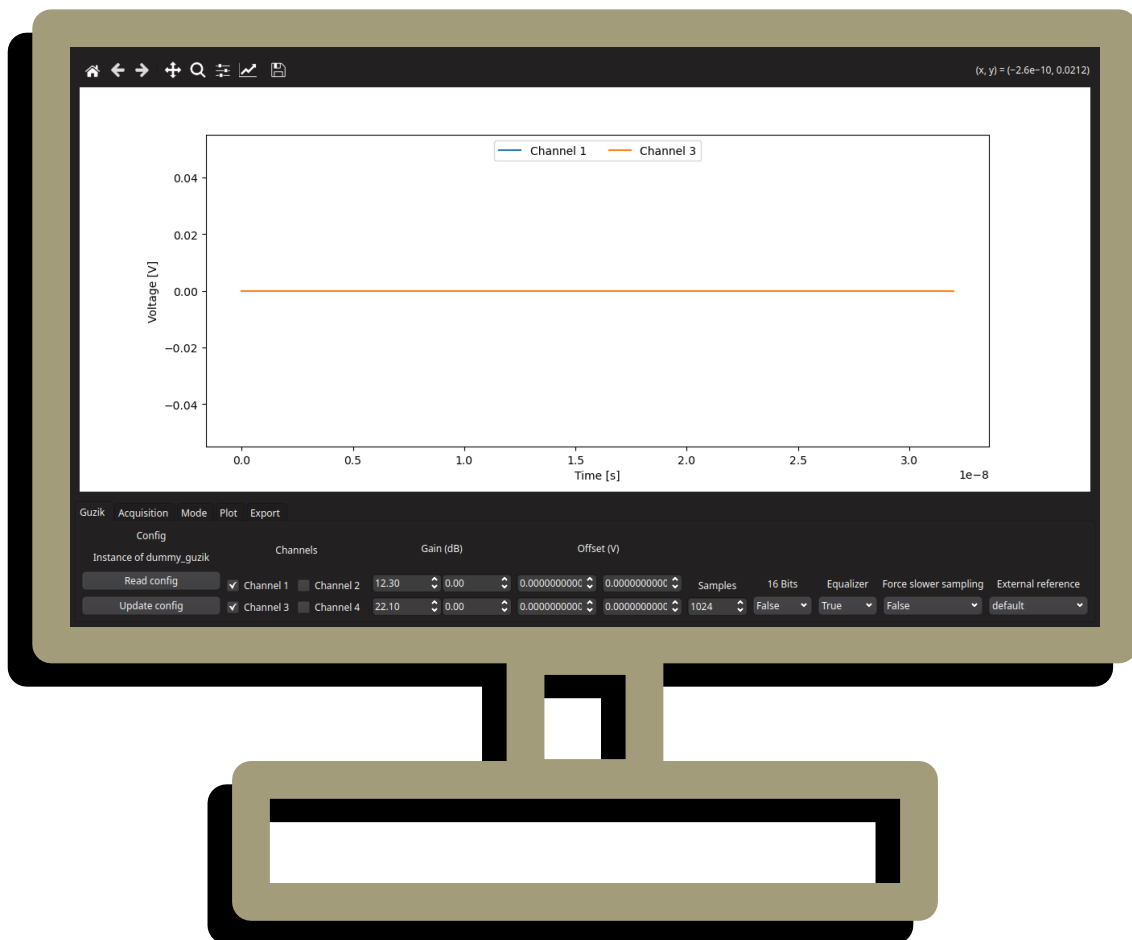


Guzik-O-Scope



Featuring a GUI!

Guzik-O-Scope

Guide de l'utilisateur

Alexandre Dumont
ReuletLab

Garantie

Les instructions, le code et ce manuel sont fournis sans garanties et sans support et peuvent ne pas fonctionner.

