- Noter uniquement les secondes valeurs de chaque paire :

gyro x,y,z (current/average) = -29.00/(-27.98) 102.00/100.51 -5.00/(-5.85)

- N'hésitez pas à réinitialiser les mesures en tapant *#oc*

### Calibrage du magnétomètre

Il est important de calibrer le module headtracker dans les conditions réelles (head tracker installé sur le casque, fils reliés ...) afin de prendre en compte toutes les facteurs externes liés aux champs magnétiques.

- Quitter l'*IDE Arduino* (afin de libérer le port série).
- Ouvrir le fichier « *Magnetometer calibration.pde* » depuis « *Processing/Magnetometer calibration/Magnetometer calibration.pde* »
- Suivez les instructions situées en début de fichier concernant l'installation de la librairie *EJML*.  
Ou décompresser *EJML.zip* dans le dossier où est installé votre application *Processing/libraries*

- Exécuter le script via l'interface en cliquant sur le bouton suivant

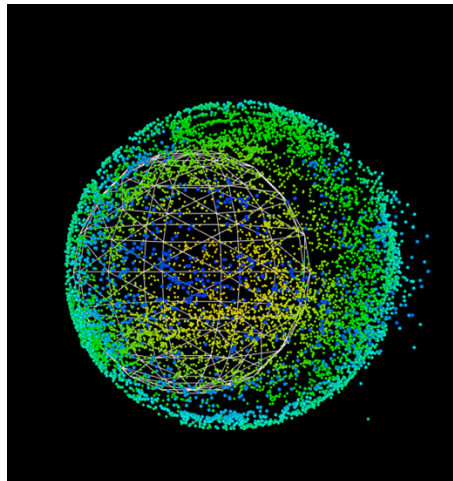


Si vous rencontrez une erreur, consultez les informations de la console, située en bas de l'interface. Il s'agit en général du mauvais port série. N'hésitez pas à le modifier puis à relancer le script.

Par défaut, sa valeur vaut 0. Sous Mac, il peut-être à 1 :

```
final static int SERIAL_PORT_NUM = 1;
```

- Bouger le casque de manière à parcourir toutes les orientations possibles pour recouvrir uniformément de points la sphère apparaissant à l'écran.



- Appuyer sur la touche « Espace » et recopier les lignes de code apparaissant dans la console de *Processing*.
- Fermer *Processing* et ouvrir à nouveau le fichier « Arduino/Razor AHRS/Razor AHRS.ino » depuis l'IDE *Arduino*.
- Revenir sur la section « *USER SETUP AREA* » / « *SENSOR CALIBRATION* » / « *Magnetometer (extended calibration mode)* » du script, et coller les valeurs précédentes, issues de *Processing*. Vérifier que ces valeurs sont de type floatant noté avec un **point** et non virgule (ce qui provoquerai des erreurs de syntaxe sur l'application Arduino).

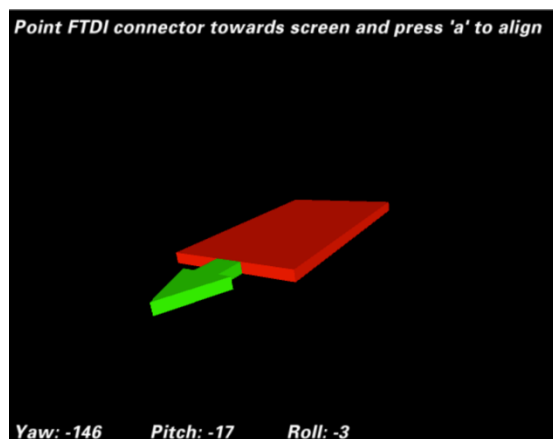
Exemple :

```
#define CALIBRATION__MAGN_USE_EXTENDED true
const float magn_ellipsoid_center[3] = {121.429, -17.1218, -17.1479};
const float magn_ellipsoid_transform[3][3] = {{0.912026, 0.00883780, -
0.00178482},{0.00883780, 0.914674, 0.0173662}, {-0.00178482, 0.0173662,
0.996465}};
```

- Téléversez à nouveau le fichier. Le headtracker est prêt à l'emploi 😊

## APERÇU SOUS PROCESSING

- Ouvrir le fichier depuis *Processing/Razor\_AHRS\_test/Razor\_AHRS\_test.pde*.
- Si vous rencontrez l'erreur suivante : « *The field PConstants.OPENGL is deprecated* » : allez dans la partie « *Setup serial port I/O* » et remplacer la ligne `println(Serial.List());` par `println((Object[])Serial.List());`
- Vérifier que vous utilisez le bon port (par défaut à 0) :  
`final static int SERIAL_PORT_NUM = 1;`
- Lancer l'application, appuyez sur la touche « a » pour initialiser l'axe de lacet (Yaw), puis tester votre headtracker.



## ANNEXE

Source :

<https://github.com/Razor-AHRS/razor-9dof-ahrs/wiki/Tutorial>

Vidéo :

<https://www.youtube.com/watch?v=n7l41bs94To&t=188s>