

Mesure physique de consommation d'énergie d'un smartphone dans l'outil GREENSPECTOR

Khalil Boulkenafet

1 Introduction

Ce document fait le bilan du travail réalisé sur la mesure hardware. On y présente le principe de la mesure, le travail d'intégration à l'outil GREENSPECTOR, les expériences menées, ainsi que le travail qu'il reste à effectuer.

2 Acquisition de la mesure

2.1 Dispositif

On utilise une carte ARDUINO UNO avec un shunt INA219. Le shunt, placé entre le smartphone et sa batterie, permet de mesurer l'intensité du courant qui sort de cette dernière. La carte ARDUINO permet de récupérer et traiter ces données de courant.

Le dispositif fonctionne sur SAMSUNG J3 et n'a pas encore été mis en place sur d'autres appareils.

2.2 Principe de la mesure

On fait l'hypothèse que le courant mesuré est constant sur l'intervalle de mesure considéré. On multiplie cette valeur de courant par l'intervalle de mesure (`M_PERIOD_ms`) pour obtenir une valeur en milliampère-heure (mAh).

2.3 Start/Stop de la mesure

L'acquisition des données est initiée en envoyant le caractère "a" à la carte ARDUINO, qui écrit sur le serial la consommation enregistrée toutes les `M_PERIOD_ms` (200 ms pour l'instant, cf fig. 1). On a observé que la carte ARDUINO nécessite un certain temps d'initialisation avant de pouvoir écrire des mesures. Ainsi on peut ne pas démarrer la mesure en envoyant "a" trop tôt. On arrête la mesure en envoyant "b" à la carte.

2.4 Code Arduino

Le code de la figure 1 a été téléversé sur ARDUINO. On utilise la librairie Adafruit_INA219. Le code et la librairie sont disponibles sur Gitlab.

```

#include <Wire.h>
#include <Adafruit_INA219.h>

Adafruit_INA219 ina219;

#if defined(ARDUINO_ARCH_SAMD)
// for Zero, output on USB Serial console, remove line below if using
// programming port to program the Zero!
#define Serial SerialUSB
#endif

#define M_PERIOD_ms 200
#define ARRAYSIZE 1000 // 5mn*1000*60/M_PERIOD_ms
int state = 0;

void setup(void)
{
  Serial.begin(115200);
  ina219.begin();
  ina219.setCalibration_32V_1A();
}

void loop(void)
{
  float current_mA = 0;
  float dischargemAh = 0;

  current_mA = ina219.getCurrent_mA();
  dischargemAh = current_mA*((float)M_PERIOD_ms/3600000.0);

  int incoming = 0;
  int serialavailable = 0;
  serialavailable = Serial.available();
  if (serialavailable > 0) {
    incoming = Serial.read();
    if (state == 0) {
      state = 1;
    } else if (state == 1) {
      state = 0;
    }
  }

  if (state == 1) {
    Serial.println(dischargemAh,8);
  }

  delay(M_PERIOD_ms);
}

```

Figure 1 – Code de la carte Arduino UNO

3 Le module hardware dans le Test Runner

3.1 Intégration et activation

Le module a été intégré dans les projets *go-testapi* et *testrunner*, dans le prolongement de la feature IOT. On active le module hardware via la configuration du TEST RUNNER : il suffit de renseigner le port de la carte ARDUINO (en général `/dev/ttyACM0` sur Linux) dans le champ `hardware.address` du fichier `config.yml` et `true` dans le champ `modules.hardware_probe` du fichier `job.yml`.

3.2 Nomenclature des métriques

Lorsque le module hardware est utilisé en mode IOT, la décharge mesurée se nomme `AH_PL`. En mode mobile, la nomenclature `AH_PL` est réservée à la sonde logicielle. On utilise `AH_HARD` pour nommer la métrique mesurée physiquement ; c'est cette métrique qui est présente dans le fichier de résultats et rendue visible sur l'interface GREENSPECTOR.

4 Mesures et analyse

4.1 Premières mesures

Après l'intégration du module hardware dans le TEST RUNNER, quelques mesures ont été effectuées afin de vérifier la récupération des résultats de mesure des deux sondes, à la fois dans un fichier de résultats bruts (en local, lorsque le mode online est désactivé) et sur l'interface GREENSPECTOR (fig. 2 & 3).