Table des matières

Garantie	i
Installation	1
Utilisation	3
Fonctionnalités	5

Installation

Ce chapitre explique la procédure pour installer le code Python pour le Guzik-O-Scope. L'installation est facultative. Pour utiliser le code sans l'installer, il faut tout de même le télécharger et s'assurer que le code soit dans un répertoire accessible par l'instance de Python.

Obtenir les dépendances

Pour pouvoir fonctionner, le Guzik-O-Scope a plusieurs dépendances, dont numpy et matplotlib, qui peuvent être facilement installées, si elles ne le sont pas déjà, et la librairie `SignalProcessing`. Celle dernière est une librairie maison qui se trouve sur Github à l'adresse <https://github.com/a-dumont/SignalProcessing>. Pour installer cette librairie, il est nécessaire de compiler du code, soit à partir de cygwin sur Windows, où directement avec gcc sur Linux.

Windows

Pour que le code puisse compiler correctement, certaines dépendances doivent être installées à partir de cygwin, notamment, CMake, MinGW, FFTW3 et Pybind11. Une fois ces dépendances installées via l'installateur de cygwin, il faut ouvrir exécuter les commandes suivantes.

```
1 git clone https://github.com/a-dumont/SignalProcessing
2 cd SignalProcessing/
3 mkdir build && cd build
4 CXX=/usr/bin/x86_64-w64-mingw32-g++.exe cmake ..
5 cmake --build . && cmake --install .
6 cd .. && /c/Anaconda3/python.exe setup.py install
```

Il est très important qu'à la dernière ligne, la commande `python.exe` soit celle du système et que son emplacement peut être différent de l'exemple fourni ici. Puisqu'il faut installer les dépendances, les privilèges d'administrateur peuvent être requis. Si c'est le cas, le terminal cygwin doit être ouvert en tant qu'administrateur.

Linux

Pour que le code puisse compiler correctement, certaines dépendances doivent être installées donc, CMake, FFTW3 et Pybind11. Une fois ces dépendances installées, il faut exécuter les commandes suivantes.

```
1 git clone https://github.com/a-dumont/SignalProcessing
2 cd SignalProcessing/
3 mkdir build && cd build
4 cmake .. && cmake --build . && cmake --install .
5 cd .. && python setup.py install
```

Puisqu'il faut installer les dépendances, les privilèges d'administrateur peuvent être requis.

Obtenir le code

La façon la plus simple d'obtenir le code est d'utiliser `git` à partir d'un terminal, que ce soit sur Windows, macOS ou Linux. Plus spécifiquement, il faut naviguer au répertoire choisi via le terminal et exécuter la commande suivante,

```
1 git clone https://github.com/a-dumont/GuzikGUI
```

Une façon alternative d'obtenir le code source est d'accéder au dépôt sur Github via un navigateur web et de télécharger le code dans une archive de type ZIP. Une fois cette archive décompresser dans le répertoire voulu, l'installation peut procéder normalement.

Installation

Pour installer le code du Guzik-O-Scope comme un module Python, il faut naviguer dans le répertoire `GuzikGUI` à partir d'un terminal et lancer la commande,

```
1 python setup.py install
```

Il est possible que l'installation requiert les privilèges d'administrateur.

Utilisation

Le Guzik-O-Scope est fait pour fonctionner avec pyHegel et son implémentation du Guzik. Par contre, le code contient une classe permettant d'imiter le comportement du Guzik qui est instanciée automatiquement si le code ne parvient pas à charger l'implémentation de pyHegel. Ce cas est particulièrement pratique pour tester des modifications au code du Guzik-O-Scope sans avoir besoin du matériel physique.

Application seule

Pour lancer le Guzik-O-Scope il suffit d'exécuter la commande suivante à partir d'un terminal.

```
1 python -m GuzikGUI
```

Il est possible d'écrire un script qui contient cette commande et de créer un raccourci pour celui-ci pour lancer l'application en un clic.

