

ETAS INCA-MIP V7.5



Guide de l'utilisateur

Copyright

Les informations contenues dans le présent document ne doivent pas être modifiées ou amendées sans l'accord spécifique de ETAS GmbH. ETAS GmbH n'est tenue que des obligations contenues dans le présent document. Le logiciel décrit dans le présent document est fourni sur la base d'un accord de licence général ou individuel. L'exploitation et la copie du présent document sont autorisées uniquement selon les conditions indiquées sur ce contrat.

En aucun cas, tout ou partie du présent document ne peut être copié, reproduit ou conservé dans un système de collecte des données ou traduit dans d'autres langues sans l'accord express écrit de ETAS GmbH.

© Copyright 2024 ETAS GmbH, Stuttgart

Les noms et les désignations utilisés dans ce document sont des marques déposées appartenant à leurs propriétaires respectifs.

MATLAB et Simulink sont des marques déposées de The MathWorks, Inc. Consultez le site [mathworks.com/trademarks](https://www.mathworks.com/trademarks) pour obtenir une liste de marques supplémentaires.

INCA-MIP V7.5 | Guide de l'utilisateur R02 FR | 06.2024

Sommaire

1	Introduction	6
1.1	Utilisation prévue	6
1.2	Groupe cible	6
1.3	Classification des messages de sécurité	6
1.4	Informations de sécurité	7
1.5	Protection des données	8
1.6	Sécurité des données et de l'information	8
2	À propos de INCA-MIP	9
2.1	INCA Définitions	10
3	Installation	12
3.1	Minimum requis	12
3.2	Installation	12
3.3	Mettre à jour les « liens Cache » de la boîte à outils MATLAB®	14
3.4	Désactiver les liens Caches des répertoires des répertoires de boîte à outils MATLAB®	14
3.5	Licences	15
4	Fonctions API	16
4.1	Connaître l'AddOn INCA-MIP grâce à des exemples:	19
4.2	Fonctions Générales	20
4.2.1	Listes les messages de l'AddOn INCA-MIP	20
4.2.2	Activer l'affichage d'information durant l'exécution des scripts	23
4.2.3	Afficher l'état de la licence INCA-MIP (INCA-MIP Evoluée)	23
4.2.4	Lire informations sur toutes les versions INCA installées	23
4.2.5	Lire Information sur tous les add-ons installés	24
4.2.6	Connaître la version de INCA	25
4.2.7	Lire les propriétés de INCA (INCA-MIP Evoluée)	25
4.3	Initialisation	26
4.3.1	Ouvrir INCA	26
4.3.2	Fermer INCA (INCA-MIP Evoluée)	27
4.3.3	Ouvrir une base de données	27
4.3.4	Importer une base de données (INCA-MIP Evoluée)	28
4.3.5	Lire les éléments de la base de données (INCA-MIP Evoluée)	29
4.3.6	Assigner un projet et un jeu de données à un dispositif (INCA-MIP Evo- luée)	31

4.3.7	Ouvrir une expérimentation	31
4.3.8	Réinitialiser une expérimentation	32
4.3.9	Lire les dispositifs (INCA-MIP Evoluée)	33
4.3.10	Lire les propriétés du dispositif (INCA-MIP Evoluée)	33
4.4	Mesure et Enregistrement	34
4.4.1	Lire les éléments de mesure (INCA-MIP Evoluée)	35
4.4.2	Lecture des fréquences d'échantillonnage (INCA-MIP Evoluée)	36
4.4.3	Ajouter une variable de mesure à une fréquence d'échantillonnage	36
4.4.4	Démarrer la mesure	38
4.4.5	Arrêter la mesure	39
4.4.6	Lire les propriétés de l'enregistrement (INCA-MIP Evoluée)	39
4.4.7	Régler les propriétés de l'enregistrement (INCA-MIP Evoluée)	41
4.4.8	Mode Lecture Enregistrement (INCA-MIP Evoluée)	44
4.4.9	Set Recording Mode (INCA-MIP Evoluée)	45
4.4.10	Démarre l'enregistrement	46
4.4.11	Arrêter un enregistrement	46
4.4.12	Régler le mode de lecture des données (Données Online / Offline)	47
4.4.13	Lecture des données de mesure	48
4.4.14	Réinitialiser les buffers tournant	51
4.4.15	Lecture de l'état du matériel (INCA-MIP Evoluée)	52
4.4.16	Définir les triggers (INCA-MIP Evoluée)	52
4.4.17	Execution du trigger manuel (INCA-MIP Evoluée)	56
4.4.18	Lire l'état d'un enregistrement (INCA-MIP Evoluée)	56
4.4.19	Lire la liste des variables de mesure (INCA-MIP Evoluée)	57
4.5	Calibration	57
4.5.1	Lire les éléments de calibration (INCA-MIP Evoluée)	58
4.5.2	Ajouter un élément de calibration	59
4.5.3	Lire la valeur de calibration	60
4.5.4	Modifier une valeur de calibration	62
4.5.5	Assigner un jeu de données au dispositif (INCA-MIP Evoluée)	66
4.5.6	Lister les jeux de données d'un dispositif (INCA-MIP Evoluée)	66
4.5.7	Etablir le mode de calibration (INCA-MIP Evoluée)	67
4.5.8	Groupement des dispositifs (INCA-MIP Evoluée)	68
4.5.9	Ecrire un fichier DCM (INCA-MIP Evoluée)	68
4.6	Gestion des pages mémoires	69
4.6.1	Activer une page mémoire	69
4.6.2	Get Current Page (INCA-MIP Evoluée)	69
4.6.3	Vérifier la protection en écriture	70
4.6.4	Télécharger vers l'ECU la page mémoire	70

4.6.5	Copier la page mémoire	71
4.6.6	Télécharger les différences	71
4.6.7	Télécharger depuis l'ECU les pages mémoires (INCA-MIP Evoluée)	72
4.7	Exemples	72
5	Création et distribution de fichiers exécutables autonomes à l'aide du compilateur MATLAB®	74
5.1	Compilation des fichiers m	74
5.2	Distribution des fichiers exécutables autonome	75
6	Informations des Contacts	76
Index	77

1 Introduction

1.1 Utilisation prévue

INCA et les add-ons INCA ont été développés et approuvés pour les applications et procédures automobiles telles que décrites dans la documentation utilisateur pour INCA et add-ons INCA.

L'AddOn INCA-MIP (INCA MATLAB® Integration Package) est une interface de programmation qui permet de contrôler les fonctionnalités de INCA depuis MATLAB®.

INCA et les add-ons INCA sont destinés à être utilisés dans des laboratoires industriels et des véhicules d'essai.

ETAS GmbH ne peut être tenue responsable des dommages causés par une utilisation incorrecte et le non-respect des .

1.2 Groupe cible

Ce produit logiciel et ce guide utilisateur s'adressent au personnel qualifié travaillant dans les domaines du développement et de la calibration des ECU automobiles, ainsi qu'aux administrateurs système et aux utilisateurs disposant de privilèges d'administrateur qui installent, entretiennent ou désinstallent le logiciel. Des connaissances spécialisées dans les domaines de la mesure et de la technologie des ECU sont requises.

Pour utiliser les APIs d'INCA-MIP, vous devrez être familier avec INCA et MATLAB®. Et également avec les scripts dans MATLAB®.

1.3 Classification des messages de sécurité

Les messages de sécurité signalent les dangers susceptibles d'entraîner des blessures ou des dommages matériels :



DANGER

DANGER indique une situation dangereuse avec un risque élevé de mort ou de blessure grave si elle n'est pas évitée.



AVERTISSEMENT

AVERTISSEMENT indique une situation dangereuse à moyen risque, susceptible d'entraîner la mort ou des blessures graves si elle n'est pas évitée.



ATTENTION

ATTENTION indique une situation dangereuse à faible risque, susceptible d'entraîner des blessures légères ou modérées si elle n'est pas évitée.

AVIS

AVIS indique une situation susceptible d'entraîner des dommages matériels si elle n'est pas évitée.

1.4 Informations de sécurité

Observer les informations de sécurité suivantes en travaillant avec INCA et les add-ons INCA :



AVERTISSEMENT

Risque de comportement inattendu du véhicule

Les opérations de calibration influencent le comportement de l'ECU et des systèmes connectés à l'ECU.

Ceci peut entraîner un comportement inattendu du véhicule, notamment l'arrêt du moteur, un freinage, une accélération ou une embardée du véhicule.

N'effectuer des opérations de calibration que si vous êtes formé à l'utilisation du produit et capable d'évaluer les réactions possibles des systèmes connectés.



AVERTISSEMENT

Risque de comportement inattendu du véhicule

La transmission de messages via des systèmes de bus, tels que CAN, LIN, FlexRay ou Ethernet, influence le comportement des systèmes qui y sont connectés.

Ceci peut entraîner un comportement inattendu du véhicule, notamment l'arrêt du moteur, un freinage, une accélération ou une embardée du véhicule.

Ne procéder à la transmission de messages via un système de bus que si vous disposez de connaissances suffisantes sur l'utilisation du système de bus concerné et êtes capable d'évaluer les réactions possibles des systèmes connectés.

Respecter les instructions des Consignes de sécurité ETAS et les informations de sécurité fournies dans l'aide en ligne et les guides utilisateur. Ouvrir les Consignes de sécurité ETAS dans le menu Aide d'INCA ? > **Consignes de sécurité.**

1.5 Protection des données

Si le produit contient des fonctions qui traitent des données personnelles, les exigences légales en matière de protection des données et les lois sur la confidentialité des données doivent être respectées par le client. En tant que contrôleur des données, le client conçoit généralement le traitement ultérieur. Il doit donc vérifier si les mesures de protection sont suffisantes.

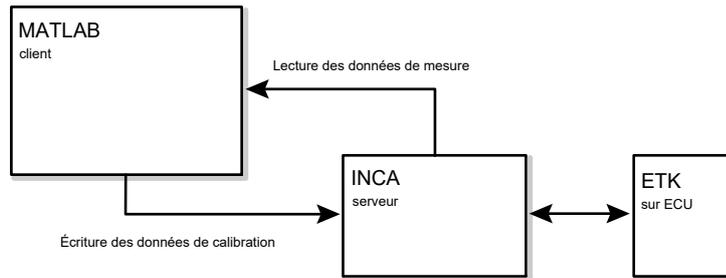
1.6 Sécurité des données et de l'information

Pour traiter les données en toute sécurité dans le contexte de ce produit, voir la section « Sécurité des données et de l'information » de l'Aide INCA.

2 À propos de INCA-MIP

L'AddOn INCA-MIP (INCA INCA Integration Package) est une interface de programmation qui permet de contrôler les fonctionnalités de MATLAB® depuis MATLAB®. Dans ce cas, INCA se comporte comme un client accédant aux ressources de MATLAB®, qui est donc le serveur.

Le tableau suivant illustre une application typique de l'API INCA-MIP, utilisant INCA pour adresser un ETK.



L'aperçu suivant présente les fonctionnalités de INCA auxquelles il est possible d'accéder à partir de MATLAB®.

gestionnaire de pages de mémoire

Le passage d'une page de mémoire à l'autre et le téléchargement de pages de mémoire vers l'unité de commande sont pris en charge.

Calibration

Toutes les variables d'étalonnage d'une expérience INCA peuvent être modifiées. Les valeurs peuvent être lues et mises à jour pour chaque élément et pour les distributions de points de rupture associées, le cas échéant.

Measuring

Toutes les variables de mesure d'une expérience INCA peuvent être lues. En outre, les mesures peuvent être lancées et arrêtées depuis le site MATLAB®. Toutes les données de performance qui sont disponibles sur INCA sont également accessibles sur MATLAB®. Le débit des données de performance à l'interface INCA-MIP a été optimisé.

Comme INCA génère les variables de mesure et d'étalonnage sous forme de doubles, les formules de conversion pour le retraitement dans MATLAB® ne sont pas nécessaires.

Les fonctions API de INCA décrites dans ce document sont appelées à partir des scripts de INCA (fichiers M), et peuvent être utilisés pour définir l'ensemble du flux de contrôle des expérimentations MATLAB®.

Les chapitres suivants décrivent l'architecture sous-jacente de l'API INCA-MIP et les fonctions API disponibles, ainsi que les procédures d'installation. Les opérations sous INCA ou sous MATLAB® ne font pas partie de ce manuel.

2.1 INCA Définitions

La description des APIs utilise certains termes qu'un utilisateur expérimenté de INCA connaît. Ci-dessous se trouve une brève définition de ces termes.

Variable de calibration

Une variable de calibration est un élément qui peut être lu et modifié. Les variables de calibration peuvent être des scalaires, des vecteurs, des matrices, des courbes ou des cartographies. Les distributions associées de points d'appui sont également lisibles et modifiables.

Enregistrement des données

Un enregistrement est composé d'une base de temps et de toutes les valeurs de mesure d'une fréquence d'échantillonnage pour une acquisition simple. Les données d'enregistrement pour une fréquence d'échantillonnage est composé de plusieurs enregistrements qui sont générés durant la procédure d'enregistrement globale.

Module

Un module de mesure utilisé pour extraire les variables de mesure d'une grille de mesure particulière. Certains modules de mesure supporte également la calibration de variables. Par exemple, les modules SMB sont utilisables pour les mesures uniquement, alors que les ETK conviennent à la fois aux mesures et aux calibrations.

Données de mesure

Tous les enregistrements capturés d'une mesure pour différentes fréquences d'échantillonnage.

Raster de mesure

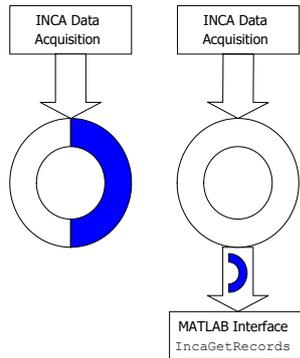
Taux d'acquisition (fréquence de mesure) utilisé pour mesurer un ou plusieurs signaux dans un groupe de signaux.