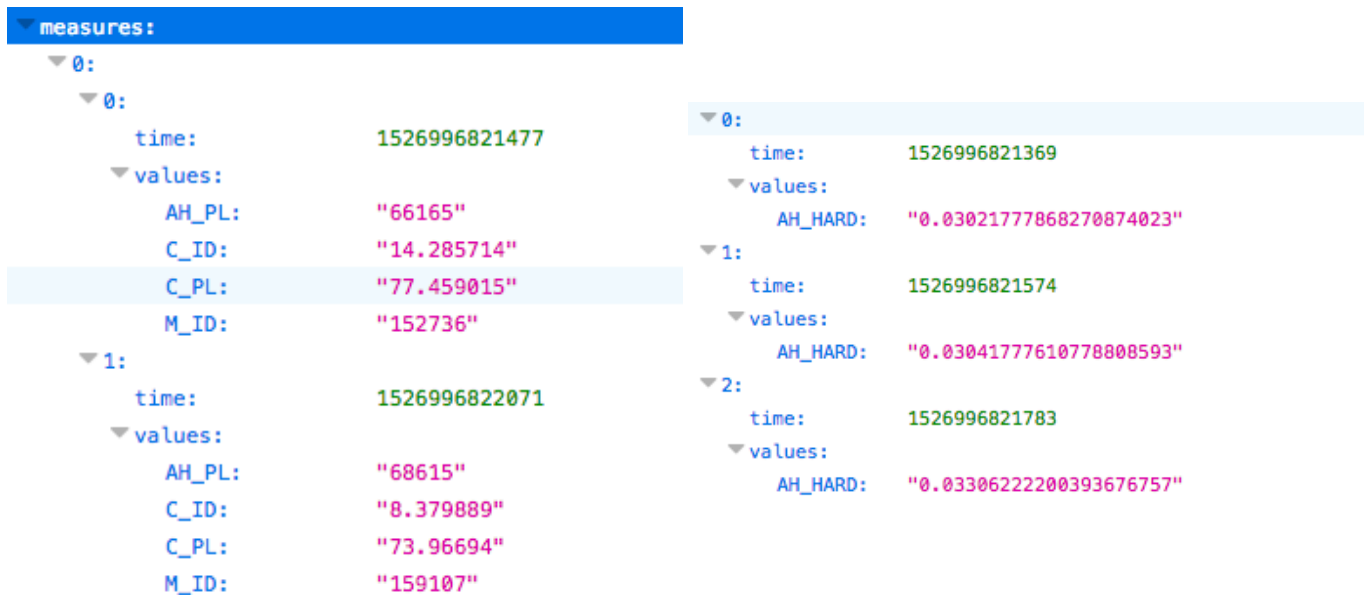


Figure 2 – Récupération des mesures des deux sondes en local dans un fichier json.

