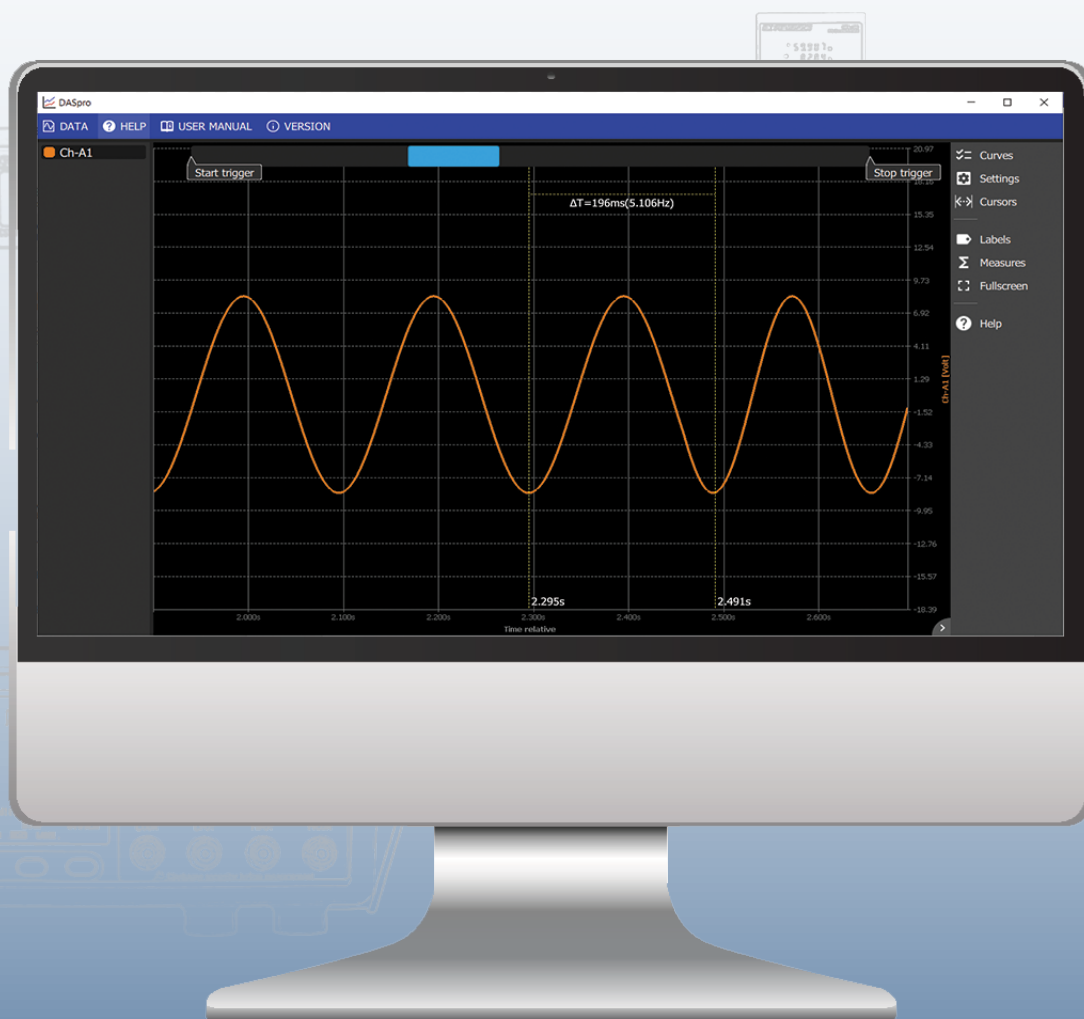


Manuel d'utilisation

DAS PRO



Sefram

a B&K Precision company

BK PRECISION

Table des matières

0.1	Visualisation et analyse graphique	4
0.2	Analyse d'un enregistrement	6
1	Format de fichier MDF4	7
1.1	Format	7
1.2	Version et conformité avec la norme ASAM	7
1.3	Interopérabilité	7
1.4	Fonctionnalités	7
1.5	Exemple	8

Table des figures

1	Réglages min et max des axes X et Y	4
2	Zoom et dézoom des axes X et Y	5
3	Paramètres de visualisation graphique	5
4	Fichier enregistré	6

Liste des tableaux

0.1 | Visualisation et analyse graphique



L'interface utilisateur permettant la visualisation $F(t)$, l'analyse d'un enregistrement sur l'appareil ou sur ordinateur (via le logiciel DASpro) est similaire.