Il est possible de combiner deux rasters ou plus dans un soi-distant raster multiple. Il suffit de combiner les noms de raster au moyen d'un caractère « + », p. ex. '10 ms+100 ms'. L'utilisation d'un tel raster multiple crée un nouveau raster virtuel. Chaque signal peut uniquement être mesuré dans exactement un raster ou un raster multiple.

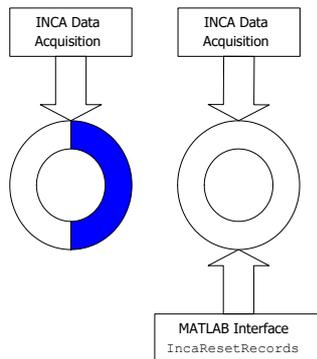
Buffer tournant

Pour assurer un transfert fiable des données de mesure d'INCA vers MATLAB[®], un buffer tournant dédié est utilisé pour chaque fréquence d'échantillonnage (groupe de signaux). Pendant une visualisation online d'INCA ou un enregistrement, les variables de mesure acquises sont automatiquement sauvegardées dans le buffer tournant.

La commande `IncaGetRecords` peut être utilisée pour lire les échantillons de temps et les données du buffer tournant vers MATLAB® :



Le buffer tournant est limité à 1 Mo par signal et peut enregistrer les données de mesure pendant environ 8 secondes en fonction de la fréquence de mesure. Après ce temps de conservation les informations seront écrasées. Pour éviter de perdre des données la commande `IncaGetRecords` doit être exécutée régulièrement. Typiquement cela doit être fait toutes les secondes. Avec la commande `IncaResetRecords` la base de temps et les données contenues dans le buffer tournant pour toutes les fréquences d'échantillonnage peuvent être effacées. Toutes les données déjà sauvegardées dans le buffer tournant seront perdues :



Signal

Un signal est un élément dont la valeur est mesurée dans INCA. Chaque signal est caractérisé par son type de données (booléen, entier, flottant), sa longueur (1, 2, 4 ou 8 octets) et sa formule de conversion. La conversion à partir de la valeur de mesure physique sur le niveau d'implémentation est spécifiée dans la formule de conversion.

Fréquence d'échantillonnage

Groupe de signaux; une fréquence d'échantillonnage est composée de plusieurs signaux individuels. Elle est caractérisée par sa fréquence d'acquisition (fréquence de mesure) qui est la même pour tous les signaux du groupe. Chaque groupe de signaux a un nom unique.

3 Installation

INCA-MIP est une extension fonctionnelle de INCA.

MATLAB[®] utilise des appels de fonctions liés dynamiquement, appelés fichiers MEX, pour communiquer avec d'autres applications. INCA-MIP est un ensemble de fichiers MEX qui sont copiés dans les sous-répertoires associés au répertoire du programme MATLAB[®] lors de l'installation.

INCA-MIP est disponible en deux versions. L'AddOn de base : « INCA-MIP Base API » est disponible directement après l'installation. Pour utiliser la version évoluée « INCA-MIP Extended API set » une licence sous forme de clef est requise. Une liste des APIs et des descriptions respectives peut être trouvée dans "[Fonctions API](#)" à [la page 16](#).

3.1 Minimum requis

Pour utiliser l'AddOn INCA-MIP, INCA doit être installé sur votre ordinateur. Pour plus d'informations sur les exigences du système INCA, consultez le guide d'installation INCA.

Si vous souhaitez développer vous-même des scripts INCA pour accéder à MATLAB[®], vous aurez besoin d'une licence complète de MATLAB[®].

INCA-MIP pour INCA V7.5 nécessite les versions de programme suivantes :

- INCA V7.5 SPx



Note

INCA V7.5 est nécessaire pour l'installation de cette version INCA-MIP.

Assurez-vous que le numéro de version INCA de l'installation INCA est compatible avec le numéro de version du module complémentaire INCA-MIP.

Après l'installation, vous pouvez utiliser cette version INCA-MIP pour travailler avec toute version INCAV7.x. "[Ouvrir INCA](#)" à [la page 26](#)

- MATLAB[®] 64 bit version 2016a ou supérieure (pour l'installation intégrée de MATLAB[®])

Pour plus d'informations sur les versions de INCA prises en charge, contactez votre support MATLAB[®].

3.2 Installation

Avant d'installer l'AddOn il est nécessaire de connaître le type de l'installation. Les types suivants sont possible :

- **Installation en mode MATLAB[®] intégré (MATLAB integrated installation)**

Sélectionnez cette option si vous utilisez une version de MATLAB[®] pour

développer des scripts .MATLAB[®]

– Installation dans ETASData

Sélectionnez l'option runtime si vous souhaitez uniquement lancer des scripts compilé de INCA-MIP ou si vous souhaitez utiliser MATLAB[®] avec différentes versions de MATLAB[®] sur votre PC. Pour plus de détails, voir ci-dessous.

Pour installer INCA-MIP :

Vérifiez que INCA est installé sur votre PC et que la version d'INCA est compatible avec la version de l'AddOn INCA-MIP.

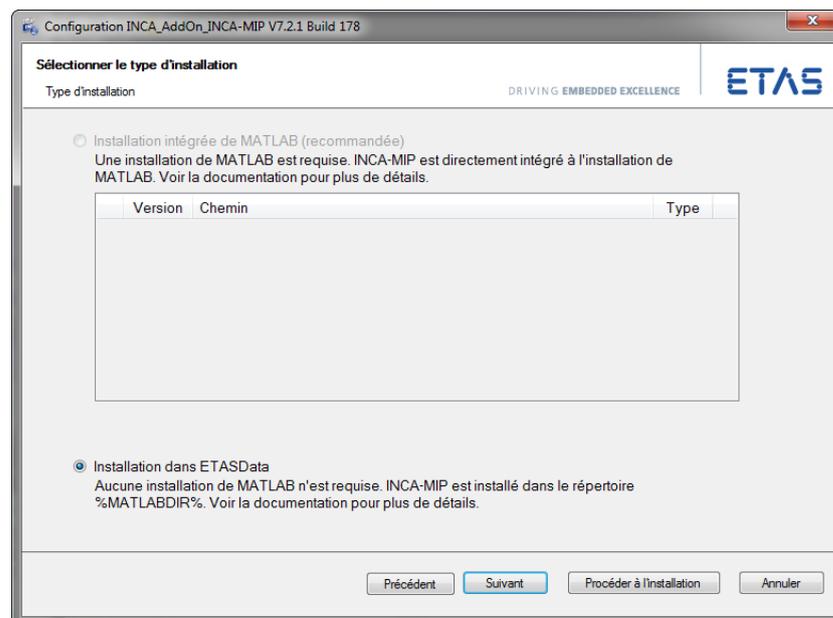
Si vous souhaitez développer vos propres scripts MATLAB[®] pour accéder à INCA, vérifiez que MATLAB[®] est installé sur votre ordinateur et que la version de MATLAB[®] est compatible avec celle de l'AddOn INCA-MIP.

1. Fermez tous les programmes actifs.
2. Selon la réglementation propre à votre entreprise, les fichiers d'installation sont fournis sur un lecteur réseau ou sur un DVD.

En utilisant le DVD, la routine d'installation démarre automatiquement. Si ce n'est pas le cas, exécutez manuellement le fichier `Autostart.exe` sur le DVD, cliquez sur Installation, puis sélectionnez l'installation INCA-MIP.

Si vous installez le programme depuis un lecteur réseau, exécutez le fichier `setup.exe`.

3. Suivez les instructions de la routine d'installation pour installer INCA-MIP sur votre ordinateur.
4. Dans le programme d'installation, vous êtes prié de préciser le type d'installation souhaitée :



5. Si vous souhaitez développer des scripts MATLAB[®] avec une seule version MATLAB[®] installée sur votre ordinateur, sélectionnez l'option **MATLAB[®] Integrated installation**.

ou

Sélectionnez l'option **Installation into ETASData** pour les cas suivant :

- Vous souhaitez utiliser INCA-MIP avec différentes versions de MATLAB®. Dans ce cas vous devez ajouter le sous-répertoire d'INCA-MIP dans le répertoire toolbox INCA-MIP pour chaque installation MATLAB® avant de pouvoir utiliser les commandes .MATLAB®. Consulter votre documentation utilisateur de MATLAB® pour savoir comment ajouter des chemins d'accès supplémentaires à MATLAB.MATLAB®.
- Vous ne voulez exécuter que des exécutables autonomes facilement disponibles MATLAB® créés avec MATLAB®.



Note

L'installation dans ETASData est nécessaire si vous utilisez des fichiers programme qui contiennent des instructions INCA pour la commande d'MATLAB®. Dans ce cas vous n'avez pas besoin de licence MATLAB®. Les fichiers exécutables doivent être mise à disposition par les développeurs avec une installation MATLAB® (voir "[Création et distribution de fichiers exécutables autonomes à l'aide du compilateur MATLAB®](#)" à la page 74).

6. Continuez l'installation.

Obtenir la licence pour INCA-MIP :

Si vous projetez d'utiliser la version évoluée des fonctions APIs « INCA MATLAB Extended API set », un fichier de licence sera requis.

Pour plus d'informations sur les licences, voir "[Licences](#)" à la page suivante.

3.3 Mettre à jour les « liens Cache » de la boîte à outils MATLAB®

Après installation de l'API INCA-MIP, vous devez d'abord mettre à jour le cache pour les répertoires de la boîte à outils MATLAB® au cas où ce cache a été activé lors de votre installation de .MATLAB®. C'est le cas avec la version MATLAB® V6 ou plus si vous utilisez les paramètres par défaut ; les liens sont désactivés dans les versions antérieures à la V6. Les liens Cache doivent être mis à jour pour que les fichiers utilisés dans l'AddOn INCA-MIP API soient enregistrés dans MATLAB®.

Veillez consulter la documentation utilisateur de MATLAB® pour mettre à jour les liens Cache des répertoires de la boîte à outils .MATLAB®.

3.4 Désactiver les liens Caches des répertoires des répertoires de boîte à outils MATLAB®

Si vous travaillez avec l'AddOn INCA-MIP, il est recommandé de désactiver les « liens Cache » des répertoires de la boîte à outils MATLAB®. Sinon, des dysfonctionnements peuvent survenir si l'AddOn INCA-MIP ou des scripts

indépendamment nouvellement intégrés se peuvent pas être trouvés.

Une autre solution que la désactivation des « liens Caches » consiste à forcer la mise à jour des liens Cache s'ils sont activés comme indiqué précédemment.

Cependant pour éviter toute mauvaise manipulation nous vous recommandons de désactiver les « liens Caches » lors de l'utilisation de l'AddOn INCA-MIP.

Veillez consulter la documentation utilisateur de MATLAB[®] pour activer ou désactiver les liens Cache des répertoires de la boîte à outils .MATLAB[®]

3.5 Licences

Une licence valide est nécessaire pour utiliser le logiciel. Vous pouvez obtenir une licence de l'une des manières suivantes :

- de votre coordinateur d'outils
- via le portail en libre-service sur le site web d'ETAS à l'adresse www.etas.com/support/licensing
- via le gestionnaire de licences ETAS

Pour activer la licence, vous devez saisir l'ID d'activation que vous avez reçu d'ETAS lors de la procédure de commande.

Pour plus d'informations sur la gestion des licences ETAS, consultez la [FAQ sur la gestion des licences ETAS](#) ou l'aide du gestionnaire de licences ETAS.

Pour ouvrir l'aide du gestionnaire de licences ETAS

Le gestionnaire de licences ETAS est disponible sur votre ordinateur après l'installation de tout logiciel ETAS.

1. Dans le menu Démarrer de Windows, sélectionnez **E > ETAS > ETAS License Manager**.
via le gestionnaire de licences ETAS
2. Cliquez dans la fenêtre du gestionnaire de licences ETAS et appuyez sur la touche F1.
L'aide du gestionnaire de licence ETAS s'ouvre.

Utilisation distante d'INCA

Il n'est pas permis d'utiliser une licence de poste de travail INCA (LICENCE BASÉE SUR MACHINE) via des applications Internet ou réseau, par exemple Microsoft Remote Desktop, ou d'autres services de terminaux ou de dispositifs. Cela ne s'applique pas pour le fonctionnement d'INCA par le biais de protocoles de banc d'essai (ASAP 3, MCD-3 MC, iLinkRT).

L'utilisation d'une licence de type UTILISATEUR NOMMÉ ou d'une licence FLOTTANTE permet d'accéder à INCA via Microsoft Remote Desktop dans la mesure où il est garanti qu'un seul utilisateur utilise une licence en même temps.

4 Fonctions API

L'AddOn INCA-MIP fournit une liste de fonctions pour automatiser le fonctionnement de INCA. Quelques fonctions sont disponibles dans la version de base de l'AddOn INCA-MIP, d'autres ne sont disponibles qu'en achetant la version évoluée de cet AddOn INCA-MIP.



Note

Les commandes qui ne sont disponibles qu'avec la version évoluée de INCA-MIP sont protégées par une licence. Vous ne pourrez pas utiliser les fonctions de la version évoluée sans licence valide, l'exécution de ces scripts MATLAB® générera une exception.

Nous vous recommandons, avant d'utiliser les commandes de la version évoluée de l'AddOn INCA-MIP, de vérifier la validité de la licence à l'aide de la commande `IncaIsLicenseValid`.

Le tableau suivant répertorie toutes les fonctions de l'API INCA-MIP qui sont disponibles dans le module complémentaire pour INCA V7.5. Il indique

- Si la fonction est disponible dans la version de base de l'AddOn (INCA-MIP Base Package) ou uniquement dans sa version évoluée (INCA-MIP Extended package);
- Si la fonction est utilisée pour l'initialisation, la mesure, la calibration, la gestion des pages mémoire ou s'il s'agit d'une fonction plus générale;
- L'endroit dans le document où vous pourrez trouver plus d'informations concernant la fonction.