Application avec debug

Pour une raison ou une autre, il peut être utile d'avoir accès aux variables qu'utilise le Guzik-O-Scope. Il est possible de lancer l'application via Ipython de manière interactive avec les commandes suivantes.

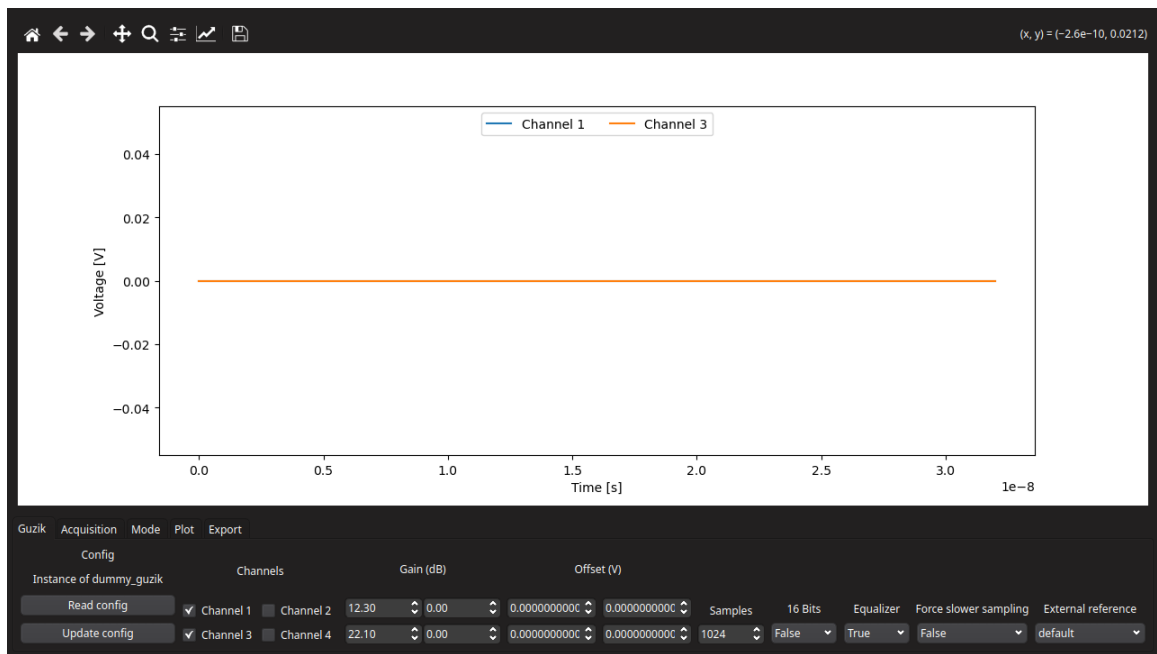
```
1 from GuzikGUI import launch
2 application, window = launch()
```

L'objet `window` contient toute les variables intéressantes.

Fonctionnalités

L'interface du Guzik-O-Scope est séparée en plusieurs onglets, ce chapitre passera, rapidement, en revue le rôle de chacun de ces onglets.