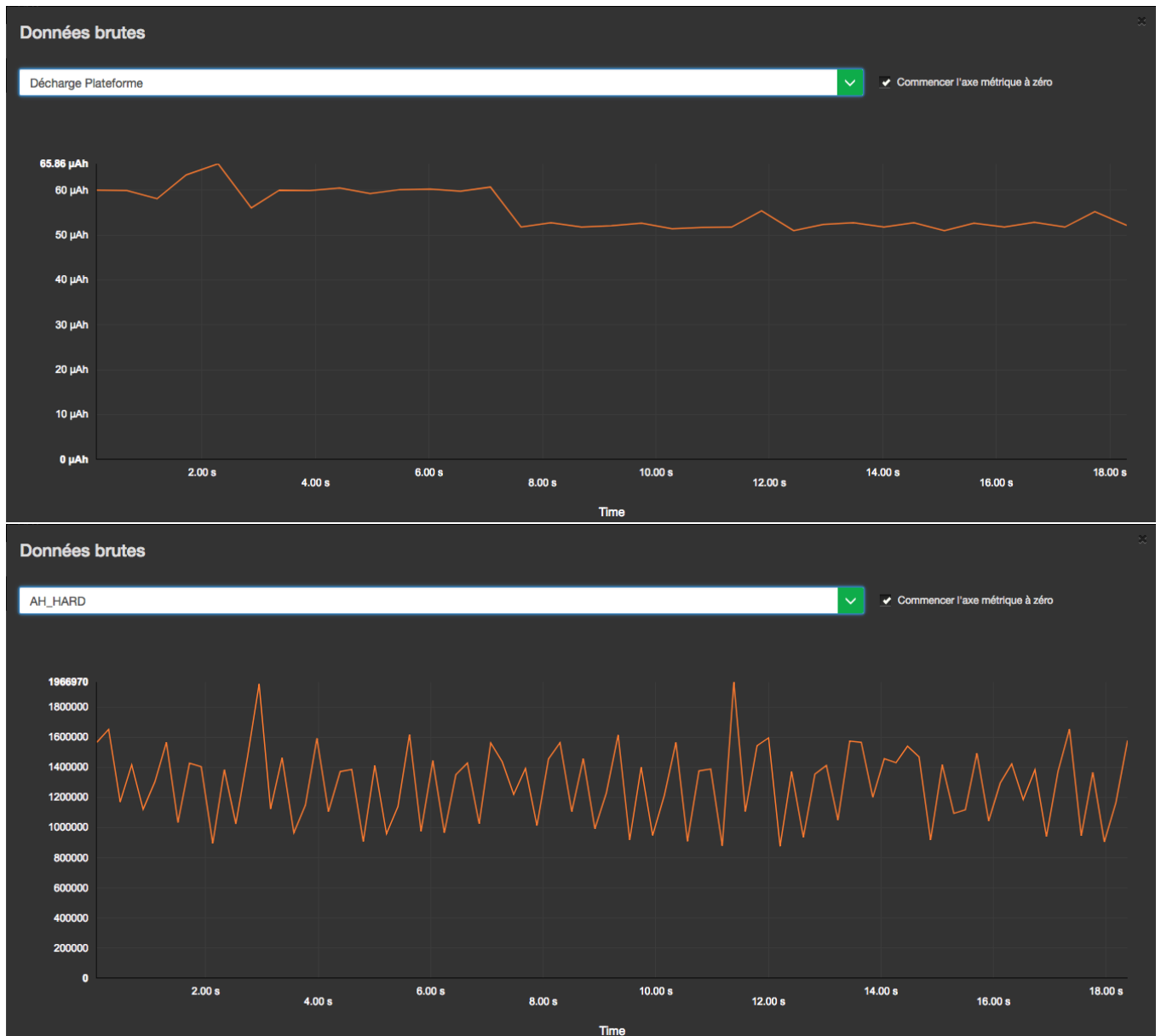


Figure 3 – *Visualisation des mesures des deux sondes sur l'interface GREENSPECTOR. Les résultats obtenus ne sont pas significatifs, ici on s'assure seulement de récupérer effectivement des données.*

4.2 Analyse des mesures brutes

On a commencé par analyser les données de mesure à partir des résultats bruts, dans le but de comparer les sondes. La confrontation de ces données n'a pas permis de relier la mesure physique à la mesure logicielle. Cependant, elle a permis de relever quelques points :

- Malgré un intervalle de mesure prévu de 200 ms, la sonde logicielle, lorsqu'elle relève les métriques CPU, mémoire etc... met davantage de temps à fournir un point de mesure (environ 500 ms). La période de la sonde physique étant de 200 ms, le retard de la sonde logicielle ne permettait pas la comparaison des courbes.
- Lorsqu'on s'intéresse à la décharge totale (somme de tous les points de mesure), on observe des instabilités au niveau de la sonde logicielle et des écarts parfois importants avec la sonde physique.

Pour la suite de l'analyse des mesures, on a mis à jour la sonde logicielle du smartphone pour ne mesurer que la décharge plateforme (toutes les 200 ms).

4.3 Analyse des mesures sur l'interface Greenspector

4.3.1 Comportement de la sonde logicielle

Après avoir mis à jour la sonde logicielle du SAMSUNG J3, plusieurs mesures de benchmark apk ont été effectuées. On a également allongé l'étape de chargement (2mn) afin d'avoir une meilleure vision des variations (fig. 4)