Fonction	Base ^a	Evo. ^b	Catégorie	Voir
<code>IncaAddCalibrationElement</code>	x	x	Calibration	à la page 59
<code>IncaAddMeasureElement</code>	x	x	Mesure	à la page 36
<code>IncaBrowseCalibrationElements</code>		x	Calibration	à la page 58
<code>IncaBrowseItemsInFolder</code>		x	Initialisation	à la page 29
<code>IncaBrowseMeasureElements</code>		x	Mesure	à la page 35
<code>IncaClose</code>		x	Initialisation	à la page 27
<code>IncaCopyPageFromTo</code>	x	x	Gestion Page Mémoire	à la page 71
<code>IncaDatabaseImport</code>		x	Initialisation	à la page 28
<code>IncaDownloadDifferences</code>	x	x	Gestion Page Mémoire	à la page 71

Fonction	Base ^a	Evo. ^b	Catégorie	Voir
IncaDownloadPage	x	x	Gestion Page Mémoire	à la page 70
IncaExecuteManualTrigger		x	Mesure	à la page 56
IncaGetCalibrationValue	x	x	Calibration	à la page 60
IncaGetCurrentPage		x	Gestion Page Mémoire	à la page 69
IncaGetDatasetsForDevice		x	Calibration	à la page 66
IncaGetDeviceProperties		x	Initialisation	à la page 33
IncaGetDevices		x	Initialisation	à la page 33
IncaGetHardwareStatus		x	Mesure	à la page 52
IncaGetInstalledAddOnInfo	x	x	General	à la page 24
IncaGetInstalledProductInfo	x	x	General	à la page 23
IncaGetMeasureRatesForDevice		x	Mesure	à la page 36
IncaGetProperties		x	General	à la page 25
IncaGetRecordingMode		x	Mesure	à la page 44
IncaGetRecordingProperties		x	Mesure	à la page 39
IncaGetRecordingState		x	Mesure	à la page 56
IncaGetRecords	x	x	Mesure	à la page 48
IncaGetRecordStruct		x	Mesure	à la page 57
IncaGetVersion	x	x	General	à la page 25
IncaGroupDevices		x	Calibration	à la page 68
IncalsLicenseValid		x	General	à la page 23
IncaMessagelds	x	x	General	à la page 20
IncalsPageWriteProtected	x	x	Gestion Page Mémoire	à la page 70
IncaOpen	x	x	Initialisation	à la page 26
IncaOpenDatabase	x	x	Initialisation	à la page 27
IncaOpenExperiment	x	x	Initialisation	à la page 31
IncaResetExperiment	x	x	Initialisation	à la page 32
IncaResetRecords	x	x	Mesure	à la page 51

Fonction	Base ^a	Evo. ^b	Catégorie	Voir
IncaSetCalibrationMode		x	Calibration	à la page 67
IncaSetCalibrationValue	x	x	Calibration	à la page 62
IncaSetDatasetInDevice		x	Calibration	à la page 66
IncaSetMeasureReadMode	x	x	Mesure	à la page 47
IncaSetProjectAndDatasetInDevice		x	Initialisation	à la page 31
IncaSetRecordingMode		x	Mesure	à la page 45
IncaSetRecordingProperties		x	Mesure	à la page 41
IncaSetTrigger		x	Mesure	à la page 52
IncaShowMessages	x	x	General	à la page 23
IncaStartMeasurement	x	x	Mesure	à la page 38
IncaStartRecording	x	x	Mesure	à la page 46
IncaStopMeasurement	x	x	Mesure	à la page 39
IncaStopRecording	x	x	Mesure	à la page 46
IncaSwitchPage	x	x	Gestion Page Mémoire	à la page 69
IncaUploadPages		x	Gestion Page Mémoire	à la page 72
IncaWriteToFile		x	Calibration	à la page 68

a Fonctions supportées par la version de base de l'AddOn ("INCA-MIP Base Package")

b Fonctions supportée par la version évoluée de l'AddOn ("INCA-MIP Extended Package")

Par la suite, la description des fonctions est organisée en fonction de leurs champs d'application:

- ["Fonctions Générales" à la page 20](#)
- ["Initialisation" à la page 26](#)
- ["Mesure et Enregistrement" à la page 34](#)
- ["Calibration" à la page 57](#)
- ["Gestion des pages mémoires" à la page 69](#)

De plus, un certain nombre de fichiers d'exemples fournis avec la version de développement de l'AddOn INCA-MIP sont décrits au ["Connaître l'AddOn INCA-MIP grâce à des exemples:" à la page suivante.](#)

Des exemples sont donnés dans ["Exemples" à la page 72.](#)

**Note**

L'interface INCA-MIP fonctionne toujours à partir des paramètres globaux des Options Utilisateur de .INCA. Pour plus d'information sur les Options Utilisateur de INCA veuillez consulter la documentation INCA.

4.1 Connaître l'AddOn INCA-MIP grâce à des exemples:

L'AddOn INCA-MIP est livré avec des exemples. Dans le cas où vous sélectionnez l'installation de la version évoluée, ces fichiers d'exemples types sont automatiquement installés sur votre ordinateur en complément des fichiers MEX. Les fichiers types utilisent plusieurs exemples pour expliquer l'utilisation de l'AddOn INCA-MIP.

Les exemples incluent certains fichiers M qui utilisent l'AddOn INCA-MIP, ainsi qu'une base de données INCA où les éléments utilisés dans les fichiers scripts ont déjà été créés.

Les exemples sont copiés pendant l'installation dans les répertoires suivants (voir "[Installation](#)" à la page 12) :

- Pour une installation en mode Installation into MATLAB[®] :
Fichiers M: %MatlabDir%\toolbox\matlab\demos
- Pour une installation en mode Installation into ETASData:
Fichiers M : %EtasDataDir%\INCA-MIPx64
Base de données démo INCA: %EtasDataDir%\Database\db_matlabtest

Pour utiliser les fichiers exemples, vous devez d'abord lancer INCA et ouvrir la base de données fournie. Il n'est pas nécessaire d'avoir du matériel.

Les fonctions des fichiers M sont décrites ci-dessous.

- `tOpen.m` – établit une connexion entre INCA et MATLAB[®]. Cette fonction doit être utilisée au démarrage de chaque session INCA-MIP avant l'utilisation d'une autre fonction de l'API MATLAB[®].
- `tDummy.m` – ouvre une expérimentation INCA vide en utilisant une configuration matérielle ayant un module de test VADI. Le script crée plusieurs variables de mesure dans l'environnement INCA.
- `tEtKDummy.m` – ouvre une expérimentation INCA vide en utilisant une configuration matérielle ayant un module de test ETK. Le script crée plusieurs variables de mesure et de calibration dans l'environnement INCA. Il reçoit également les pages de travail et de référence, il lit les variables de mesure et de calibration et modifie les valeurs des variables de calibration indépendantes.

- `tGetRecords (aGroupName) .m` – rassemble les données de mesure du groupe `aGroupName` pendant 20 secondes puis les transmet à MATLAB®. Cette fonction est utilisable avec les exemples VADI et ETK. (pour les “fréquences d’échantillonnage,” voir ["INCA Définitions" à la page 10.](#))
- `tPrintDB ({aFolder{, aFileId}}) .m` – Écrit tout le contenu de la base de données commençant par le répertoire `aFolder` vers un fichier `aFileId`. Si cette fonction est utilisée sans paramètre, toute l’arborescence de la base de données est imprimée par la sortie standard.
- `tHWStatus .m` – Exemple de l’utilisation de la fonction API `IncaGetHardwareStatus`. MATLAB® se connecte à une expérimentation déjà ouverte et choisit le premier élément de mesure trouvé dans le premier dispositif de mesure trouvé. Il continue une mesure de 5 minutes. S’il y a un avertissement ou une erreur pendant la mesure le cycle de mesure est abandonné puis remis en marche après un délais de 5 secondes.

4.2 Fonctions Générales

Les fonctions API générales suivantes sont disponibles:

4.2.1 Listes les messages de l’AddOn INCA-MIP

Les commandes de l’AddOn INCA-MIP peuvent retourner des erreurs.

En utilisant les blocs try/catch, une information détaillée de l’erreur est retournée.

Exemple

```
try,
    <command_1>
    ...
    <command_n>
catch,
    [msgstr,msgid] = lasterr
    ...
end
```

avec:

<code>msgstr</code>	Une phrase de description
<code>msgid</code>	L'identificateur de message. Les identificateurs de message suivant sont disponibles: <ul style="list-style-type: none"> — <code>INCA:ParameterError</code> — <code>INCA:ReturnParameterError</code> — <code>INCA:WrongParameterValue</code> — <code>INCA:WrongParameterType</code> — <code>INCA:NaN</code> — <code>INCA:ExecutionError</code> — <code>INCA:ResourceError</code> — <code>INCA:RasterFull</code> — <code>INCA:ObjectIsWriteProtected</code> — <code>INCA:CallSequenceError</code> — <code>INCA:LicenseError</code> — <code>INCA:RecordingInProgress</code> — <code>INCA:NotInstalled</code> — <code>INCA:WrongVersion</code>

Description des identificateurs de message:

<code>INCA:ParameterError</code>	Mauvais nombre de paramètres d'entrée passé dans la fonction (Paramètres de droite)
<code>INCA:ReturnParameterError</code>	Mauvais nombre de paramètre de sortie renvoyé par la fonction (paramètre de gauche)
<code>INCA:WrongParameterValue</code>	Une des valeurs des paramètres d'entrée de la fonction est en dehors de sa plage valide ou de sa spécification
<code>INCA:WrongParameterType</code>	Un des paramètre d'entrée de la fonction n'est pas du bon type
<code>INCA:NaN</code>	Un des paramètre contient une valeur du type 'not a number' (pas un nombre)
<code>INCA:ExecutionError</code>	Durant l'exécution de la commande une erreur a eu lieu. Essayer d'exécuter la fonctionnalité à partir de INCA peut donner plus d'informations sur l'erreur. Redémarrer INCA ou l'ordinateur peut aussi aider.

<code>INCA:ResourceError</code>	Impossible d'obtenir des ressources du système d'exploitation. Redémarrer INCA ou l'ordinateur peut aider.
<code>INCA:RasterFull</code>	La liste d'acquisition est pleine pour cette fréquence d'échantillonnage lors de l'ajout d'une mesure
<code>INCA:ObjectIsWriteProtected</code>	Impossible de calibrer à cause d'une protection en écriture de l'élément
<code>INCA:CallSequenceError</code>	Avant d'exécuter cette commande d'autres commandes doivent être exécutées en premier. Par Exemple <code>IncaOpenExperiment</code> est nécessaire avant <code>IncaAddMeasurementElement</code> .
<code>INCA:LicenseError</code>	Pour exécuter la commande avec les paramètres donnés, une licence est requise.
<code>INCA:RecordingInProgress</code>	Impossible d'exécuter la commande demandée (p. ex. activer ou désactiver les signaux pour l'enregistrement avec <code>SetRecordingMode</code>) parce qu'un enregistrement est en cours.
<code>INCA:NotInstalled</code>	Impossible d'ouvrir la version INCA spécifiée avec la commande <code>IncaOpen</code> parce que la version INCA correspondante n'est pas installée.
<code>INCA:WrongVersion</code>	Impossible d'ouvrir la version INCA spécifiée avec la commande <code>IncaOpen</code> pour l'une des raisons suivantes : <ul style="list-style-type: none"> – INCA est déjà démarré et une commande <code>IncaOpen</code> est exécutée avec un paramètres de version différent de la version INCA déjà ouverte – Une commande <code>IncaOpen</code> est exécutée depuis INCA-MIP pour INCA Vx.y avec un paramètre de version avec version majeure != x

4.2.2 Activer l'affichage d'information durant l'exécution des scripts

Nom	<code>IncaShowMessages</code>	
Description	Active ou désactive l'affichage d'information dans la fenêtre MATLAB [®] pendant l'exécution du script.	
Syntaxe	<code>IncaShowMessages (VraiOuFaux)</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>VraiOuFaux</code>	Paramètre numérique qui est soit égal à zéro soit différent de zéro. Si le paramètre est zéro, l'affichage de l'information est désactivé, autrement l'affichage est activé (défaut).
Exemples	<pre>IncaShowMessages (0) ; IncaShowMessages ;</pre>	

4.2.3 Afficher l'état de la licence INCA-MIP (INCA-MIP Evoluée)

Nom	<code>IncaIsLicenseValid</code>	
Description	Renvoie un statut indiquant si la licence INCA-MIP est valide ou non	
Syntaxe	<code>s = IncaIsLicenseValid</code>	
Arguments de sortie	<code>s</code>	Statut de la licence — 0 : Pas de licence valide — 1 : Licence valide
Arguments d'entrée	none	
Exemples	<code>statut = IncaIsLicenseValid</code>	

4.2.4 Lire informations sur toutes les versions INCA installées

Nom	<code>IncaGetInstalledProductInfo</code>	
Description	Lire informations sur toutes les versions INCA installées. Cette commande peut être exécutée avant <code>IncaOpen</code> .	
Syntaxe	<code>info = IncaGetInstalledProductInfo</code>	

Arguments de sortie	<code>info</code>	information sur les versions INCA installées en tant que struct. MATLAB® pour chaque installation et comprenant les entrées suivantes :
	<code>info.name</code>	le nom du produit
	<code>info.version</code>	la chaîne de version du produit
	<code>info.hotfixVersion</code>	le hotfix installé sous forme de chaîne ou une chaîne vide si aucun hotfix n'est installé
Arguments d'entrée	<code>none</code>	
Exemples	<code>i = IncaGetInstalledProductInfo;</code>	

4.2.5 Lire Information sur tous les add-ons installés

Nom	<code>IncaGetInstalledAddOnInfo</code>	
Description	Fournit des informations sur tous les add-ons installés pour un produit donné. Cette commande peut être exécutée avant <code>IncaOpen</code> .	
Syntaxe	<code>info = IncaGetInstalledAddOnInfo(productName, productVersion)</code>	
Arguments de sortie	<code>info</code>	information sur les add-ons installés en tant que struct. MATLAB® pour chaque installation et comprenant les entrées suivantes :
	<code>info.name</code>	le nom de l'add-on installé
	<code>info.version</code>	la chaîne de version de l'add-on installé

Arguments d'entrée	<code>productName</code>	le nom du produit
	<code>productVersion</code>	la version du produit sous forme de chaîne. La chaîne de version complète est pertinente.
Exemples	<pre>i = IncaGetInstalledAddOnInfo('INCA', 'V7.5.0'); i = IncaGetInstalledAddOnInfo('INCA', 'V7.5.1 Beta 100');</pre>	

**Note**

Assurez-vous de bien utiliser pour les arguments d'entrée `productName` et `productVersion` exactement le nom et la version d'un produit tel que retourné par `IncaGetInstalledProductInfo`.