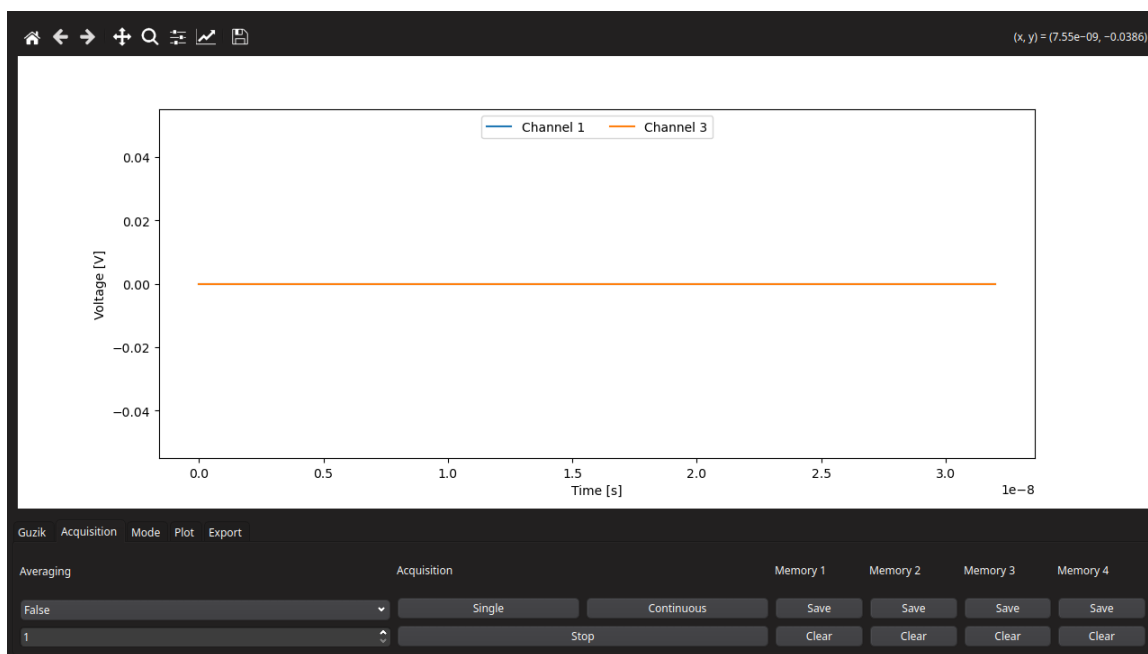
Guzik



Ce premier onglet sert à configurer les différents paramètres du Guzik. Il est possible d'y changer les canaux utiliser pour la détection, le gain et le décalage en tension sur chacun des canaux. Les paramètre restants sont le nombre de points pour chaque mesure, la résolution de chaque point de mesure, l'égaliseur, la fréquence de mesure et la présence d'un signal de référence externe. Pour envoyer les modifications effectuées à l'instrument, il faut cliquer sur le bouton *Update config*. Pour lire la configuration actuelle de l'instrument, il faut cliquer sur le bouton *read config*.

Acquisition

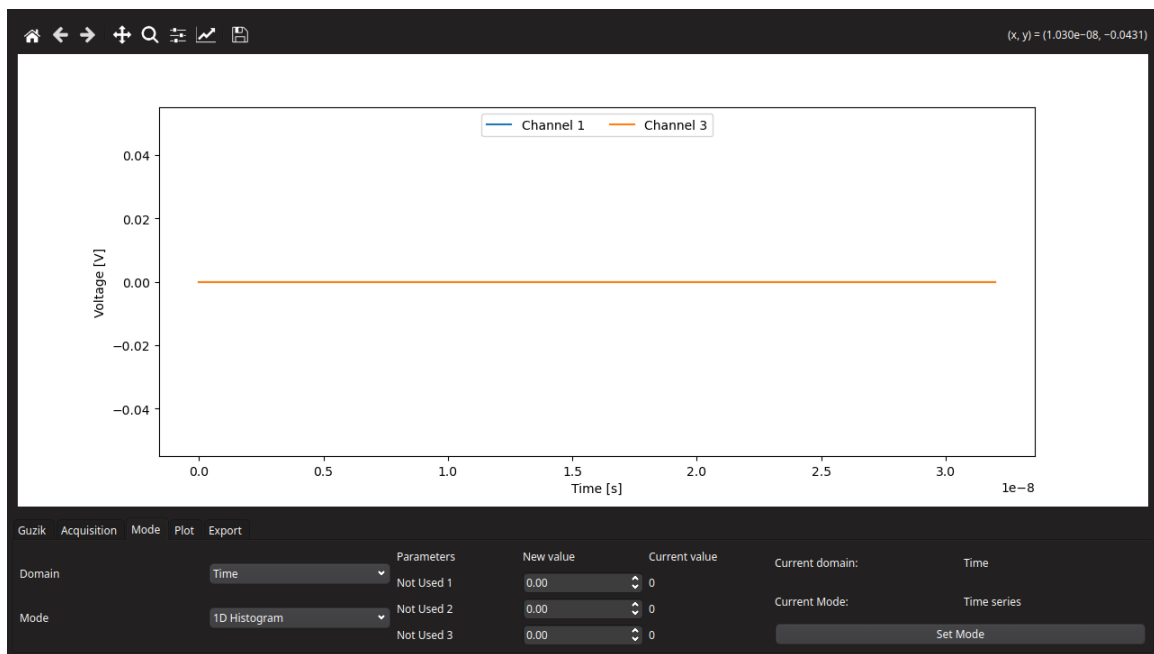
L'onglet *Acquisition* permet de choisir si les mesures seront moyennées ou non, et si oui combien de fois. Pour activer le moyennage, sous *Averaging* choisir l'option *True* et entrer un nombre dans le champ de texte. Il est aussi possible de choisir entre des mesures individuelle ou en continu en utilisant les boutons *Single* et *Continuous* respectivement. Pour une mesure individuelle, le Guzik-O-Scope moyenne le nombre de mesures spécifié et affiche le résultat en temps réel. Pour des mesures en continu, la moyenne est cyclique, c'est-à-dire que qu'à chaque fois que le nombre de mesures spécifié est atteint, la moyenne se réinitialise.