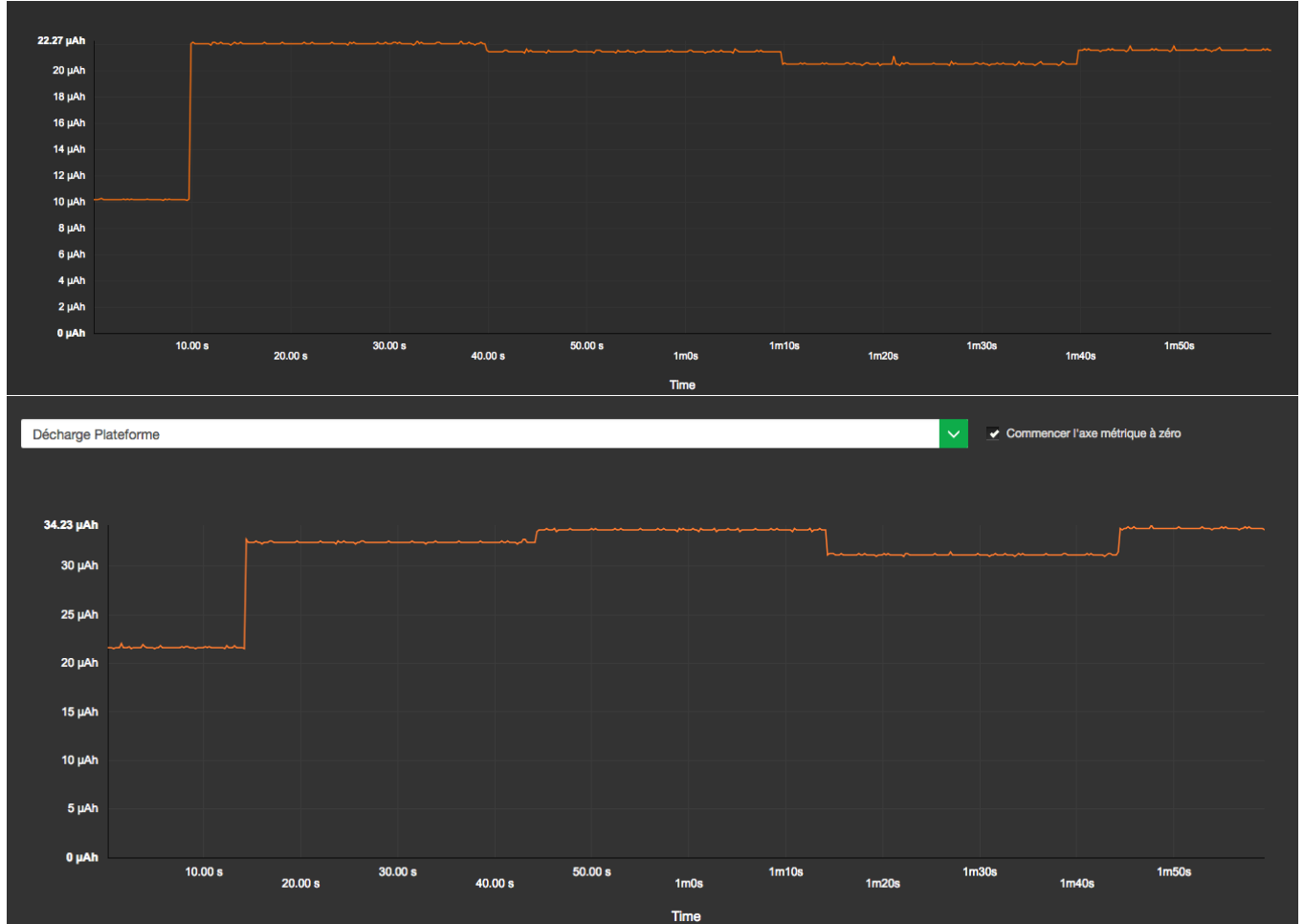


Figure 4 – Mesures logicielles. Étape de chargement (2mn). Les deux courbes correspondent respectivement à un test en luminosité minimum et maximum.

Les points de mesures semblent se regrouper autour de "valeurs paliers". En analysant les fichiers de mesure brute, on remarque que la majorité des points de mesure sur un palier correspond à une seule valeur (les variations au sein d'un palier sont peu nombreuses).

Ces observations semblent questionner la fiabilité de la sonde logicielle du SAMSUNG J3. On observe le même comportement (les périodes de mesure sont moins régulières) sur certaines mesures du SAMSUNG J3 de staging (fig. 5).

Ce comportement peut s'expliquer par le principe de mesure de la sonde logicielle, qui est le même que celui de la sonde physique. On récupère une valeur de courant que l'on "intègre" sur la durée de la mesure. Contrairement à la sonde physique, la mesure de courant est mise à jour beaucoup moins fréquemment (environ 30s). Comme les intervalles de mesure ne sont pas rigoureusement les mêmes, on observe quelques variations au sein de chaque palier.