4.2.6 Connaître la version de INCA

Nom	<code>IncaGetVersion</code>
Description	Cette fonction permet de connaître la version de INCA.
Syntaxe	<code>IncaGetVersion</code>
Arguments de sortie	<code>version</code> la version INCA comme une chaîne
Arguments d'entrée	<code>none</code>
Exemples	<code>IncaGetVersion;</code>

4.2.7 Lire les propriétés de INCA (INCA-MIP Evoluée)

Nom	<code>IncaGetProperties</code>
Description	Lit les propriétés de INCA
Syntaxe	<code>p = IncaGetProperties</code>

Arguments de sortie	<code>p</code>	Propriétés de INCA affichées en tant que structure MATLAB [®] , constituée des entrées suivantes: <ul style="list-style-type: none"> – <code>p.databasePath</code> - Chemin d'accès de la base de données INCA ouverte. S'il n'y a pas de base de données INCA ouverte une chaîne de caractère vide est renvoyée. – <code>p.dataPath</code> - Chemin d'accès du répertoire des données de INCA. – <code>p.installationPath</code> - Chemin d'accès du répertoire d'installation de INCA. – <code>p.tempPath</code> - Chemin d'accès du répertoire utilisé par les outils ETAS pour les fichiers temporaires.
Arguments d'entrée	<code>none</code>	
Exemples	<code>p = INCAGetProperties;</code>	

4.3 Initialisation

Toutes les opérations de mesure et de calibration dans INCA sont effectuées dans le cadre d'une expérimentation. Avant d'ouvrir une expérimentation, un espace de travail valide et une configuration matériel valide doivent être créés et assigné.

Pour travailler avec l'AddOn INCA-MIP, une expérimentation vide doit exister dans la base de données INCA pour laquelle un espace de travail valide et une configuration matérielle doivent d'abord être créés et attribués. Puis l'expérimentation peut être ouverte à partir de MATLAB[®].

Les fonctions API suivantes sont disponibles pour l'initialisation:

4.3.1 Ouvrir INCA

Nom	<code>IncaOpen</code>
Description	Ouvre INCA et initialise la communication entre INCA et MATLAB [®] .
Syntaxe	<code>IncaOpen</code> <code>IncaOpen(version)</code>

Arguments de sortie	none	
Arguments d'entrée	version	Version INCA à ouvrir (facultatif). Syntaxe : < MajorVersion>.<MinorVersion>. INCA-MIP pour INCA x.y peut uniquement se connecter aux installations INCA de la même version majeure x.INCA
Exemples	<pre>IncaOpen; IncaOpen('7.3');</pre>	

4.3.2 Fermer INCA (INCA-MIP Evoluée)

Nom	IncaClose	
Description	Se déconnecte par INCA et ferme INCA (optionnel) après s'être connecté avec succès à INCA avec IncaOpen.	
Syntaxe	<pre>IncaClose IncaClose(isDisconnectOnly)</pre>	
Arguments de sortie	none	
Arguments d'entrée	isDisconnectOnly	Spécifie si INCA seulement déconnecte par INCA ou si en plus il ferme MATLAB® (optionnel). Valeurs disponibles: – 0 : Déconnecte par INCA et ferme INCA (défaut). – 1 : Déconnecte par INCA et laisse INCA ouvert.
Exemples	<pre>INCAClose ; INCAIncClose(1);</pre>	

4.3.3 Ouvrir une base de données

Nom	IncaOpenDatabase	
Description	Cette fonction ouvre une base de données dans le répertoire spécifié.	
Syntaxe	<pre>IncaOpenDatabase({pathName})</pre>	

Arguments de sortie	none	
Arguments d'entrée	pathName	Le répertoire dans lequel la base de données est ouverte et enregistrée. Si aucun répertoire n'est spécifié, la base de données actuelle est ouverte.
Exemples	<pre>IncaOpenDatabase; % ouvrir la base de données actuelle IncaOpenDatabase ('c:\etasdata\madatabase');</pre>	

4.3.4 Importer une base de données (INCA-MIP Evoluée)

Nom	IncaDatabaseImport	
Description	Cette fonction importe le fichier export d'une base de données (*.exp) dans INCA. Les éléments de la base de données déjà existant seront toujours écrasés.	
Syntaxe	<pre>IncaDatabaseImport (path) nom = IncaDatabaseImport (path) [nom, type] = IncaDatabaseImport (path)</pre>	
Arguments de sortie	nom	Tableau de tous les chemins des éléments importés
		<p>Utilisez la fonction <code>deblank()</code> lorsque vous accéder à un tableau:</p> <pre>name2 = deblank(nom(2,:))</pre>

	<code>type</code>	<p>Tableau de tous les types des éléments importés</p> <p>Les valeurs possibles sont:</p> <ul style="list-style-type: none"> – <code>Folder</code>: Un répertoire de base de données – <code>Experiment</code>: Une expérimentation – <code>Workspace</code>: Un espace de travail – <code>Asap2Project</code>: Un projet A2L – <code>MeasurementCatalog</code>: Un catalogue de mesure ASAP2 – <code>CanDB</code>: Une base de données CAN ASAP2
		<p>Utilisez la fonction <code>deblank()</code> lorsque vous accéder à un tableau:</p> <pre>type2 = deblank(type(2,:))</pre>
Arguments d'entrée	<code>path</code>	Le chemin complet du fichier *.exp à importer
Exemples	<pre>noms = IncaDatabaseImport('D:\ETASData\INCA7.5\export\Project0815.exp')</pre>	

4.3.5 Lire les éléments de la base de données (INCA-MIP Evoluée)

Nom	<code>IncaBrowseItemsInFolder</code>	
Description	Cette fonction permet de lister les éléments de la base de données avec un filtre de recherche	
Syntaxe	<code>[nom, type] = IncaBrowseItemsInFolder(pattern, folderName)</code>	
Arguments de sortie	<code>nom</code>	Liste des noms des éléments de la base de données

	<code>type</code>	Liste des types des éléments de la base de données: <ul style="list-style-type: none"> – <code>Folder</code>: Répertoire – <code>Experiment</code>: Expérimentation – <code>Workspace</code>: Espace de travail – <code>Asap2Project</code>: Projet A2L ASAP2
Arguments d'entrée	<code>pattern</code>	Filtre de recherche à appliquer. Un '*' correspond à zéro, un ou plusieurs caractères supplémentaires. Un '#' correspond uniquement à un caractère. Tous les autres caractères doivent correspondre avec les éléments de la base de données. Il n'y a pas de différence entre les majuscules et les minuscules.
	<code>folderName</code>	Répertoire de la base de données où les éléments seront lus. L'arborescence des répertoires est séparée par un '\'. Un argument vide correspond à l'arborescence la plus haute.
Exemples	<pre>[n, t]=IncaBrowseItemsInFolder('**', 'DEFAULT\MyProject'); [nom, type]=IncaBrowseItemsInFolder('Prj*_##', '');</pre>	

4.3.6 Assigner un projet et un jeu de données à un dispositif (INCA-MIP Evoluée)

Nom	<code>IncaSetProjectAndDatasetInDevice</code>	
Description	Cette fonction permet d'assigner un projet et un jeu de données à un dispositif dans un espace de travail précis. Cela ne peut pas être fait si une expérimentation est ouverte.	
Syntaxe	<code>IncaSetProjectAndDatasetInDevice (workspace, dispositif, projet, dataset)</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>workspace</code>	Chemin d'accès de l'espace de travail
	<code>dispositif</code>	Nom du dispositif
	<code>projet</code>	Chemin d'accès du projet
	<code>dataset</code>	Chemin d'accès du jeu de donnée
Exemples	<code>IncaSetProjectAndDatasetInDevice ('DEFAULT\workspace', 'ETK:1', 'DEFAULT\Prj0815', 'Ds4711\Ds4711_3')</code>	

4.3.7 Ouvrir une expérimentation

Nom	<code>IncaOpenExperiment</code>	
Description	Cette fonction permet d'ouvrir l'expérimentation spécifiée. Après l'ouverture de l'expérimentation, vous pouvez utiliser l'AddOn INCA-MIP pour ajouter les variables de mesure et de calibration désirées.	
Syntaxe	<code>IncaOpenExperiment ({closeAllViewsFlag})</code> ou <code>IncaOpenExperiment (expFolderName, experimentName, workspaceFolderName, workspaceName {, closeAllViewsFlag})</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>expFolderName</code>	Répertoire dans lequel l'expérimentation est enregistrée

<code>experimentName</code>	Nom de l'expérimentation
<code>workspaceFolderName</code>	Répertoire dans lequel l'espace de travail est enregistré
<code>workspaceName</code>	Nom de l'espace de travail
<code>closeAllViewsFlag</code>	Ferme toutes les fenêtres des variables de mesure et de calibration dans l'expérimentation sélectionnée. Paramètre possible : <ul style="list-style-type: none"> — 1 : Ferme toute les fenêtres (par défaut) — 0 : Laisse les fenêtres inchangées

Exemples

```
IncaOpenExperiment('ExpFolder',
'MyExperiment', 'WorkspaceFolder',
'MyWorkspace');
```



Note

Si l'expérimentation est déjà ouverte lors de l'appel à la fonction `IncaOpenExperiment`, les arguments d'entrée spécifiant l'environnement sont optionnels.

Si l'expérimentation n'est pas encore ouverte, vous devez utiliser la fonction `IncaOpenDatabase` avant `IncaOpenExperiment`.

4.3.8 Réinitialiser une expérimentation

Nom	<code>IncaResetExperiment</code>
Description	Réinitialise et ferme l'expérimentation actuelle. Vous pouvez utiliser cette fonction pour retirer toutes les variables d'une expérimentation. Retirer des variables individuellement n'est actuellement pas possible.
Syntaxe	<code>IncaResetExperiment</code>
Arguments de sortie	none
Arguments d'entrée	none
Exemples	<code>IncaResetExperiment;</code>

**Note**

Si l'expérimentation a été ouverte manuellement et non une commande MATLAB®, `IncaResetExperiment` libère l'expérimentation sans la fermer. Dans ce cas vous devez utiliser la commande `IncaOpenExperiment` avant de pouvoir accéder de nouveau à l'expérimentation.

4.3.9 Lire les dispositifs (INCA-MIP Evoluée)

Nom	<code>IncaGetDevices</code>	
Description	Cette fonction permet de lire tous les dispositifs de l'expérimentation	
Syntaxe	<code>[nom, type] = IncaGetDevices</code>	
Arguments de sortie	<code>nom</code>	Liste des noms des dispositifs
	<code>type</code>	Liste des types des dispositifs: <ul style="list-style-type: none"> — <code>WorkbaseDevice</code>: Dispositif avec jeu de données — <code>MeasurementDevice</code>: Dispositif de mesure
Arguments d'entrée	<code>none</code>	
Exemples	<code>[nom,type]=IncaGetDevices;</code>	

4.3.10 Lire les propriétés du dispositif (INCA-MIP Evoluée)

Nom	<code>IncaGetDeviceProperties</code>	
Description	Cette fonction permet de lire les propriétés d'un dispositif	
Syntaxe	<code>p = IncaGetDeviceProperties(deviceName)</code>	
Arguments de sortie	<code>p</code>	Propriétés du dispositif sous la forme d'une structure MATLAB®, avec les entrées suivantes:
	<code>p.nom</code>	Nom du dispositif
	<code>p.descriptionFile</code>	Chemin d'accès du fichier de description du projet assigné au dispositif. Un texte vide est renvoyé s'il n'y a pas de projet assigné au dispositif.

<code>p.binaryFile</code>	Chemin d'accès du fichier de données assigné au dispositif. Un texte vide est renvoyé s'il n'y a pas de projet assigné au dispositif.
<code>p.projectDBPath</code>	Chemin d'accès dans la base de données INCA du projet assigné au dispositif. Un texte vide est renvoyé s'il n'y a pas de projet assigné au dispositif.
<code>p.isWriteProtected</code>	<ul style="list-style-type: none"> — 0 : Dispositif n'ayant pas de page mémoire ou utilisant une page non protégée en écriture — 1 : Page actuelle protégée en écriture
<code>p.isActive</code>	<ul style="list-style-type: none"> — 0 : Dispositif non connecté ou inactif — 1 : Dispositif connecté et actif
<code>p.isWorkbaseDevice</code>	<ul style="list-style-type: none"> — 0 : Dispositif sans jeu de données — 1 : Dispositif avec jeu de données
Arguments d'entrée	<code>deviceName</code> Nom du dispositif
Exemples	<code>p = IncaGetDeviceProperties('Device');</code>

4.4 Mesure et Enregistrement

Un signal ou une variable de mesure est toujours extrait/e comme faisant partie d'une fréquence d'échantillonnage de ce module de mesure uniquement. Chaque variable ne peut être affectée qu'à une fréquence d'échantillonnage. Pour configurer une expérimentation, attribuez d'abord les variables de mesure à une fréquence d'échantillonnage séparée.



Note

Les noms des éléments, des dispositifs, des signaux et des trames de mesure sont sensibles à la casse.

