**Notice d’utilisation, d’acquisition et de visualisation**

**de données avec le Sensitive Pen**

**Pierre Libault, Ana Phelippeau, Adrien Husson, Joël Chevrier**

**CRI Paris, Motion Lab**

**Sommaire :**

1 - Introduction

2 - Le mode offline :

2.1 - Explication des programmes

2.2 - Déroulement du mode offline

3 - Le mode online :

3.1 - Explication des programmes

3.2 - Déroulement du mode offline

*Travail effectué dans le cadre de la thèse d’Ana Phelippeau*

*sous la direction de Joël Chevrier*

**1 - Introduction**

Le SensitivePen est un stylo développé dans le cadre de la thèse d’Ana Phelipeau, le but de ce dernier est de caractériser l’écriture via différents capteurs. En effet, il est équipé :

* d’un [movuino](http://www.movuino.com/) : IMU programmable développé au CRI, il nous permet d’avoir accès à l’accélération, la vitesse angulaire (gyroscope) et le champ magnétique qu’il détecte (magnétomètre)
* d’une nappe de pression : C’est un capteur qui nous permet d’avoir la pression des doigts
* d’un capteur de pression (de la mine): Il nous permet d’avoir le pression de la mine sur la feuille

Pour l’instant le SensitivePen n’est équipé que du movuino.

L’acquisition des données du movuino du SensitivePen peut se faire de deux manières différentes :

* - Mode offline : permet d’enregistrer le données dans le movuino et d’ensuite les transférer via une connection filaire sur un ordinateur
* - Mode live/online : permet de visualiser en direct les données du movuino sur l’ordinateur

Pour ces deux modes de fonctionnement, il est nécessaire d’avoir Arduino (<https://www.arduino.cc/en/software>) pour les firmwares, Unity (<https://unity3d.com/fr/get-unity/download>) pour la visualisation, un IDE Python pour le téléchargement des données (Pycharm par exemple), Max msp (<https://cycling74.com/downloads>) qui permet de faire une connection avec Unity pour le mode online, et last but not least il vous faudra télécharger le dossier du projet sur Git via GitHub Desktop (<https://desktop.github.com/>) en clonant le dossier (repository) ou directement sur internet.

**2 - Le mode Offline :**

Dans ce mode les données de l’acquisition sont stockées dans le movuino durant le test et seront extraites ensuite via un programme python, il pourra aussi les traiter. On pourra ensuite charger le fichier dans le programme Unity pour visualiser le test.

**2.1 - Explication des programmes**

*Arduino :*

Le firmware a pour nom firmware\_movuino\_NoWifi, il suffit d’ouvrir le fichier qui se situe dans le dossier Arduino avec Arduino et de le téléverser dans le movuino afin de pouvoir l’utiliser. Ce firmware permet au movuino de stocker les données et on pourra les extraire de ce dernier grâce à une connexion usb.

Pour le bon téléversement du firmware il est important d’avoir les librairies suivantes: *MPU9250 BolderFlight*, *YetAnotherButtonLibrary* (a compléter) que l’on peut télécharger dans le Manager de librairie/Gestionnaire de bibliothèques (*Croquis -> Inclure une bibliothèque -> Gérer les bibliothèques*). De plus afin que la carte du movuino soit disponible pour que le fichier se téléverse, vous devez ajouter l’[Addon ESP8266](https://learn.sparkfun.com/tutorials/esp8266-thing-hookup-guide/installing-the-esp8266-arduino-addon) avec ce lien dans l’URL de gestionnaire de carte supplémentaire <http://arduino.esp8266.com/stable/package_esp8266com_index.json> (*Fichier->Préférence*) .Suite à cela vou pourrez télécharger la bibliothèque *esp8266* de *ESP8266 community (version 2.3.0)* via le gestionnaire de carte (outils -> Type de carte -> Gestionnaire de carte)

Sélectionnez le bon port et le firmware pourra être téléversé. (Outils -> Port).

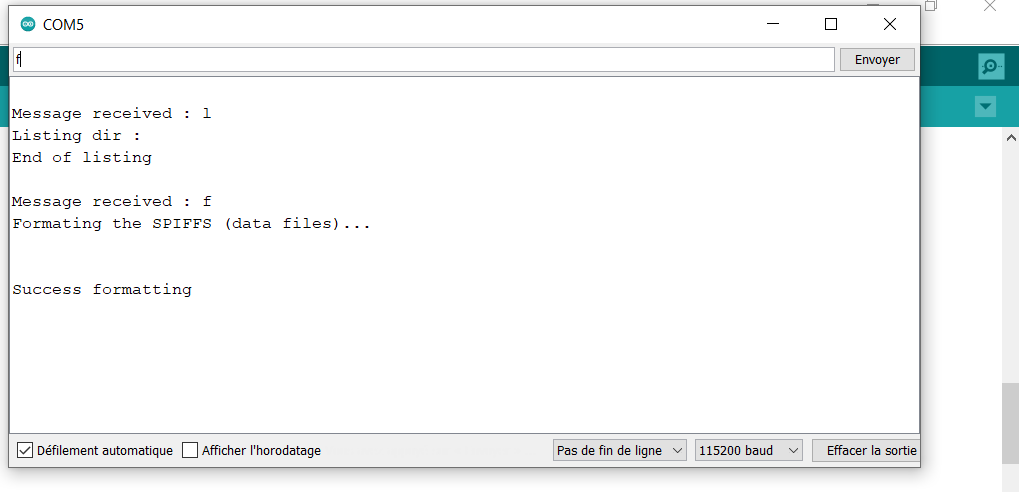
Fonctionnement du firmware :