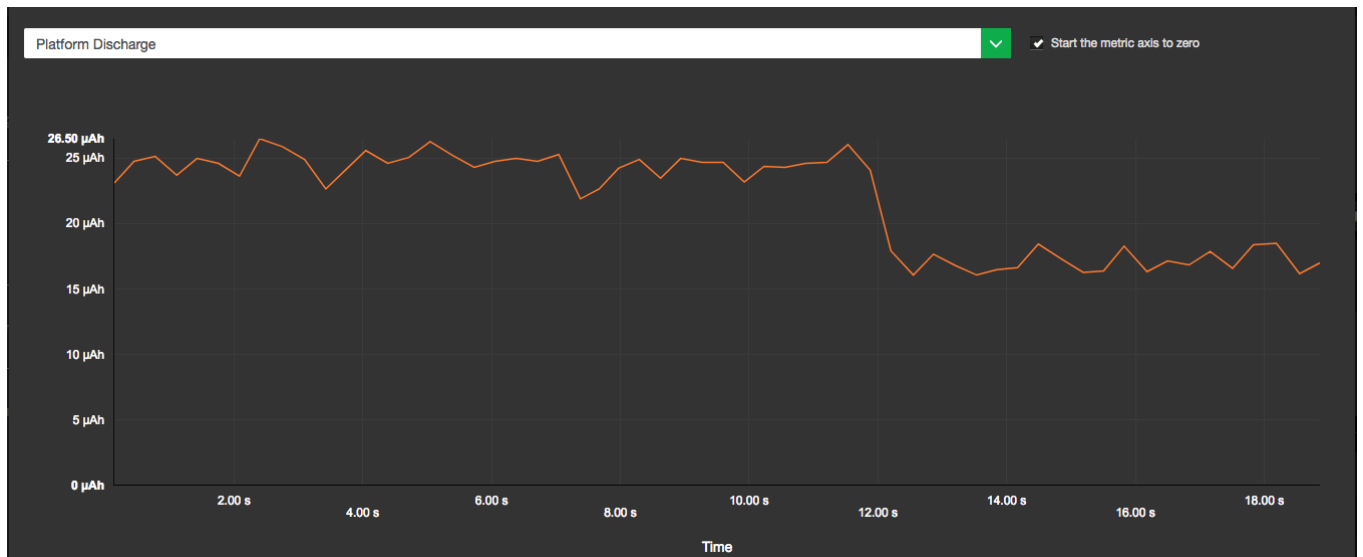


Figure 5 – *Résultat de mesure de l'étape de référence sur le SAMSUNG J3 de staging.*

En faisant varier progressivement la luminosité pendant une étape de 2mn, on observe néanmoins une augmentation progressive de la décharge (fig. 6).

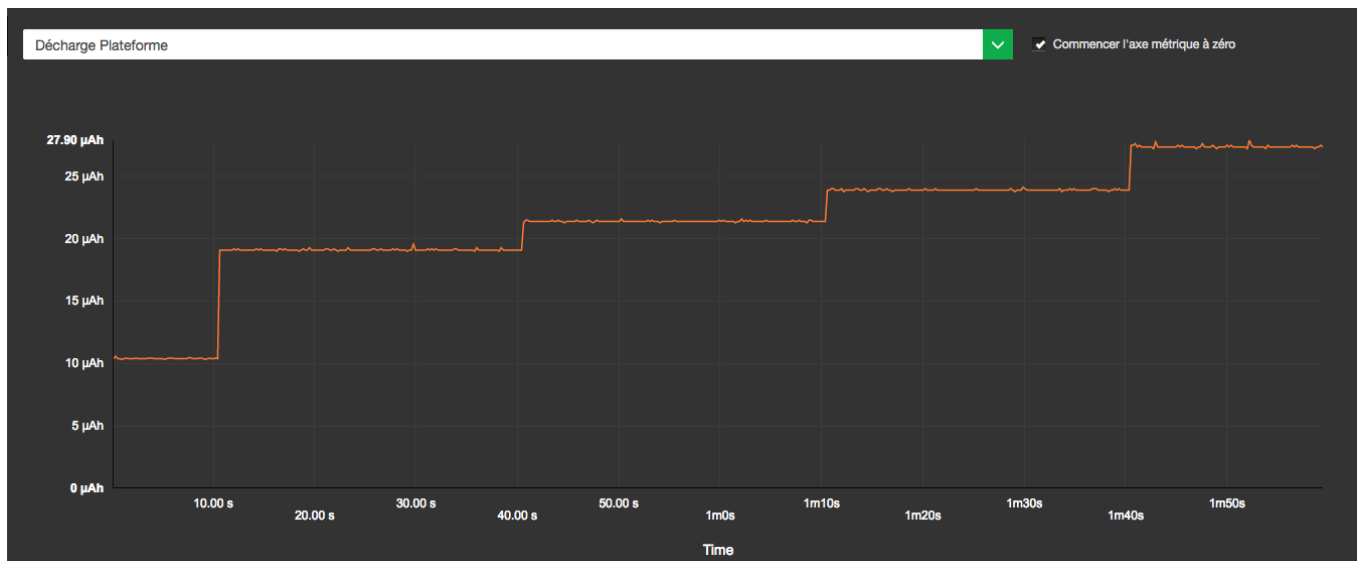


Figure 6 – *Décharge observée lors d'une augmentation progressive de la luminosité pendant 2mn.*