4.4.1 Lire les éléments de mesure (INCA-MIP Evoluée)

Nom	<code>IncaBrowseMeasureElements</code>	
Description	Cette fonction permet d'obtenir les éléments de mesure d'une expérimentation avec un filtre de recherche et éventuellement d'un dispositif	
Syntaxe	<pre>[nom, type] = IncaBrowseMeasureElements (pattern, {deviceName}) [nom] = IncaBrowseMeasureElements(pat- tern, {deviceName})</pre>	
Arguments de sortie	<code>nom</code>	Liste des noms des éléments de mesure
	<code>type</code>	Liste des types des éléments de mesure: <ul style="list-style-type: none"> – <code>Scalar</code>: Scalaire – <code>Array</code>: Vecteur – <code>Matrix</code>: Matrice
Arguments d'entrée	<code>pattern</code>	Filtre de recherche à appliquer sur les éléments de mesure. Un '*' correspond à zéro, un ou plusieurs caractères supplémentaires. Un '#' correspond uniquement à un caractère. Tous les autres caractères doivent correspondre avec les éléments de mesure. Il n'y a pas de différence entre les majuscules et les minuscules.
	<code>deviceName</code>	Nom du dispositif
Exemples	<pre>[n,t]=IncaBrowseMeasureElements('ign*', 'Device'); [nom,type]=IncaBrowseMeasureElements ('*');</pre>	

4.4.2 Lecture des fréquences d'échantillonnage (INCA-MIP Evoluée)

Nom	IncaGetMeasureRatesForDevice	
Description	Cette fonction permet d'obtenir toutes les fréquences d'échantillonnage d'un dispositif	
Syntaxe	<code>[nom] = IncaGetMeasureRatesForDevice(deviceName)</code>	
Arguments de sortie	<code>nom</code>	Liste des noms des fréquences d'échantillonnage
Arguments d'entrée	<code>deviceName</code>	Nom du dispositif
Exemples	<pre>n=IncaGetMeasureRatesForDevice('Device'); nom=IncaGetMeasureRatesForDevice('Dev');</pre>	

4.4.3 Ajouter une variable de mesure à une fréquence d'échantillonnage

Nom	IncaAddMeasureElement	
Description	Ajoute une variable de mesure à une expérimentation avec ou sans raster de mesure donné.	
Syntaxe	<pre>IncaAddMeasureElement(deviceName, groupName, signalName {, displayMode}) groupName = IncaAddMeasureElement(deviceName, [], signalName{, displayMode})</pre>	
Arguments de sortie	none	
Arguments d'entrée	<code>deviceName</code>	Nom du dispositif

groupName	<p>nom du raster de mesure</p> <p>Il est possible d'utiliser plusieurs rasters en combinant tout simplement les noms de raster au moyen d'un caractère « + », p. ex. '10 ms+100 ms'. L'utilisation d'un tel raster multiple crée un nouveau raster virtuel.</p> <p>Chaque signal peut uniquement être mesuré dans exactement un raster ou un raster multiple.</p> <p>Le nom de groupe peut être [] (voir remarque ci-dessous).</p>
signalName	<p>Nom du signal de mesure. Pour les scalaires, le nom est suffisant. Pour les vecteurs et matrices, l'index au format [n] ou [n,m] doit être accolé au nom. Le premier élément a l'index "zéro".</p>
displayMode	<p>Mode d'affichage de l'élément:</p> <ul style="list-style-type: none"> — 1 : la variable de mesure est affichée (par défaut) — 0 : pas d'affichage

Exemples

```
IncaAddMeasureElement('MyDevice', '10ms',
'Channel01', 0);
IncaAddMeasureElement('ETK:1', '1.0ms',
'Matrix[2,1]');
group = IncaAddMeasureElement('CalcDev',
[], 'MyCalcSig1');
```

**Note**

Si la fréquence d'échantillonnage est pleine, la variable de mesure n'est pas ajoutée à cette fréquence.

Note

Si l'argument d'entrée `groupName` est [] (c'est-à-dire vide), le groupe de signaux sera déterminé de la manière suivante :

- Si le signal fait déjà partie de l'expérimentation, c'est son nom de groupe de signaux existant qui est utilisé.
- Si le signal ne fait pas partie de l'expérimentation, un groupe de signaux disponible est utilisé de manière arbitraire. Dans le cas du dispositif Calculé (DispCalc, utilisé pour les signaux calculés) ou de CAN Monitoring, c'est le groupe de signaux défini pour ce signal qui est utilisé.

Comme le nom du groupe de signaux est requis pour `IncaGetRecords`, `IncaGetRecordStruct` ou `IncaGetRecordCount`, il est retourné en tant que paramètre de gauche en option.

Exemples :

```
groupName = IncaAddMeasureElement( 'DispCalc', [],
'MonSigCalc')
groupName = IncaAddMeasureElement( 'CAN-Monitoring:1',
[], 'nmot', 1)
```

Note

Le nombre total de signaux qui peuvent être ajoutés est spécifique à l'appareil ainsi qu'au protocole. Le nombre de signaux est limité par la quantité de mémoire tampon libre allouée par le processus du serveur cible. La taille totale de la mémoire tampon dépend de la fréquence d'échantillonnage utilisée.

Exemple :

Un signal avec une fréquence d'échantillonnage de 0,1 ms nécessite > 3 Mo de données. Par conséquent, le nombre total de signaux qui peuvent être ajoutés se situe entre 400 et 600 signaux. Des fréquences d'échantillonnage plus lentes permettent d'ajouter plus de signaux.

4.4.4 Démarrer la mesure

Nom	<code>IncaStartMeasurement</code>
Description	Démarre la mesure sous INCA
Syntaxe	<code>IncaStartMeasurement</code>

Arguments de sortie	none
Arguments d'entrée	none
Exemples	<code>IncaStartMeasurement;</code>

4.4.5 Arrêter la mesure

Nom	<code>IncaStopMeasurement</code>	
Description	Cette fonction arrête la mesure en court et l'enregistrement (s'il y a lieu) sous INCA	
Syntaxe	<code>IncaStopMeasurement { (mdfFileName) }</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>mdfFileName</code>	Nom du fichier MDF dans lequel les données enregistrées sont sauvegardées si l'enregistrement s'arrête en même temps que la mesure en cours. Il faut toujours spécifier en entier le chemin d'accès au fichier (ex. 'c:\mydata\store1.dat').
Exemples	<code>IncaStopMeasurement ('c:\mydata\store1.dat')</code>	



Note

Pour éviter de perdre des données du fait des mesures en cours, il faut s'assurer d'arrêter l'enregistrement en cours avec la commande `IncaStopMeasurement (mdfFileName)` si le volume de données est élevé. Ensuite il faut utiliser la commande `IncaGetRecords` pour transférer à MATLAB® les données restantes.

4.4.6 Lire les propriétés de l'enregistrement (INCA-MIP Evoluée)

Nom	<code>IncaGetRecordingProperties</code>	
Description	Lit les propriétés du fichier de sortie primaire de l'enregistreur par défaut ainsi que l'extension de fichier pour le format d'enregistrement primaire sélectionné.	
Syntaxe	<code>properties = IncaGetRecordingProperties</code>	

Arguments de sortie	<code>properties</code>	Propriétés de l'enregistrement sous la forme d'une structure MATLAB [®] , avec les entrées suivantes:
	<code>properties.fileName</code>	Le nom du fichier d'enregistrement y compris le format du fichier de sortie primaire
	<code>properties.directory</code>	Répertoire du fichier d'enregistrement
	<code>properties.fileFormat</code>	Format du fichier d'enregistrement; les informations textes suivantes sont possibles: <ul style="list-style-type: none"> — ETASBinary — #DiademATF — ETASAscii — ETASGroupAscii — ETASMATLABMFILE — ETASGroupMatlabM — FamosRecord — ETASMDF — ETASMDF4
	<code>properties.autoIncrement</code>	Incrément automatique du nom du fichier d'enregistrement <ul style="list-style-type: none"> — 0 : Pas d'incréméntation automatique — 1 : Incréméntation automatique

<code>properties.comment</code>	Commentaire de l'entête du fichier d'enregistrement. Il ne doit pas excéder 1024 caractères moins les caractères utilisés pour les commentaires par défauts.
<code>properties.defaultComment</code>	Commentaire par défaut généré par INCA de l'entête du fichier d'enregistrement
<code>properties.company</code>	Nom de la société dans l'entête du fichier d'enregistrement
<code>properties.project</code>	Nom du projet dans l'entête du fichier d'enregistrement
<code>properties.user</code>	Nom de l'utilisateur dans l'entête du fichier d'enregistrement
<code>properties.vehicle</code>	Nom du véhicule dans l'entête du fichier d'enregistrement
Arguments d'entrée	none
Exemples	<code>properties = IncaGetRecordingProperties;</code>

4.4.7 Régler les propriétés de l'enregistrement (INCA-MIP Evoluée)

Nom	<code>IncaSetRecordingProperties</code>
Description	Cette fonction permet de régler les propriétés du prochain enregistrement
Syntaxe	<code>IncaSetRecordingProperties(properties)</code>
Arguments de sortie	none

Arguments d'entrée	<code>properties</code>	Propriétés de l'enregistrement sous la forme d'une structure MATLAB, avec les entrées suivantes:
	<code>properties.fileName</code>	Nom du fichier d'enregistrement
	<code>properties.directory</code>	Répertoire du fichier d'enregistrement
	<code>properties.fileFormat</code>	Format du fichier d'enregistrement; les informations textes suivantes sont possibles: <ul style="list-style-type: none"> — <code>ETASBinary</code> — <code>#DiademATF</code> — <code>ETASAscii</code> — <code>ETASGroupAscii</code> — <code>ETASMATLABMFILE</code> — <code>ETASGroupMatlabM</code> — <code>FamosRecord</code> — <code>ETASMDF</code> — <code>ETASMDF4</code>
	<code>properties.autoIncrement</code>	Incrément automatique du nom du fichier d'enregistrement <ul style="list-style-type: none"> — <code>0</code> : Pas d'incréméntation automatique — <code>1</code> : Incréméntation automatique
	<code>properties.comment</code>	Il ne doit pas excéder 1024 caractères moins les caractères utilisés pour les commentaires par défauts.
	<code>properties.company</code>	Commentaire par défaut généré par INCA de l'entête du fichier d'enregistrement

<code>properties.project</code>	Nom du projet dans l'entête du fichier d'enregistrement
<code>properties.user</code>	Nom de l'utilisateur dans l'entête du fichier d'enregistrement
<code>properties.vehicle</code>	Nom du véhicule dans l'entête du fichier d'enregistrement

Exemples

```
properties.user = 'Michael';
properties.project = 'K70';
IncaSetRecordingProperties(properties);
```



Note

Lorsque vous définissez l'enregistrement avec `IncaSetRecordingProperties`, vous ne devez pas terminer un enregistrement avec `IncaStopRecording`. Au lieu de cela, il est possible d'utiliser `IncaSetTrigger`. Vous pouvez utiliser aucune condition de trigger pour finir l'enregistrement.

Exemple:

Finir l'enregistrement après une durée fixe:

```
TIMEDURATION_SECONDS = 25 ;
IncaSetTrigger('none', 'none', 'none', 'none',
TIMEDURATION_SECONDS) ;
IncaStartRecording ;
```

% L'enregistrement s'arrête automatiquement après
TIMEDURATION_SECONDS secondes

Finir l'enregistrement après un trigger manuel:

```
IncaSetTrigger('none', 'manual');
IncaStartRecording;
% Do anything until the stop trigger condition is met
...
IncaExecuteManualTrigger('stop');
```

4.4.8 Mode Lecture Enregistrement (INCA-MIP Evoluée)

Nom	IncaGetRecordingMode	
Description	Indique si un signal est enregistré ou non dans l'enregistreur par défaut.	
Syntaxe	IncaGetRecordingMode (deviceName, signalName)	
Arguments de sortie		<p>mode d'enregistrement pour l'enregistreur par défaut :</p> <ul style="list-style-type: none"> – 0 : le signal n'est pas enregistré dans l'enregistreur par défaut – 1 : le signal est enregistré dans l'enregistreur par défaut
Arguments d'entrée	deviceName	Nom du dispositif
	signalName	Nom du signal de mesure. Pour les scalaires, le nom est suffisant. Pour les vecteurs et matrices, l'index au format [n] ou [n,m] doit être accolé au nom. Le premier élément a l'index "zéro".
Exemples	<pre>m = IncaGetRecordingMode('ETK:1', 'hfm'); mode = IncaGetRecordingMode('CalcDev', 'MyCalcSig1');</pre>	



Note

Avant d'utiliser IncaSetRecordingMode, le signal doit être ajouté avec IncaAddMeasureElement.

4.4.9 Set Recording Mode (INCA-MIP Evoluée)

Nom	<code>IncaSetRecordingMode</code>	
Description	Active ou désactive l'enregistrement d'un signal dans l'enregistreur par défaut. L'enregistrement peut uniquement être désactivé pour des signaux qui sont affichés dans l'expérimentation INCA. Avant d'exécuter cette commande, le signal doit être ajouté à l'expérimentation avec <code>IncaAddMeasureSignal</code> .	
Syntaxe	<code>IncaSetRecordingMode(deviceName, signalName, recordingMode)</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>deviceName</code>	Nom du dispositif
	<code>signalName</code>	Nom du signal de mesure. Pour les scalaires, le nom est suffisant. Pour les vecteurs et matrices, l'index au format <code>[n]</code> ou <code>[n,m]</code> doit être accolé au nom. Le premier élément a l'index "zéro".
	<code>recordingMode</code>	mode d'enregistrement pour l'enregistreur par défaut : <ul style="list-style-type: none"> – 0 : la variable de mesure est supprimée dans l'enregistreur par défaut – 1 : le signal de mesure est ajouté à l'enregistreur par défaut
Exemples	<pre>IncaSetRecordingMode('ETK:1', 'hfm', 1); IncaSetRecordingMode('DispCalc', 'MonSigCalc1', 0);</pre>	



Note

Avant d'utiliser `IncaSetRecordingMode`, le signal doit être ajouté avec `IncaAddMeasureElement`.