Une fois le movuino allumé (dès que vou appuyez sur le bouton), il faut calibrer son magnétomètre en remuant lentement le movuino dans sa zone “d’action” (la feuille pour le sensitivePen) jusqu’à ce que la led bleu s’allume, il est maintenant opérationnel. (Il faudra refaire cette manipulation à chaque fois que vous allumerez)

Pour démarrer un enregistrement il faut double cliquer , à ce moment la un clignotement rapide (3 fois) démarre, ce clignotement indique que l’enregistrement a démarré. Pour arrêter un enregistrement il faut de nouveau double cliquer sur le bouton du movuino, un clignotement plus lent apparaît il indique que l’enregistrement s’est arrêté. Il est possible d’enchaîner plusieurs enregistrements à la fois, le programme python les stockera dans des fichiers différents.

Pour transférer les données, il faut lancer le programme python avec les bons paramètres (expliqués ci-dessous) et ensuite appuyer longtemps sur le bouton du movuino jusqu’à un arrêt du clignotement très rapide.

Remarque :

* Vous pouvez effacer les données du movuino en ouvrant le moniteur série quand le movuino est branché, la commande “l” vous permets de voir la taille du fichier enregistré, et la commande “f” vous permet de l’effacer (ça peut prendre un peu de temps)

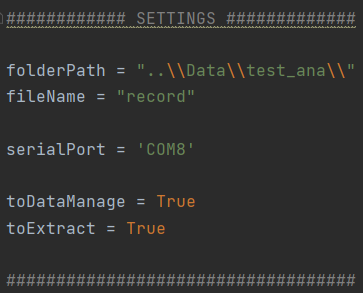
*Python :*

Les fichiers correspondant se situent dans le dossier “Python” du projet dans le dossier *MovuinoDataHandler,* dans ce dossier il n’y a normalement que des fichiers .py et .pyc. Il faut ouvrir un projet python avec l’IDE que vous avez choisi avec le fichiers de ce dossier. Il faudra surement configurer un environnement python, soit vous en avez déjà un (une base) avec anaconda par exemple soit avec un environnement virtuel (peu importe lequel normalement mais VirtualEnvironnement c’est bien), une fois que ceci sera fait, allez dans le terminal python et entrez cette commande qui téléchargera les librairies requises :

pip install pyserial matplotlib pandas scipy numpy

Fonctionnement du programme :

Comme dit plus haut, ce programme Python , sert à récupérer les données du movuino et à faire un premier traitement des données. Dans le fichier *main.py* qui correspond au programme principal il faut indiquer le device que vous utilisez (ici “sensitivePen”), le dossier et le nom du fichier dans lequel vous voulez transférer les données ainsi que le port série sur lequel les données vont être envoyées (vous pouvez le retrouver grâce à Arduino en ouvrant le moniteur série). Deux variables booléennes sont aussi utiles, il y a *ToExtract* que l’on met à True si on veut extraire les données du movuino et/ou *ToDataManage* qui sert au traitement et à l’affichage des données.

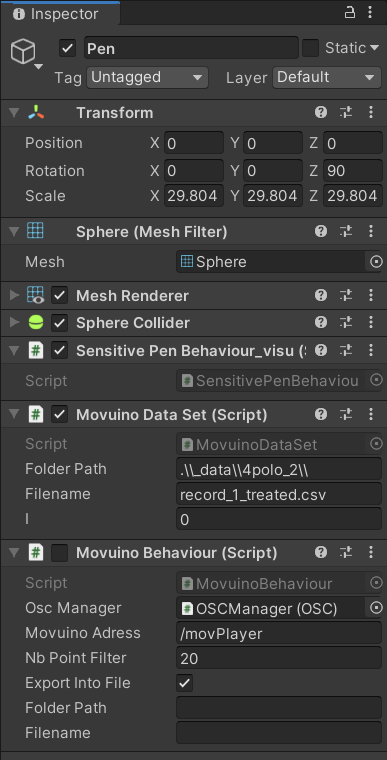


*Unity :*

La scène Unity qui va nous servir est la scène SensitivePen\_visu (Assets -> Scènes), normalement elle contient un objet OSCManager, UI et Pen (ainsi que des lumières etc.. mais elles n’influent en rien dans le fonctionnement du programme)

Il faut cliquer sur l’objet Pen, dans [l’inspecteur Unity](https://docs.unity3d.com/Manual/UsingTheInspector.html) du stylo vous devriez voir 3 scripts : SensitivePen\_visu, MonoBehaviour et MovuinoDataSet.