Ces observations montrent que la mesure logicielle du SAMSUNG J3 demeure utilisable (on a effectivement une hausse de consommation lorsqu'on augmente la luminosité). Il convient toutefois de considérer un niveau de confiance plus bas pour la sonde logicielle.

4.3.2 Comportement et fiabilité de la sonde physique

Affichage des mesures La mesure physique présente des courbes certainement moins lisibles (fig. 7), en raison de pics parasites¹, mais aussi d'une fréquence de mise à jour permettant d'enregistrer une valeur différente à chaque point (contrairement à la sonde logicielle). Dans tous les cas, la sonde physique ne présente pas le comportement en paliers observé avec la sonde logicielle.

1. ou pas ?

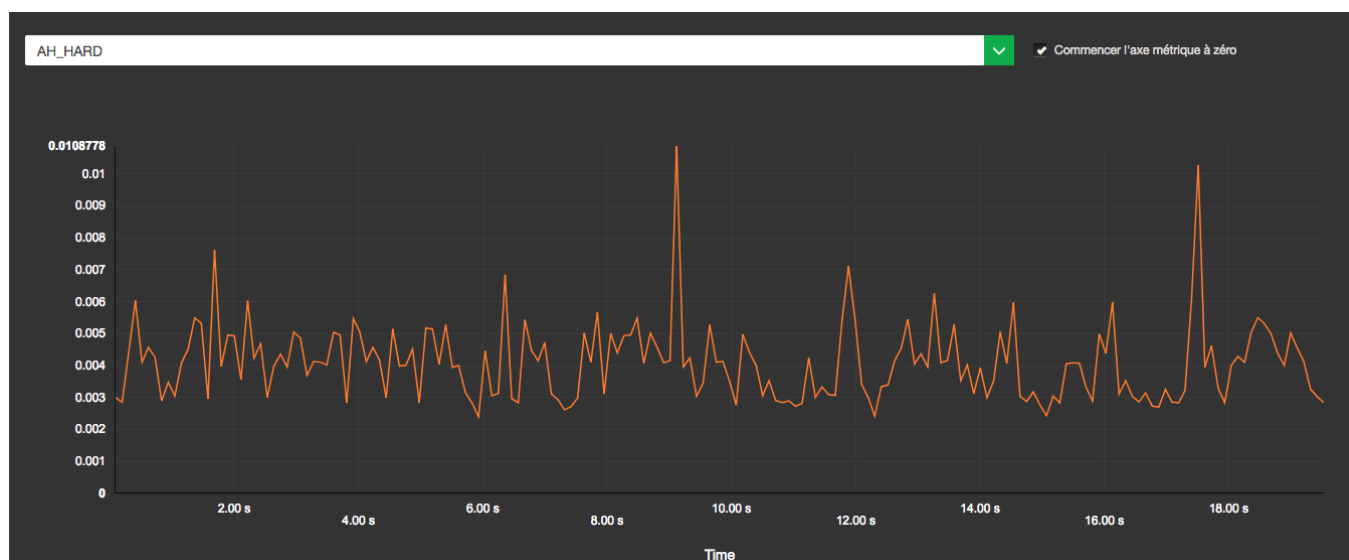


Figure 7 – Affichage de la mesure physique. Étape de référence.

Comportement de la sonde physique Au niveau de la décharge totale, la sonde physique présente des valeurs globalement stables sur l'ensemble des étapes de test (fig. 8).

AH_HARD ▾	Décharge Plateforme ▾	AH_HARD ▾	Décharge Plateforme ▾
1.96699	2.90 mAh	1.75432	2.46 mAh
1.73066	2.53 mAh	1.73848	2.03 mAh
0.928017	1.78 mAh	1.71077	1.99 mAh
0.844189	1.17 mAh	1.72924	1.96 mAh
0.8857	936.90 µAh	1.76326	1.94 mAh
0.879734	929.51 µAh	1.68853	1.91 mAh
0.798884	905.16 µAh	1.68098	1.87 mAh
0.820527	824.85 µAh	1.67196	1.80 mAh
0.9219	794.56 µAh	1.68258	1.73 mAh
0.897495	742.36 µAh	1.68141	1.50 mAh

Figure 8 – Décharge totale des sondes sur l'étape de référence et idle au premier plan.