4.4.10 Démarre l'enregistrement

Nom	<code>IncaStartRecording</code>
Description	Cette fonction permet de démarrer l'enregistrement dans INCA. Cette fonction peut être utilisée après ou à la place de la fonction <code>IncaStartMeasurement</code> . Une fois une mesure ou un enregistrement lancé/e, les données mesurées sont également disponibles dans MATLAB®.
Syntaxe	<code>IncaStartRecording</code>
Arguments de sortie	none
Arguments d'entrée	none
Exemples	<code>IncaStartRecording;</code>

4.4.11 Arrêter un enregistrement

Nom	<code>IncaStopRecording</code>
Description	Cette fonction permet d'arrêter l'enregistrement en cours dans INCA. La mesure continue et doit être arrêtée au moyen de la fonction <code>IncaStopMeasurement</code> . Il est possible d'activer/désactiver cette fonction plusieurs fois au cours de la mesure.
Syntaxe	<code>IncaStopRecording(mdfFileName)</code>
Arguments de sortie	none
Arguments d'entrée	<code>mdfFileName</code> Nom du fichier MDF où les données sont enregistrées. Il faut toujours spécifier le chemin d'accès entier du fichier. (ex. <code>'c:\mydata\store1.dat'</code>).
Exemples	<code>IncaStopRecording('c:\mydata\store1.dat');</code>

4.4.12 Régler le mode de lecture des données (Données Online / Offline)

Nom	<code>IncaSetMeasureReadMode</code>	
Description	<p>Cette fonction API permet de définir la source à partir de laquelle les données mesurées sont transmises à MATLAB®. Les données sont, soit préparées dans INCA puis transférées à MATLAB® (données offline), soit sont lues à partir du buffer circulaire (données online). Pour certains modules, tels que la carte ES1303 et les modules de série ES6xx, aucune donnée offline n'est disponible pendant l'affichage des données mesurées. Il est donc recommandé d'utiliser les données online pendant l'affichage.</p> <p>Lors de l'enregistrement des données mesurées, les données online ou offline sont utilisables. Les données online peuvent être incomplètes au moment des chargements importants, alors que les données offline sont toujours complètes lorsque les données mesurées sont enregistrées. Cependant, les données offline peuvent être transmises avec un certain délai lors d'importants chargements. Il est recommandé d'utiliser uniquement les données offline lors de l'enregistrement des données mesurées.</p>	
Syntaxe	<code>IncaSetMeasureReadMode (measureReadMode)</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>measureReadMode</code>	Paramètre numérique dont la valeur spécifie la source de données. Paramètres possibles : <ul style="list-style-type: none"> — 1 : Données offline — 0 : Données online (par défaut)
Exemples	<code>IncaSetMeasureReadMode (0) ;</code>	

4.4.13 Lecture des données de mesure

Nom	IncaGetRecords	
Description	<p>Transfère les données de mesure vers MATLAB®. Les données de mesure de chaque groupe de signaux sont archivées dans un buffer circulaire dédié pouvant contenir jusqu'à 30 secondes de temps de mesure. Les données de mesure sont récupérées par groupes dans MATLAB®. Il faudrait donc arrêter l'exécution de votre script dans MATLAB® après avoir récupéré les données de mesure. Plus la quantité de données transférées est importante à chaque fois, plus le transfert de données est efficace.</p> <p>Cette fonction transmet un nombre spécifié d'enregistrements pour le raster de mesure spécifié. Pour de plus amples informations sur le buffer circulaire, consultez l'entrée correspondante dans le "INCA Définitions" à la page 10.</p>	
Syntaxe	<pre>[time, data {,state}] = IncaGetRecords (deviceName, groupName, maxRecords {{,la- test{, exact}}})</pre>	
Arguments de sortie	time	Un vecteur contenant les tampons d'horodatage des enregistrements transférés. Cette variable contient un nombre maximal de valeurs m , tandis que $m \leq \text{maxRecords}$.
	data	Une matrice bidimensionnelle contenant les valeurs de données pour chaque variable de mesure dans l'ordre où elle a été ajoutée à l'expérimentation par <code>IncaAddMeasureElement</code> . Dans cette matrice, la dimension m reflète le nombre d'enregistrements transférés, tandis que n indique le nombre de raster de mesure.

	state	<p>Paramètre en retour optionnel :</p> <ul style="list-style-type: none"> – 0 : succès. Enregistrements reçus – 1: acquisition non opérationnelle. Aucun enregistrement reçu – 2: enregistrements insuffisants. Aucun enregistrement reçu. Ne peut être retourné que si <code>exact = 1</code>
Arguments d'entrée	deviceName	Nom du dispositif
	groupName	<p>Nom du raster de mesure</p> <p>Il est possible d'utiliser plusieurs rasters en combinant tout simplement les noms de raster au moyen d'un caractère « + », p. ex. '10 ms+100 ms'. L'utilisation d'un tel raster multiple crée un nouveau raster virtuel. Chaque signal peut uniquement être mesuré dans exactement un raster ou un raster multiple.</p>
	maxRecords	<p>Nombre maximal d'enregistrements à transférer. Voir aussi le paramètre <code>exact</code>. Ce nombre correspond à la dimension <code>m</code> pour la variable de temps ou de données dans les valeurs retournées. Si la dimension dans les valeurs retournées atteint la valeur <code>maxRecords</code>, les enregistrements ne seront pas tous lus et un dépassement du buffer circulaire peut survenir.</p>

latest	Définit si les enregistrements les plus anciens ou les plus récents seront reçus
exact	Définit de recevoir des enregistrements même si $n < \text{maxRecords}$ sont présents dans le buffer circulaire ou définit si le buffer circulaire doit rester inchangé.

Exemples

```
[t, d] = IncaGetRecords('ETK:1', '100ms',
500);
data = [data; d];
time = [time; t];
[t, d, s] = IncaGetRecords('ETK:1',
'100ms', 25, 1, 1);
```

Pour un contexte plus large utilisant cette partie de code, reportez-vous à ["Exemple 2" à la page 72](#).



Note

La raster utilisée dans `IncaGetRecords` correspond directement à la raster utilisée dans `IncaAddMeasureElement`, à savoir vous devez utiliser la même raster ou une raster multiple.

Exemple:

```
IncaAddMeasureElement('ETK test device:1', 'RASTER_
A+RASTER_B', 'N')
IncaAddMeasureElement('ETK test device:1', 'RASTER_
A+RASTER_B', 'n')
[t,d]= IncaGetRecords('ETK test device:1', 'RASTER_
A+RASTER_B', 15)
```

Pour contrôler les affectations de raster de signaux, vous pouvez utiliser la commande `IncaGetRecordStruct`.

Exemple:

```
l=IncaGetRecordStruct('ETK test device:1', 'RASTER_
A+RASTER_B')
```

**Note**

Les combinaisons de paramètres suivantes s'exécutent comme suit :

- `latest = 0, exact = 0` : (par défaut)
Retourne les enregistrements les plus anciens jusqu'à `maxRecords` depuis le buffer circulaire. Les enregistrements plus récents restent inchangés.
- `latest = 1, exact = 0` :
Retourne les enregistrements les plus récents jusqu'à `maxRecords` depuis le buffer circulaire. Tous les enregistrements plus anciens sont délibérément rejetés.
- `latest = 0, exact = 1` :
Retourne les `maxRecords` plus anciens enregistrements depuis le buffer circulaire. Les enregistrements plus récents restent inchangés. Si seulement $n < \text{maxRecords}$ enregistrements sont disponibles dans le buffer circulaire, rien n'est reçu.
- `latest = 1, exact = 1` :
Retourne les `maxRecords` plus récents enregistrements depuis le buffer circulaire. Tous les enregistrements plus anciens sont délibérément rejetés. Si seulement $n < \text{maxRecords}$ enregistrements sont disponibles dans le buffer circulaire, rien n'est reçu.

4.4.14 Réinitialiser les buffers tournant

Nom	<code>IncaResetRecords</code>
Description	<p>Cette fonction réinitialise le buffer circulaire de toutes les fréquences d'échantillonnage. Elle est utilisable même en cours de mesure pour réinitialiser tous les buffers tournant. Les buffers tournant sont réinitialisés automatiquement lors du lancement de la mesure ou de l'enregistrement. Il n'est donc pas nécessaire de spécifier explicitement l'exécution de cette commande.</p> <p>Pour de plus amples informations sur le buffer circulaire, consultez l'entrée correspondante dans le "INCA Définitions" à la page 10.</p>
Syntaxe	<code>IncaResetRecords</code>
Arguments de sortie	none
Arguments d'entrée	none
Exemples	<code>IncaResetRecords;</code>

4.4.15 Lecture de l'état du matériel (INCA-MIP Evoluée)

Nom	<code>IncaGetHardwareStatus</code>	
Description	Cette fonction permet de connaître l'état du matériel pendant une mesure ou un enregistrement	
Syntaxe	<code>[status, message] = IncaGetHardwareStatus</code>	
Arguments de sortie	<code>status</code>	Etat actuel du matériel — 0 : OK — 1 : Warning — 2 : Erreur
	<code>message</code>	Si <code>status</code> renvoie la valeur 1 ou 2, la valeur <code>message</code> renvoie un texte décrivant le warning ou l'erreur.
		Si <code>status</code> renvoie la valeur 1 ou 2, la mesure ou l'enregistrement doivent être arrêtés avant un autre appel de la fonction <code>IncaGetHardwareStatus</code> .
		Voir aussi le script exemple dans le fichier <code>tHWStatus.m</code> au " Connaître l'AddOn INCA-MIP grâce à des exemples: " à la page 19
Arguments d'entrée	<code>none</code>	
Exemples	<code>[s, m] = IncaGetHardwareStatus;</code>	

4.4.16 Définir les triggers (INCA-MIP Evoluée)

Nom	<code>IncaSetTrigger</code>	
Description	Cette fonction permet de définir la condition de trigger avec le démarrage d'une mesure ou d'un enregistrement avec les fonctions <code>IncaStartMeasurement</code> ou <code>IncaStartRecording</code>	
Syntaxe	<code>IncaSetTrigger(startTrigger{, stopTrigger{, preTriggerTime{, postTriggerTime{, duration}}})</code>	
Arguments de sortie	<code>none</code>	

Arguments d'entrée	<code>startTrigger</code>	Trigger de début. <ul style="list-style-type: none"> – <code>manual</code> pour un trigger manuel – <code>none</code> s'il n'y a pas de trigger
	<code>stopTrigger</code>	Trigger de fin. <ul style="list-style-type: none"> – <code>manual</code> pour un trigger manuel – <code>none</code> s'il n'y a pas de trigger (par défaut)
	<code>preTriggerTime</code>	Durée de pré-trigger en seconde <ul style="list-style-type: none"> – <code>none</code> si non spécifié (par défaut)
	<code>postTriggerTime</code>	Durée de post-trigger en seconde <ul style="list-style-type: none"> – <code>none</code> si non spécifié (par défaut)
	<code>duration</code>	Durée de l'enregistrement de la mesure ou de l'enregistrement en seconde <ul style="list-style-type: none"> – <code>none</code> si non spécifié (par défaut) dans ce cas, la durée est infinie.
Exemples	<pre>IncaSetTrigger('nmot\ETK:1 > 2000', 'none', 2.0, 3.0) IncaSetTrigger('none', 'none', 'none', 'none', 360)</pre>	

Le tableau suivant énumère toutes les combinaisons de paramètres d'entrée qui sont prises en charge (les autres combinaisons entraînent une exception).

Combinaisons de paramètres d'entrée

Fonctionnalité trigger	startTrigger	stopTrigger	preTriggerTime	postTriggerTime	durée
Enregistrement avec durée de temps	'none'	'none'	'none'	'none'	value
Enregistrement avec trigger de départ manuel, temps de pré-trigger et condition de trigger d'arrêt manuel	'manual'	'manual'	value	'none'	'none'
Enregistrement avec trigger de départ manuel, temps de pré- et post-trigger	'manual'	'none'	value	value	'none'
Enregistrement avec trigger de départ manuel, temps de pré-trigger et condition de trigger d'arrêt	'manual'	value	value	'none'	'none'
Enregistrement avec condition de trigger d'arrêt manuel	'none'	'manual'	'none'	'none'	'none'
Enregistrement avec condition de trigger de départ et durée d'enregistrement	value	'none'	'none'	'none'	value
Enregistrement avec condition de trigger de départ et trigger d'arrêt manuel	value	'manual'	'none'	'none'	'none'

Fonctionnalité trigger	startTrigger	stopTrigger	preTriggerTime	postTriggerTime	durée
Enregistrement avec condition de trigger de départ, temps de pré-trigger et trigger d'arrêt manuel	value	'manual'	value	'none'	'none'
Enregistrement avec condition de trigger de départ, temps de pré- et post-trigger	value	'none'	value	value	'none'
Enregistrement avec condition de trigger de départ, temps de pré-trigger et condition de trigger d'arrêt	value	value	value	'none'	'none'

4.4.17 Execution du trigger manuel (INCA-MIP Evoluée)

Nom	<code>IncaExecuteManualTrigger</code>	
Description	Cette commande permet de déclencher le trigger manuel de début ou de fin. Cette fonction n'a d'effet que si la commande <code>IncaSetTrigger</code> a été utiliser avant avec les paramètres <code>startTrigger</code> OU <code>stopTrigger</code> régler sur <code>manual</code> (manuel).	
Syntaxe	<code>IncaExecuteManualTrigger (type)</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>type</code>	Type de trigger <ul style="list-style-type: none"> – <code>start</code> pour déclencher le trigger manuel de début – <code>stop</code> pour déclencher le trigger manuel de fin
Exemples	<code>IncaExecuteManualTrigger ('start')</code>	

4.4.18 Lire l'état d'un enregistrement (INCA-MIP Evoluée)

Nom	<code>IncaGetRecordingState</code>	
Description	Cette fonction permet d'obtenir l'état de l'enregistrement en cours.	
Syntaxe	<code>result = IncaGetRecordingState</code>	
Arguments de sortie	<code>result</code>	Etats de l'enregistrement <ul style="list-style-type: none"> – 0 : non actif – 1 : en attente de trigger ou enregistrement en cours
Arguments d'entrée	none	
Exemples	<code>s = IncaGetRecordingState</code>	

4.4.19 Lire la liste des variables de mesure (INCA-MIP Evoluée)

Nom	<code>IncaGetRecordStruct</code>	
Description	Cette fonction permet de connaître la liste des variables de mesure de la mesure ou de l'enregistrement. La liste renvoyée contient les noms des variables de mesure dans le même ordre où elles ont été appelées par la commande <code>IncaAddMeasurementElement</code> .	
Syntaxe	<code>list = IncaGetRecordStruct(device, groupName)</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>device</code>	Nom du dispositif
	<code>groupName</code>	Nom de la fréquence d'échantillonnage Il est possible d'utiliser plusieurs rasters en combinant tout simplement les noms de raster au moyen d'un caractère « + », p. ex. '10 ms+100 ms'.
Exemples	<pre>l = IncaGetRecordStruct('ETK:1', '10ms'); list = IncaGetRecordStruct('device1', 'Syncro');</pre>	

4.5 Calibration

La calibration peut se faire sur des scalaires, courbes et cartographies, notamment des distributions de points d'interruption. Dans chaque expérimentation, il est possible de définir autant de variables de calibration qu'on le souhaite.