Pour afficher le mesurande, effectuez un glisser-déposer dans la zone graphique (1), vous pouvez ajuster l'échelle souhaitée grâce aux différents gestes tactiles implémentés :

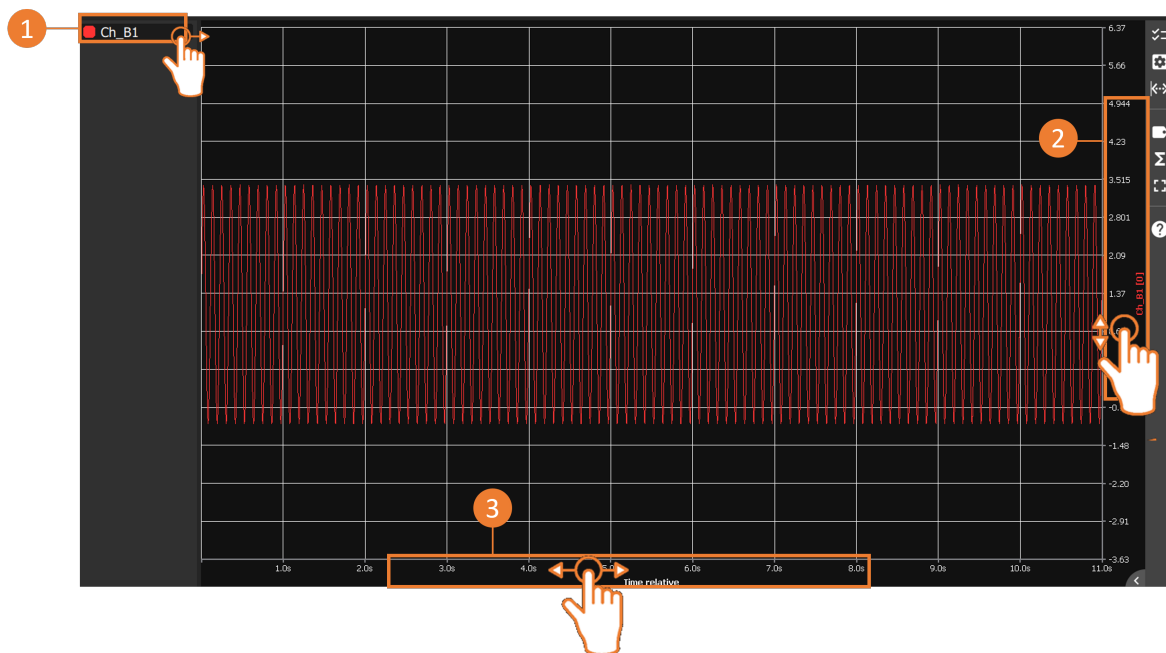


FIGURE 1 – Réglages min et max des axes X et Y

Vous pouvez définir les bornes minimum et maximum en glissant sur l'axe (2). Il en est de même pour l'axe des abscisses X (temps) (3).



Avec un appui court sur chaque axe, vous ouvrez une fenêtre de réglage où il est possible de rentrer manuellement la limite des bornes. Depuis ce menu, vous pouvez par exemple effectuer un « auto zoom » sur l'axe Y pour centrer automatiquement le mesurande ou encore ajouter une échelle supplémentaire sur l'axe Y.