On a cherché à comparer les comportement de la sonde physique et des sondes logicielles d'autres appareils (NEXUS 6 et SAMSUNG GALAXY). La sonde physique fournit un classement des étapes (ordonnées par consommation, fig. 9) similaire autres téléphones, alors que la sonde logicielle présente des différences plus importantes — par exemple, l'étape de chargement consomme moins que l'étape idle en arrière plan. On observe même des cas où l'étape de chargement est celle qui consomme le moins.

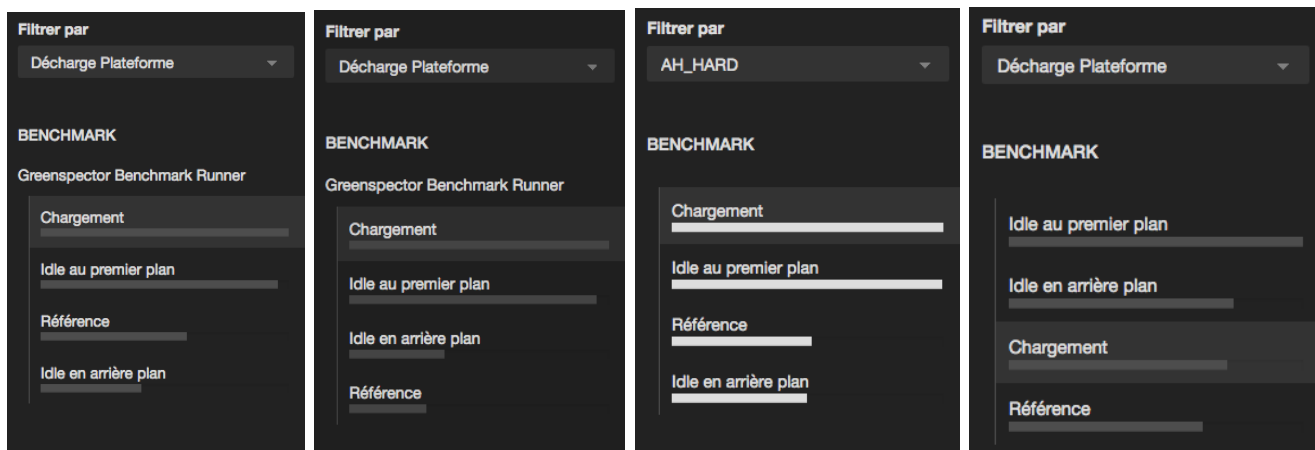


Figure 9 – Classement des étapes de test par ordre décroissant de consommation. Dans l'ordre : SAMSUNG GALAXY, NEXUS 6, SAMSUNG J3 (sonde physique), SAMSUNG J3 (sonde logicielle).

4.4 Conclusion.s

Les mesures réalisées à l'aide du dispositif hardware permettent de tirer plusieurs conclusions :

- La sonde logicielle du SAMSUNG J3 ne semble pas assez fiable pour obtenir des résultats précis sur la consommation d'énergie du téléphone. Elle peut cependant être utile pour avoir une idée, un ordre de grandeur de cette consommation sur ce type d'appareil.
- Au vu des résultats de mesure physique, on peut affirmer que la sonde est plus fiable que la sonde logicielle. Dans le cas du SAMSUNG J3, le dispositif hardware pourra venir compléter la mesure logicielle en apportant plus de précision. Si le montage est adaptable, il conviendrait d'effectuer des mesures physiques sur d'autres appareils afin de préciser le niveau de fiabilité de la sonde.

5 Travail restant

5.1 Montage

Avant d'envisager une éventuelle industrialisation du dispositif, il convient de le rendre plus stable. En effet, le prototype utilisé pour l'étude est fragile : les connecteurs côté batterie perdent le contact de temps en temps, et côté téléphone, le bout de plastique qui tient le connecteur est très instable, si bien qu'il n'est pas possible de déplacer normalement le téléphone sans qu'il ne s'éteigne. Des socles imprimés en 3D sont envisageables (selon les appareils) afin de stabiliser les points de contact, comme sur la figure 10.