Comme le Guzik-O-Scope n'implémente pas de parallélisation, lorsque des mesures sont en cours l'interface graphique répond plus lentement. Pour retrouver le contrôle complet de l'interface il est possible d'interrompre une mesure en continue ou le moyennage d'une mesure individuelle en utilisant le bouton *stop*. Finalement, cet onglet permet aussi de stocker jusqu'à quatre courbes en mémoire pour permettre la comparaison entre différentes mesures en utilisant les différents boutons *save* et *clear*.

Mode

L'onglet *Mode* permet de choisir le type de mesure qui sera effectué par le Guzik-O-Scope. Il est possible de changer entre des mesures temporelles et des mesures en fréquence en changeant la valeur du champ *Domain*. Pour l'espace du temps ou l'espace des fréquences les différentes mesures disponibles sont listées dans le champ *Mode* juste en dessous de *Domain*. En temps, les mesures disponibles sont, *1D Histogram* qui trace les histogrammes en une dimension sur chaque canal de mesure actif, *2D Histogram* qui trace l'histogramme en deux dimensions des deux canaux actifs et finalement *Time series* qui trace le signal brut de chaque canal.



En fréquence, les mesures disponibles sont *Power Spectrum* qui trace la densité spectrale de bruit sur chaque canal actif et *Cross Power Spectrum* qui trace le module de la corrélation croisée entre les deux canaux actifs. Ces deux fonctions utilisent le paramètre *Block Size* qui permet de choisir la taille des blocs utilisés pour les transformés de Fourier. Pour changer le type de mesure ou un des paramètres comme *Block Size*, il est nécessaire de cliquer le bouton *Set Mode*.

Plot

Présentement, l'onglet *Plot* est peu utile mais le but est qu'il serve à changer le type de graphique pour une mesure donnée.

Export

Le dernier onglet, *Export*, permet d'exporter les données vers un fichier. Il est possible de sauvegarder les données de la dernière mesure ainsi que des quatre courbes en mémoire en cochant les cases associées. L'option *Compression* permet de déterminer si les fichiers seront compressés par la fonction `numpy.save` ou non. Finalement le bouton *Export* permet d'ouvrir l'explorateur de fichier pour effectuer la sauvegarde. Il est important de noter que le graphique affiché peut être exporter à partir de n'importe quel onglet en utilisant le bouton de sauvegarde qui se trouve au haut de l'interface.