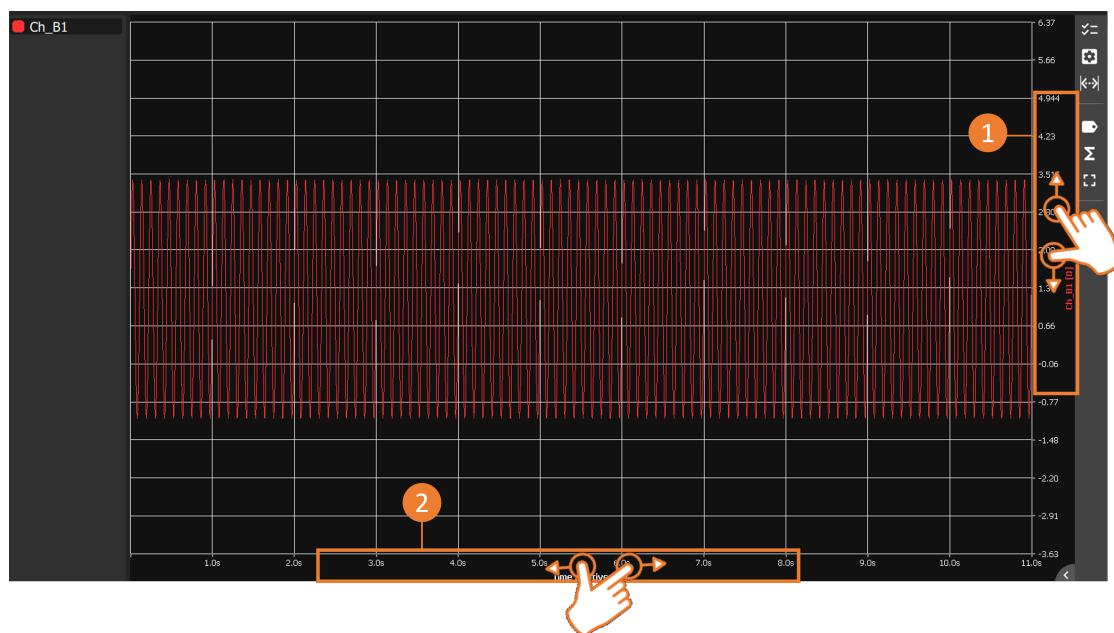


FIGURE 2 – Zoom et dézoom des axes X et Y

En éloignant ou rapprochant le pouce et l'index sur l'axe des ordonnées Y (amplitude) il est possible de zoomer et dé-zoomer entre les bornes définies (1). Il en est de même pour l'axe des abscisses X pour changer la base de temps (2).



Sur ordinateur ou si une souris est connecté à l'appareil, utilisez la roulette de la souris pour effectuer cette fonction en positionnant le curseur sur l'axe souhaité

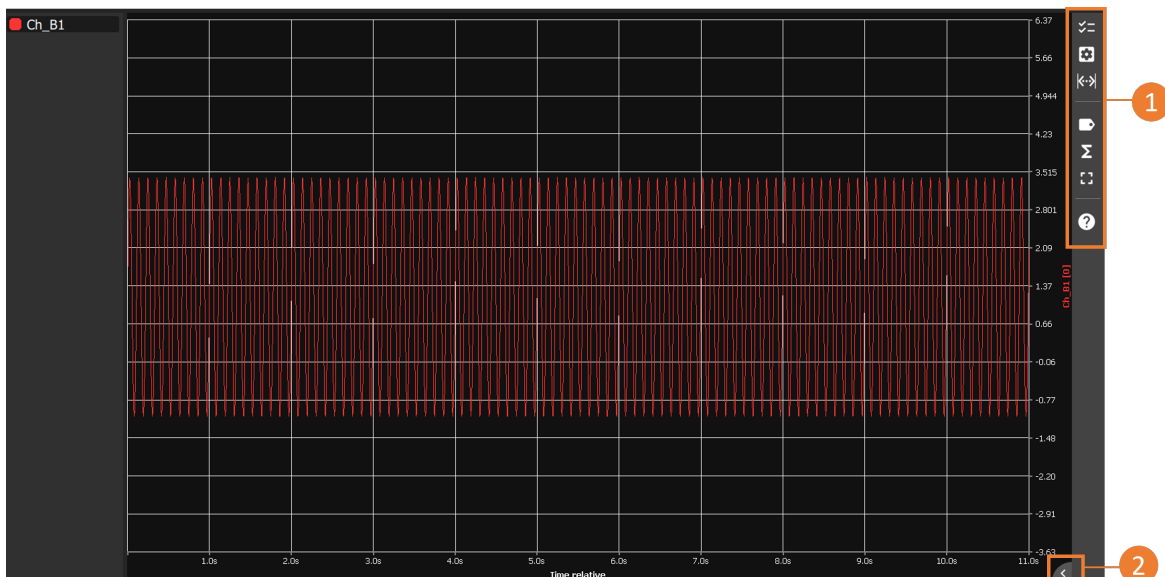


FIGURE 3 – Paramètres de visualisation graphique

Sur la barre verticale à droite de l'écran, un ensemble de paramètres est disponible (1). A l'aide de la flèche en bas à droite de l'écran, vous ouvrez la description textuelle de chaque paramètre (2).

Symbole	Description
	Choisit les mesurandes à afficher dans la zone graphique
	Permet de régler les paramètres d'affichage : division de la zone graphique en plusieurs écrans, choix des couleurs, image en arrière plan...
	Affiche/masque les curseurs verticaux et horizontaux
	Affiche/masque le nom complet de(s) mesurande(s) affiché(s) avec accès aux paramètres d'affichage
	Affiche/masque les calculs mathématiques prédéfinis en temps réel (voir chapitre calculs mathématiques)
	Affiche/quitte le mode plein écran
	Ouvre la fenêtre d'aide



Les calculs mathématiques prennent en compte l'ensemble des points du mesurande affichés à l'écran. Si des curseurs verticaux sont affichés, le calcul prendra en compte seulement les points entre les curseurs.