Il faudra également adapter le dispositif pour d'autres téléphones. Pour cette étude on a utilisé un SAMSUNG J3 car sa batterie se retire facilement, et les points de contact électrique sont relativement accessibles. Ce n'est pas le cas de tous les appareils, il conviendra donc de réfléchir à la mise en place – accès à la batterie, ouverture, types de connecteurs – d'un tel dispositif sur des téléphones plus difficiles à manipuler.

5.2 Intégration

La fonctionnalité de mesure physique est implémentée seulement dans le TEST RUNNER pour l'instant. Si on veut l'intégrer dans le TESTBENCH AGENT, il faudra réfléchir à un certain nombre de questions – combien de sondes sur le même PC ? comment l'agent sait que la sonde est activée ?

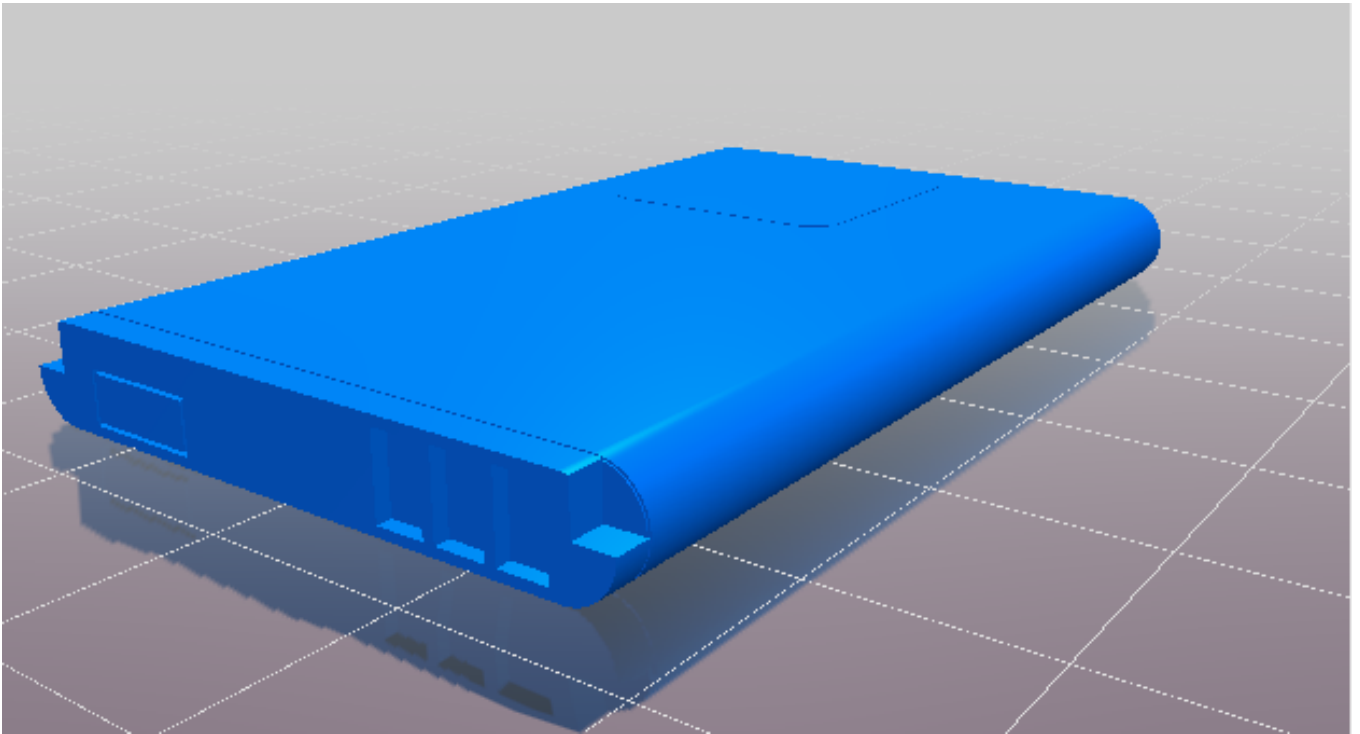


Figure 10 — *Exemple de "dummy battery" 3D, à placer à la place de la batterie. La batterie pourra être placée dans un socle 3D.*

tous les agents vont-ils avoir une sonde ? — qui n'ont pas été traitées lors de l'implémentation dans le TEST RUNNER.

5.3 Interface Greenspector

Au niveau du front, il faudra harmoniser l'affichage des mesures, en incluant notamment l'unité. De plus on pourra renommer les métriques `AH_HARD` et `AH_HARD/s` afin de les rendre plus compréhensibles par l'utilisateur de l'outil.