Note

Notez que le nom des variables de calibration est sensible à la casse.

4.5.1 Lire les éléments de calibration (INCA-MIP Evoluée)

Nom	IncaBrowseCalibrationElements	
Description	Cette fonction permet d'obtenir les éléments de calibration de l'expérimentation avec un filtre de recherche et éventuellement d'un dispositif.	
Syntaxe	<pre>[nom, type] = IncaBrowseCalibrationElements (pattern, {deviceName}) nom = IncaBrowseCalibrationElements (pattern, {deviceName})</pre>	
Arguments de sortie	nom	Liste des noms des éléments de calibration.
	type	Liste de types de calibration: <ul style="list-style-type: none"> – Distribution: Axie de distribution – OneDTable: Courbe – TwoDTable: Cartographie – Scalar: Scalaire – Array: Vecteur – Matrix: Matrice
Arguments d'entrée	pattern	Filtre de recherche à appliquer sur les éléments de mesure. Un '*' correspond à zéro, un ou plusieurs caractères supplémentaires. Un '#' correspond uniquement à un caractère. Tous les autres caractères doivent correspondre avec les éléments de mesure. Il n'y a pas de différence entre les majuscules et les minuscules.
Exemples	deviceName	Nom du dispositif
	<pre>[n,t]=IncaBrowseCalibrationElements ('MAP*', 'Device'); [nom,type]=IncaBrowseCalibrationElements ('*');</pre>	

4.5.2 Ajouter un élément de calibration

Nom	<code>IncaAddCalibrationElement</code>	
Description	Ajoute une variable de calibration à l'expérimentation en cours. Les calibrations peuvent être effectuées avec des scalaires, des courbes caractéristiques et des cartes englobant les distributions de points d'axe associées. Dans chaque expérimentation, il est possible de définir un nombre quelconque de variables de calibration. Cette commande prend également en charge les distributions de points d'axe et les distributions de points d'axe groupés.	
Syntaxe	<code>IncaAddCalibrationElement (deviceName, calibrationName {, displayMode})</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>deviceName</code>	Nom du dispositif
	<code>calibrationName</code>	Nom de la calibration
	<code>displayMode</code>	Mode d'affichage de l'élément: <ul style="list-style-type: none"> — 2 : La variable de calibration est affichée et constamment rafraîchie (par défaut) — 1 : La variable de calibration est affichée mais non rafraîchie — 0 : La variable de calibration n'est pas affichée <p>Sélectionner 1 (affichage uniquement) peut améliorer considérablement la performance pour de grosse quantité de données.</p>
Exemples	<pre>IncaAddCalibrationElement ('anEtk', 'Scalar'); IncaAddCalibrationElement ('anEtk', 'Curve'); IncaAddCalibrationElement ('anEtk', 'Map');</pre>	

**Note**

Pour les axes groupés, aucune interpolation des courbes et des cartes dépendantes n'est exécutée.

4.5.3 Lire la valeur de calibration

Nom	<code>IncaGetCalibrationValue</code>	
Description	Cette fonction permet de lire la valeur actuelle de la variable de calibration ou des distributions de points d'interruption	
Syntaxe	<code>value = IncaGetCalibrationValue(deviceName, calibrationName {, start, size} {, valueType})</code>	
Arguments de sortie	<code>value</code>	La valeur actuelle de la calibration; elle doit correspondre aux types de données indiqués ci-dessous: <ul style="list-style-type: none"> – Scalaires: une matrice (1,1) – Courbes: une matrice (x,1) – Cartographies: une matrice (x,y) – Distributions de points d'interruption: une matrice (x,1)
Arguments d'entrée	<code>deviceName</code>	Nom du dispositif
	<code>calibrationName</code>	Nom d'élément de calibration
	<code>start</code>	Index de départ. Types de données supportées: <ul style="list-style-type: none"> – Pour les courbes et distributions de points d'interruption un index de départ <code>x</code> doit être spécifié. <code>x >= 1</code> – Pour les cartographies, un index de départ <code>[x, y]</code> doit être spécifié. <code>x, y >= 1</code>

size	<p>Nombre de valeur à lire. Types de données supportées:</p> <ul style="list-style-type: none"> – Pour les courbes et les distributions de points de rupture, un nombre n doit être spécifié. $n \geq 1$ – Pour les cartographies, un nombre $[n, m]$ doit être spécifié. $n, m \geq 1$
valueType	<p>Sélection de l'argument de sortie (texte). La fonction renvoie soit la valeur de la variable de calibration (mode par défaut), soit la distribution du point d'appui en X ou Y.</p> <p>Paramètres possibles :</p> <ul style="list-style-type: none"> – v: la valeur – x: renvoie le point d'appui en X (courbes et cartographies) – y: renvoie le point d'appui en Y (cartographies)

Exemples

```

aValue = IncaGetCalibrationValue('anEtk',
'Scalar');
aCurve = IncaGetCalibrationValue('anEtk',
'Curve');
aMap = IncaGetCalibrationValue('anEtk',
'Map');
xMap = IncaGetCalibrationValue('anEtk',
'Map', 'x');
yMap = IncaGetCalibrationValue('anEtk',
'Map', 'y');
aCurveRange = IncaGetCalibrationValue
('anEtk', 'Curve', 2, 3);
aMapRange = IncaGetCalibrationValue
('anEtk', 'Map', [2,3], [3,4]);
xMapRange = IncaGetCalibrationValue
('anEtk', 'Map', 2, 3, 'x');

```

4.5.4 Modifier une valeur de calibration

Nom	<code>IncaSetCalibrationValue</code>
Description	Cette fonction permet d'attribuer une valeur à une variable de calibration ou à un point d'interruption (d'appui).
Syntaxe	<pre> IncaSetCalibrationValue(deviceName, cali- brationName, value) IncaSetCalibrationValue(deviceName, cali- brationName, value, valueType) IncaSetCalibrationValue(deviceName, cali- brationName, value, start) IncaSetCalibrationValue(deviceName, cali- brationName, value, start, valueType) result = IncaSetCalibrationValue(devi- ceName, calibrationName, value) result = IncaSetCalibrationValue(devi- ceName, calibrationName, value, valueType) result = IncaSetCalibrationValue(devi- ceName, calibrationName, value, start) result = IncaSetCalibrationValue(devi- ceName, calibrationName, value, start, valueType) </pre>

Arguments de sortie	<code>result</code>	<p>Résultat de la calibration (facultatif, uniquement en cas de défaillances)</p> <p>Si aucun bit de résultat n'est posé, c'est que la valeur de calibration a été modifiée avec succès. C'est également le cas si l'un des bits 5 à 8 est posé, ce qui fournit des informations supplémentaires.</p> <p>Si toutefois, l'un des bits 0 à 4 est posé, c'est que la calibration a échoué.</p> <ul style="list-style-type: none"> — Bit 0 actif: Calibration non faite — Bit 1 actif: Limite logicielle basse dépassée — Bit 2 actif: Limite logicielle haute dépassée — Bit 3 actif: Limite matérielle basse dépassée — Bit 4 actif: Limite matérielle haute dépassée — Bit 5 actif: Valeur saturée à la limite logicielle basse — Bit 6 actif: Valeur saturée à la limite logicielle haute — Bit 7 actif: Valeur saturée à la limite matérielle basse — Bit 8 actif: Valeur saturée à la limite matérielle haute <p>Il peut y avoir plusieurs raisons pour qu'une calibration ne soit pas faite. Par exemple en fonction du mode de calibration actif, si une des limite a été dépassée. Dans ce cas il y a plus d'information renvoyée par les bits 1 à 4. Une autre raisons peut être que l'élément de calibration ou la page active est protégé en écriture ou que les points</p>
----------------------------	---------------------	--

Arguments d'entrée	<code>deviceName</code>	d'appui x ou y ne sont pas monotones. Dans tous ces cas, seul le bit 0 est activé. Nom du dispositif
	<code>calibrationName</code>	Nom de l'élément de calibration
	<code>value</code>	La valeur actuelle de la calibration. Elle doit correspondre aux types de données indiqués ci-dessous: <ul style="list-style-type: none"> – Scalaires: une matrice (1,1) – Courbes: une matrice (x,1) – Cartographies: une matrice (x,y) – Distributions de points d'interruption (d'appui) : une matrice (x,1)

start	<p>Index de départ. Types de données supportées:</p> <ul style="list-style-type: none"> – Pour les courbes et distributions de points d'interruption un index de départ x doit être spécifié. $x \geq 1$ – Pour les cartographies, un index de départ $[x, y]$ doit être spécifié. $x, y \geq 1$
valueType	<p>Sélection de la valeur de retour (texte). La fonction renvoie soit la valeur de la variable de calibration (par défaut) soit la distribution X ou Y de points d'interruption (point d'appui). Paramètres possibles :</p> <ul style="list-style-type: none"> – v: valeur – x: point d'appui en x (courbe et cartographies) – y: point d'appui en y (cartographie)

Exemples

```

IncaSetCalibrationValue('anEtk',
'Scalar', aValue);
IncaSetCalibrationValue('anEtk', 'Curve',
aCurve);
IncaSetCalibrationValue('anEtk', 'Map',
aMap);
IncaSetCalibrationValue('anEtk', 'Map',
xMap, 'x');
IncaSetCalibrationValue('anEtk', 'Map',
yMap, 'y');
IncaSetCalibrationValue('anEtk', 'Curve',
aCurveRange, 2);
IncaSetCalibrationValue('anEtk', 'Map',
aMapRange, [2, 3]);
IncaSetCalibrationValue('anEtk', 'Map',
xMapRange, 2, 'x');

```

4.5.5 Assigner un jeu de données au dispositif (INCA-MIP Evoluée)

Nom	<code>IncaSetDatasetInDevice</code>	
Description	Cette fonction permet d'assigner un jeu de donnée à un dispositif dans une expérimentation ouverte	
Syntaxe	<code>IncaSetDatasetInDevice (device, dataset)</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>device</code>	Nom du dispositif
	<code>dataset</code>	Chemin d'accès du jeu de données dans la base de données INCA
Exemples	<pre>IncaSetDatasetInDevice ('ETK:1', 'Ds4711\Ds4711_3')</pre>	

4.5.6 Lister les jeux de données d'un dispositif (INCA-MIP Evoluée)

Nom	<code>IncaGetDatasetsForDevice</code>	
Description	Cette fonction permet d'obtenir la liste des noms de tous les jeux de données pour un dispositif	
Syntaxe	<pre>nom = IncaGetDatasetsForDevice (device) [nom, properties] = IncaGetDa- tasetForDevice (device)</pre>	
Arguments de sortie	<code>nom</code>	Liste, avec le chemin d'accès, de tous les jeux de données trouvés
	<code>properties</code>	Liste des propriétés des jeux de données Les valeurs possibles sont: – " (champ vide): Jeu de données en accès lecture/écriture – r : Jeu de données en lecture seule – m : Jeu de données maître en accès lecture/écriture – mr : Jeu de donnée maître en lecture seule
Arguments d'entrée	<code>device</code>	Nom du dispositif
Exemples	<pre>l = IncaGetDatasetsForDevice ('ETK:1')</pre>	

4.5.7 Etablir le mode de calibration (INCA-MIP Evoluée)

Nom	<code>IncaSetCalibrationMode</code>	
Description	<p>Cette fonction permet de fixer le mode de calibration global valide pour toutes les calibrations définis avec la fonction <code>IncaSetCalibrationValue</code>. Le mode reste valide même après avoir fermé puis réouvert l'expérimentation. En démarrant l'interface MATLAB®, le mode par défaut pour les limites hautes et basses est : <code>rejectWeakBoundViolation</code>.</p>	
Syntaxe	<code>IncaSetCalibrationMode(lowerLimitMode, upperLimitMode)</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>lowerLimitMode</code>	Le nouveau mode de calibration pour les limites basses.
	<code>upperLimitMode</code>	<p>Le nouveau mode de calibration pour les limites hautes:</p> <ul style="list-style-type: none"> – <code>rejectWeakBoundViolation</code>: Rejette complètement la calibration si une limite logicielle est dépassé au moins une fois (par défaut) – <code>limitToWeakBound</code>: Si les limites logicielles hautes haute sont dépassées, utiliser à la place la valeur de la limite logicielle haute ou basse – <code>rejectHardBoundViolation</code>: Ignorer les limites logicielles. Rejetter la calibration si les limites matérielles sont dépassées au moins une fois – <code>limitToHardBound</code>: Ignorer les limites logicielles. Si les limites matérielles hautes haute sont dépassées, utiliser à la place la valeur de la limite matérielle haute ou basse.
Exemples	<pre>IncaSetCalibrationMode ('rejectHardBoundViolation', 'limitToHardBound')</pre>	

4.5.8 Groupement des dispositifs (INCA-MIP Evoluée)

Nom	<code>IncaGroupDevices</code>	
Description	Activer ou désactiver le groupement de dispositif	
Syntaxe	<code>IncaGroupDevices (onOff)</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>onOff</code>	<ul style="list-style-type: none"> – 0 : Désactiver le groupement de dispositif – 1 : Activer le groupement de dispositif
Exemples	<code>IncaGroupDevices (1)</code>	

4.5.9 Ecrire un fichier DCM (INCA-MIP Evoluée)

Nom	<code>IncaWriteToFile</code>	
Description	Cette fonction permet écrire un fichier DCM à partir des calibrations de l'expérimentation ouverte	
Syntaxe	<code>IncaWriteToFile (format, file, device, calibs{, options})</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>format</code>	Identificateur du format du fichier: – 'DCM': format DCM
	<code>file</code>	Chemin d'accès complet du fichier à écrire
	<code>device</code>	Dispositif des calibrations qui seront écrites
	<code>calibs</code>	Liste des éléments de calibration à écrire (tableau)
	<code>options</code>	Options utilisées pour écrire au format indiqué
Exemples	<pre>calibs = {'A0_KW', 'BRABEVI_KL', 'KFZW_GKF'}; IncaWriteToFile('DCM', 'C:\DCMOut1.dcm', 'device1', calibs); IncaWriteToFile('DCM', 'C:\DCMOut2.dcm', 'ETK:1', 'A0_KW');</pre>	

4.6 Gestion des pages mémoires

Toutes les fonctions prédemment décrites sont effectives pour la page actuellement active d'un dispositif. En principe, l'accès à la calibration est possible à partir de la page de travail uniquement. Cependant, même dans ce cas, il est possible que le droit d'écriture dans la page de travail de l'ETK soit bloqué parce que les sommes de contrôle de la page de travail entre INCA et l'ETK ne correspondent pas.