0.2 | Analyse d'un enregistrement

Pour ouvrir un fichier de mesure enregistré, allez dans « Enregistrement » depuis la barre de navigation principale.

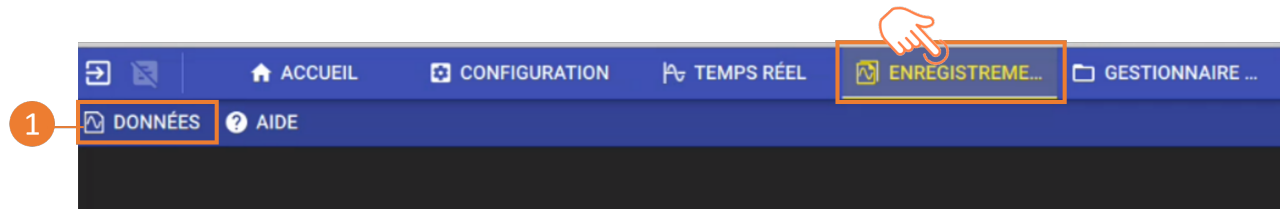


FIGURE 4 – Fichier enregistré

En appuyant sur données (1), vous pouvez :

- Accéder à l'ensemble des fichiers enregistrés
- Convertir le fichier de mesure affiché en format .csv qui se retrouvera dans le gestionnaire de fichier

Chapitre 1

Format de fichier MDF4

« Measurement Data Format version 4 (MDF4) » est un standard de fichier ASAM dédié au stockage des données de mesure dans un format de fichier binaire.

[Voir la norme](#)

1.1 | Format

Le MDF contient à la fois des données brutes de mesure ainsi que les métadonnées nécessaires pour l'interprétation des données brutes. Les métadonnées contiennent par exemple les informations pour la conversion des données brutes en des grandeurs physiques exploitables ou encore les noms des signaux respectant la norme ASAM. Le fichier est organisé en blocs binaires où chaque bloc se compose d'un nombre d'octets adjacents pouvant être vu en tant qu'enregistrement ou structure de données.

1.2 | Version et conformité avec la norme ASAM

Notre format de fichier suit le standard MDF 4.1.1. Il peut être vérifié à l'aide du MDF Validator 2.9.10.

1.3 | Interopérabilité

Nos fichiers MDF4 peuvent lus par les outils suivants :

- Flexpro
- Ni DIAdem
- Matlab + Vehicle Network Toolbox
- Python Asammdf
- Turbolab MDF4-LIB

D'autres logiciels peuvent être susceptibles d'ouvrir nos fichiers s'ils supportent le standard MDF4, toutefois nous ne les avons pas testés.

1.4 | Fonctionnalités

Principales fonctionnalités MDF4 présentes dans nos appareils :

- Champs de description du fichier : permet à l'utilisateur de stocker des informations sur le contexte de ses mesures
- Historique du fichier : sauvegarde la date de création du fichier
- Marqueur : marqueurs temporels ajoutés par l'utilisateur
- Données brutes : les données brutes sauvegardées en regard des fonctions de conversion définies dans l'en-tête
- Information de synchronisation temporelle : informations sur la source et la précision de la synchronisation temporelle
- Pièce jointe : le fichier de configuration du DAS est inclus dans le fichier d'enregistrement en tant que sauvegarde de la configuration de l'appareil
- Informations sur les voies : identifiant des voies, noms courts et longs des voies ainsi que la couleur du tracé
- Sous-échantillonnage calculé sur le groupe de fréquences le plus rapide

1.5 | Exemple

Ci-dessous un exemple d'implémentation Python utilisant la bibliothèque « Asammdf » permettant d'ouvrir un enregistrement MDF4

Listing 1.1 – Exemple d'utilisation de la bibliothèque MDF4 en Python

```
from asammdf import MDF

mdf = MDF('sample.mdf')
speed = mdf.get('WheelSpeed')
speed.plot()

important_signals = ['WheelSpeed', 'VehicleSpeed', 'VehicleAcceleration']
# get short measurement with a subset of channels from 10s to 12s
short = mdf.filter(important_signals).cut(start=10, stop=12)

# convert to version 4.10 and save to disk
short.convert('4.10').save('important_signals.mf4')

# plot some channels from a huge file
efficient = MDF('huge.mf4')
for signal in efficient.select(['Sensor1', 'Voltage3']):
    signal.plot()
```