SI ce n’est pas le cas, il faut décocher le script MovuinoBehaviour qui correspond à l'utilisation Live, et il faut cocher le script MovuinDataSet. Remplissez les le nom du fichier à visualiser et appuyez sur PLAY (sinon Pierre se fâche).



**2.2 - Déroulement du mode offline :**

Déroulement type pour un bon fonctionnement du mode offline :

- Téléversement du programme firmware\_movuino\_NoWifi sur le movuino si ce n’est pas le cas

- Enregistrement des données

- Brancher le movuino sur le port série

- Ouvrir le programme Python et mettre les bons réglages, lancer le programme (avant le transfert des données !!!)

- Téléverser les données avec le movuino (en appuyant longtemps sur le bouton)

- Ouvrir Unity et lancer le programme de Unity en mode offline (cf rubrique Unity)

Dire ce que

**3 - Le mode Online :**

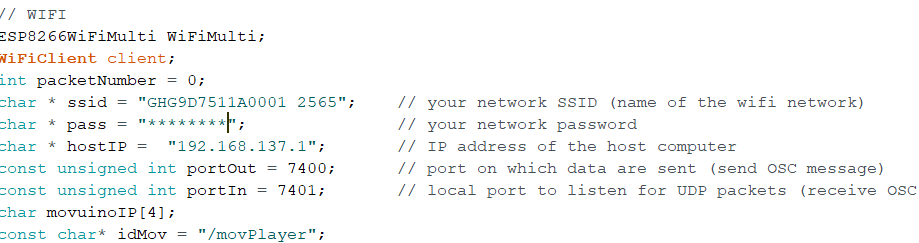
Dans ce mode les données de l'acquisition sont directement envoyées à l’ordinateur qui les enregistre et permet une visualisation live de ces dernières sur Unity, il est aussi possible de les extraire dans un fichier CSV à la fin de l’enregistrement.

**3.1 - Explication des programmes :**

*Arduino :*

Pour l'instant, le firmware que j’utilise s’appelle Movuino\_esp8266, il permet de se connecter en wifi avec l’ordinateur et envoie les messages via un protocole OSC.

Dans ce fichier il faudra juste écrire le nom du wifi sur lequel le movuino devra se connecter ainsi que le mot de passe.



Ensuite il suffit d’allumer le movuino pour qu’il se connecte, vous pourrez vérifier dans la patch max msp si le transfert des données se déroule bien.

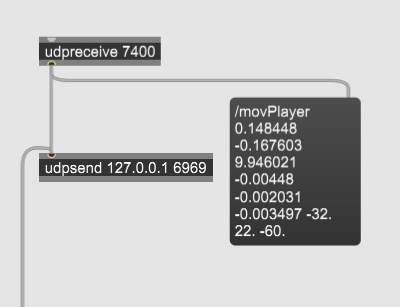
*Python :*

Le programme python va servir pour le traitement des données post-visualisation.

(A finir)

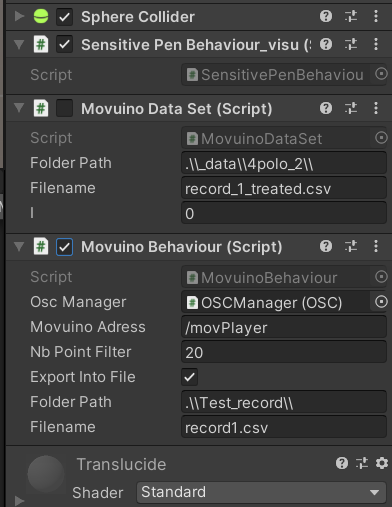
*Max msp :*

Le patch max msp sert uniquement à effectuer la connexion avec Unity.



*Unity :*

Si la connexion est bien effectuée, dans l’inspecteur Unity de l’objet Pen de la scène SesitivePen\_visu vous devez (si ce n’est pas déjà fait) cocher le script MovuinoBehaviour et décocher le script MovuinoDataSet, ensuite vous pouvez choisir si vous voulez enregistrer les données dans un fichier en indiquant le dossier et son nom.



**3.2 - Déroulement du mode online :**

Déroulement type pour un bon fonctionnement du mode online :

- Téléversement du programme firmware movuino\_esp8266 avec les bons settings wifi sur le movuino

- Ouvrez le Patch Max msp et connectez le movuino en wifi en l’allumant, si la connexion est bien effectuée vous devriez voir les données défiler

- Ouvrir Unity et lancer le programme de Unity en mode online (cf rubrique Unity)

Pour vider le Movuino : f

Pour voir s’il y a des données dedans : l