

# Mesure de soi

- La marche : Nextcloud avec GpxPod, Workflow et Analytics
- Le poids : OpenScale
- Le sommeil : SleepyHead et OSCAR
- Montre connectée : Gadgetbridge
- Une pointeuse avec Ktimetracker

# La marche : Nextcloud avec GpxPod, Workflow et Analytics

Pour ma santé, je dois faire du sport. J'ai choisi la marche rapide parce que ça peut se faire à peu près n'importe où, n'importe quand, que le matériel n'est pas hors de prix et que contrairement au *footing*, ça ne me flingue pas les articulations.

## Garmin Forerunner 110

J'ai acheté d'occasion une montre Garmin Forerunner 110 pour suivre mes traces GPS et mes stats (vitesse, longueur du parcours, fréquence cardiaque, etc). Elle se branche à l'ordinateur avec un câble et se présente comme une clé USB. Très simple et très pratique.

## Nextcloud et ses applications

GpxPod, pour peu qu'on installe le logiciel gpsbabel sur le serveur, est capable de convertir les traces `.fit` fournies par la montre Garmin en `.gpx`, format standard pour les traces GPS et permettra ensuite de voir le parcours contenu dans ce fichier, avec une jolie carte venant d'OpenStreetMap.

Je me sers d'un script Workflow perso pour extraire des données du fichier `.gpx` pour les mettre dans un fichier CSV qui sera lu par l'application Analytics pour me faire des graphes (fréquence cardiaque, vitesse moyenne, etc).

# Garmin Forerunner 610

La Forerunner 110 a un bracelet en caoutchouc qui peut claquer et qui n'est malheureusement pas remplaçable (la montre fait partie intégrante du bracelet). Certes, j'en ai retrouvé une autre d'occasion, mais le bracelet a lui aussi cassée. J'ai donc cherché une autre montre qui fait le même boulot, mais avec un bracelet changeable, et qui soit encore assez ancienne pour ne pas nécessiter une application Android.

J'ai trouvé, toujours d'occasion, une Garmin Forerunner 610. Elle se connecte à l'ordinateur avec un petit *dongle* USB, mais pour récupérer les informations de la montre, il faut installer un logiciel, antfs-cli, et sa dépendance, openant.

Antfs-cli récupère les fichiers `.fit` de la montre dans un sous-dossier de `~/config/antfs-cli`, il n'y a plus qu'à les envoyer à Nextcloud comme précédemment et c'est bon.

# Le poids : OpenScale

Pour surveiller mon poids, j'utilise l'application Android [openScale](#) qui se connecte en bluetooth à ma balance Sanitas SBF70 (balance recherchée à partir des balances supportées par openScale).

Bonus : la balance calcule aussi d'autres trucs, comme le pourcentage de masse grasse et osseuse.

Bug connu : openScale n'arrive plus à se connecter à ma balance à cause du passage à Android 11 de mon téléphone. J'ai encore un vieux téléphone qui peut se connecter, donc je récupère mes données d'un côté, j'exporte, j'envoie ça sur mon téléphone de tous les jours avec [KDE Connect](#) et j'importe. C'est un peu fastidieux, mais je ne fais pas ça tous les jours non plus.

# Le sommeil : SleepyHead et OSCAR

J'ai eu des problèmes de sommeil dus à des hypopnées, ce qui m'a amené à utiliser un appareil à pression positive continue (PPC) pendant la nuit. Un machin qui envoie de l'air sous pression dans le nez et/ou la bouche.

Ces appareils produisent de la donnée lors de leur utilisation : pression envoyée, temps d'utilisation, réveils de l'utilisateur, etc. Ces données sont envoyées à l'entreprise qui vous appareille, mais vous ne pouvez généralement pas les consulter.

C'est là qu'intervient SleepyHead, et maintenant son fork OSCAR (forké pour cause d'abandon de SleepyHead). Installez-le, donnez-lui les données de la carte SD de l'appareil et hop, vous aurez vos données avec des graphes en prime.

# Montre connectée : Gadget-bridge

J'ai eu pendant un temps un *Xiaomi MiBand 2*, petite montre connectée qui, entre autres choses, compte le nombre de pas effectués tous les jours et surveille le sommeil.

Cela fonctionne avec une application propriétaire, mais aussi avec [Gadgetbridge](#), qui permet d'avoir des graphes, de gérer les réveils, de transférer les notifications sur la montre...

Je ne me sers plus de cette montre connectée, mais je conseille toujours Gadgetbridge ☐☐

NB : Gadgetbridge gère plusieurs autres modèles de montres connectées.

# Une pointeuse avec Ktimetracker

Pour savoir combien de temps par jour je bosse, je me fais une pointeuse avec Ktimetracker : j'ai une tâche par jour (`1_lundi`, `2_mardi`, etc.), j'enclenche le chrono quand je commence ma journée et je le stoppe quand je fais une pause ou que j'ai fini ma journée.

Problème : ça nécessite d'ouvrir le logiciel pour voir où j'en suis (est-ce que je peux m'arrêter ?) et pour lancer/stopper le chrono.

J'ai enfin pris le temps de me pencher dessus et je me suis fait quelques scripts pour gérer tout ça plus simplement.

J'ai donc maintenant un script et un service *systemd*. Le script permet de lancer le chrono de la tâche du jour en cours, de stopper le chrono et de tourner en tâche de fond pour m'avertir que j'ai fait mes 8h de boulot et me proposer de stopper le chrono et de m'arrêter.

Le script communique avec Ktimetracker via `qdbus` et communique avec moi via `notify-send` (fourni par le paquet `libnotify-bin` : ça me fait des notifications comportant un bouton d'action sur lequel je peux cliquer pour lancer ou arrêter le chrono.

J'ai configuré des raccourcis claviers au niveau du système appelant les actions de lancement/stoppage du chrono, ainsi que celui qui affiche le chronomètre : j'ai toutes les informations/actions que je veux aisément disponibles et sans toucher ma souris ☐

Pour installer vous-même tout le bazar :

- installez les dépendances : `apt install ktimetracker qdbus libnotify-bin bc`
- installez le script quelque part sur votre système
- installez le service *systemd* dans `~/.config/systemd/user/ktstatus.service` (adaptez le chemin du script), puis faites `systemctl --user daemon-reload; systemctl --user enable --now ktstatus.service`
- créez-vous des raccourcis claviers vers `ktstatus.sh -g` pour lancer le chrono, `ktstatus.sh -s` pour l'arrêter et `ktstatus.sh` pour afficher le temps actuel.

Pour lancer Ktimetracker au démarrage sans qu'il n'apparaisse dans la barre des tâches ou dans les aperçus des bureaux, je le lance avec `kstart5 --iconify --skiptaskbar --skippager ktimetracker`. Faire `qdbus org.kde.ktimetracker /ktimetracker/MainWindow_1 org.qtproject.Qt.QWidget.hide` peu après (pas directement, y a un peu de latence) permet de le cacher. Il sera alors uniquement dans la boîte des miniatures (ce qui me va très bien).