Les fonctions API suivantes peuvent être utilisées pour la gestion des pages mémoires.

4.6.1 Activer une page mémoire

Nom	IncaSwitchPage	
Description	Cette fonction permet d'activer la page mémoire spécifiée	
Syntaxe	IncaSwitchPage(deviceName, pageName)	
Arguments de sortie	none	
Arguments d'entrée	deviceName	Nom du dispositif
	pageName	Nom de la page: <ul style="list-style-type: none"> — wp: Page de travail — rp: Page de référence
Exemples	<pre>IncaSwitchPage('MyDevice', 'wp'); IncaSwitchPage('Dev', 'rp');</pre>	

4.6.2 Get Current Page (INCA-MIP Evoluée)

Nom	IncaGetCurrentPage	
Description	Cette fonction permet de connaître la page mémoire actuellement active	
Syntaxe	pageName = IncaGetCurrentPage(deviceName)	
Arguments de sortie	pageName	Nom de la page mémoire active: <ul style="list-style-type: none"> — wp: Page de travail — rp: Page de référence
Arguments d'entrée	deviceName	Nom du dispositif
Exemples	<pre>p = IncaGetCurrentPage('MyDevice');</pre>	

4.6.3 Vérifier la protection en écriture

Nom	IncaIsPageWriteProtected	
Description	Cette fonction permet de vérifier si la page mémoire spécifiée est protégée en écriture	
Syntaxe	<code>isRW = IncaIsPageWriteProtected(deviceName, pageName)</code>	
Arguments de sortie	<code>isRW</code>	<ul style="list-style-type: none"> – 0 : la page n'est pas protégée en écriture – not 0 : la page est protégée en écriture
Arguments d'entrée	<code>deviceName</code>	Nom du dispositif
	<code>pageName</code>	Nom de la page: <ul style="list-style-type: none"> – wp: Page de travail – rp: Page de référence
Exemples	<pre>isETK1RW = IncaIsPageWriteProtected('ETK:1', 'wp'); isETK2RW = IncaIsPageWriteProtected('ETK:2', 'rp');</pre>	

4.6.4 Télécharger vers l'ECU la page mémoire

Nom	IncaDownloadPage	
Description	Cette fonction permet de télécharger la page mémoire spécifiée dans le calculateur	
Syntaxe	<code>IncaDownloadPage(deviceName, pageName)</code>	
Arguments de sortie	none	
Arguments d'entrée	<code>deviceName</code>	Nom du dispositif
	<code>pageName</code>	Nom de la page à télécharger <ul style="list-style-type: none"> – wp: Page de travail – rp: Page de référence
Exemples	<pre>IncaDownloadPage('ETK:1', 'wp'); IncaDownloadPage('ETK:1', 'rp');</pre>	

4.6.5 Copier la page mémoire

Nom	IncaCopyPageFromTo	
Description	Cette fonction permet de copier la page mémoire spécifiée. Actuellement il est uniquement possible de copier de la page de référence vers la page de travail; les autres combinaisons de copie entre sources et cibles ne sont pas supportées.	
Syntaxe	IncaCopyPageFromTo(deviceName, sourcePageName, destinationPageName)	
Arguments de sortie	none	
Arguments d'entrée	deviceName	Nom du dispositif
	sourcePageName	Nom de la page à copier (source): <ul style="list-style-type: none"> – wp: Page de travail – rp: Page de référence
	destinationPageName	Nom de la page à écraser (cible): <ul style="list-style-type: none"> – wp: Page de travail – rp: Page de référence
Exemples	IncaCopyPageFromTo('ETK:1', 'rp', 'wp');	

4.6.6 Télécharger les différences

Nom	IncaDownloadDifferences	
Description	Cette fonction permet de charger les différences entre la page de travail et la page de référence dans le calculateur. Comme sur INCA, ceci n'est valable que si la page de travail et la page de référence du calculateur cible correspondent à la page de référence présente sous INCA.	
Syntaxe	IncaDownloadDifferences(deviceName)	
Arguments de sortie	none	
Arguments d'entrée	deviceName	Nom du dispositif
Exemples	IncaDownloadDifferences('ETK:1');	

4.6.7 Télécharger depuis l'ECU les pages mémoires (INCA-MIP Evoluée)

Nom	IncaUploadPages	
Description	Cette fonction permet de récupérer sous INCA la page de référence et de travail du calculateur. Ces nouveaux jeux de données sont automatiquement assignés au dispositif.	
Syntaxe	IncaUploadPages (device{, referencePage, workingPage})	
Arguments de sortie	none	
Arguments d'entrée	device	Nom du dispositif
	referencePage	Nom du jeu de données pour la page de référence récupérée. Si non précisé, INCA utilise un nom par défaut
	workingPage	Nom du jeu de données pour la page de travail récupérée. Si non précisé, INCA utilise un nom par défaut
Exemples	<pre>IncaUploadPages ('ETK:1'); IncaUploadPages ('ETK:1', 'ref_1', 'work_1');</pre>	

4.7 Exemples

Exemple 1

```
% Vérifier si la page de travail est protégée en
% écriture et envoyer la page si tel est le cas.
if(IncaIsPageWriteProtected ('anEtk', 'wp'))
    IncaDownloadPage ('anEtk', 'wp');
end

% Passer à la page de travail
IncaSwitchPage ('anEtk', 'wp');
```

Exemple 2

Dans l'exemple ci-dessous, les fonctions sont utilisées pour lire les valeurs des variables de mesure du dispositif `MyDevice` avec une fréquence

d'échantillonnage de 10ms. Pour utiliser cet exemple vous devez d'abord ouvrir une expérimentation sous INCA qui inclue un dispositif nommé `MyDevice` avec 4 voies nommée : `Chan1`, `Chan2`, `Chan3`, `Chan4`.

```

% Sélectionner les signaux suivant
IncaAddMeasureElement( 'MyDevice', '10ms', 'Chan1' );
IncaAddMeasureElement( 'MyDevice', '10ms', 'Chan2' );
IncaAddMeasureElement( 'MyDevice', '10ms', 'Chan3' );
IncaAddMeasureElement( 'MyDevice', '10ms', 'Chan4' );
% Maintenant lancer la mesure
data = [];
time = [];
IncaShowMessages(0);
IncaSetMeasureReadMode(0)
IncaStartMeasurement;
deltaT = 0;
% Mesurer pendant 20 secondes
while( deltaT < 20 )
    % Faire une pause pendant 0.1 secondes pour avoir
    % plus d'un échantillon -- % économise le temps
processeur
    pause(0.1)
    % Récupérer 500 échantillons du groupe 10ms
    [ t, d ]=IncaGetRecords( 'MyDevice', '10ms', 500
);
    % Ajouter t et d au temps et aux données
    data = [data; d];
    time = [time; t];
    if( length(time) )
        % Calculer le temps de la mesure
        deltaT = time( length(time)) - time(1);
    end
end
IncaStopMeasurement;
IncaShowMessages(1);
% Tracer le résultat
plot(time, data);

```

Cet exemple n'utilise qu'une seule fréquence d'échantillonnage. Il vous est cependant possible d'utiliser plusieurs fréquences et d'effectuer une requête pour les données de chaque fréquence. MATLAB®

5 Création et distribution de fichiers exécutables autonomes à l'aide du compilateur MATLAB[®]

Avec l'AddOn INCA-MIP, vous pouvez créer et compiler des fichiers m contenant des appels aux fonctions API de INCA-MIP, y compris des fonctions MATLAB[®]. Le résultat est un fichier autonome qui peut être utilisé dans n'importe quel environnement sans forcément avoir le logiciel MATLAB[®].

Pour la création de fichiers exécutables une installation de MATLAB[®] est nécessaire. L'exécutable qui en résulte, ainsi que les copies de certaines DLL de MATLAB[®] et ETAS, peuvent être utilisés sans installation de MATLAB[®] sur le système cible.

En outre, les bibliothèques d'exécution MATLAB[®] peuvent être nécessaires pour exécuter les exécutables compilés sur le système cible.

Reportez-vous à la documentation de l'utilisateur de MATLAB[®] sous Distribuer des applications autonomes pour savoir comment installer et utiliser les bibliothèques d'exécution de MATLAB[®].

5.1 Compilation des fichiers m

Compiler des fichiers m avec le compilateur MATLAB[®] :

1. Copiez tous les fichiers `Inca*.dll` et `Inca.*.mexw64` dans le répertoire de travail actuel.

En fonction du dossier sélectionné lors de l'installation, les fichiers DLL et MEXW64 se trouvent dans ce dossier.

EXEMPLE

```
<drive:\>Program Files\MATLAB\Ryyyy\bin\win64
<drive:\>ETASData\INCAx.y\INCA-MIPx64
```

Assurez-vous que le compilateur MATLAB[®] et les chemins d'accès à INCA-MIP sont ajoutés à la variable d'environnement Windows.

Pour définir les variables d'environnement, reportez-vous à la section Paramètres système avancés de la documentation utilisateur Windows.

Exécuter la commande suivante :

```
mcc -m <m-file-script> -a incaRci2Matlab.dll
```

EXEMPLE

Avec la commande suivante un fichier exécutable est créé à partir du fichier `testCase2.m` :

```
mcc -m testCase2 -a incaRci2x64Matlab.dll
```

Comme résultat vous obtenez le fichier `testCase2.exe`.

Le compilateur MATLAB[®] crée un conteneur contenant toutes les DLL de fonctions MEX et les DLL dépendantes nécessaires à l'exécution du script compilé MATLAB[®]. Tous les fichiers Inca *.dll utilisés par le script ainsi que le fichier `incaR-ci2x64Matlab.dll` doivent faire partie de ce conteneur. Les DLL de la fonction INCA MEX ont l'extension *.mexw64.

Lorsque le script compilé est exécuté, les DLL ne doivent pas être présentes sur le système.

Reportez-vous à votre documentation utilisateur MATLAB[®], mot-clé *MATLAB[®] Compilateur* ou *mcc* pour les versions requises et les paramètres correspondants du compilateur MATLAB[®].



Note

INCA ne peut être contrôlé que par une seule session MATLAB[®] à la fois. La tentative d'exécuter une deuxième fois INCA depuis une deuxième instance MATLAB[®] ou par l'intermédiaire d'un fichier exécutable est interrompue avec un message d'erreur.



Note

Avec MATLAB[®] R2016a ou une version ultérieure, les DLL de la fonction MEX de INCA ont l'extension *.mexw64.

5.2 Distribution des fichiers exécutables autonome

L'exécution des fichiers exécutables autonomes qui ont été compilés à l'aide du compilateur MATLAB[®] ne nécessite que l'exécutable lui-même. Une installation de MATLAB[®] ou des copies des bibliothèques ne sont pas nécessaires. MATLAB[®] Distribuer des fichiers exécutables autonomes compilés avec le compilateur MATLAB[®]

- Copiez les fichiers exécutables autonomes sur le système cible.

Ensuite, vous pouvez les exécuter ; aucune autre étape n'est nécessaire.

6 Informations des Contacts

Support technique

Pour les détails sur vos services de distribution en plus que votre équipe de support et vos hotlines, regardez les pages d'internet ETAS :

www.etas.com/hotlines

ETAS propose des formations pour ses produits :

www.etas.com/academy



ETAS siège principal

ETAS GmbH

Borsigstraße 24	Phone :	+49 711 3423-0
70469 Stuttgart	Fax :	+49 711 3423-2106
Allemagne	Internet :	www.etas.com

Index

B	
Buffer tournant	10
C	
Calibration	9
D	
Dispositif	10
Données	8
Données de mesure	10
E	
Enregistrement des données	10
ETAS	
Informations des Contacts	76
F	
fichiers d'échantillonnage	19
fichiers exécutables autonomes	74
fichiers m	9
fichiers M	19
fichiers MEX	12
Fréquence d'échantillonnage	11
G	
gestionnaire de pages de mémoire	9
I	
IncaAddCalibrationElement	59
IncaAddMeasureElement	36
IncaBrowseCalibrationElements	58
IncaBrowseItemsInFolder	29
IncaBrowseMeasureElements	35
IncaClose	27
IncaCopyPageFromTo	71
IncaDatabaseImport	28
IncaDownloadDifferences	71
IncaDownloadPage	70
IncaExecuteManualTrigger	56
IncaGetCalibrationValue	60
IncaGetCurrentPage	69
IncaGetDatasetsForDevice	66
IncaGetDeviceProperties	33
IncaGetDevices	33
IncaGetHardwareStatus	52
IncaGetInstalledAddOnInfo	24
IncaGetInstalledProductInfo	23
IncaGetMeasureRatesForDevice	36
IncaGetProperties	25
IncaGetRecordingMode	44
IncaGetRecordingProperties	39
IncaGetRecordingState	56
IncaGetRecords	48
IncaGetRecordStruct	57
IncaGetVersion	25
IncaGroupDevices	68
IncalsLicenseValid	23
IncalsPageWriteProtected	70
IncaOpen	26
IncaOpenDatabase	27
IncaOpenExperiment	31
IncaResetExperiment	32
IncaResetRecords	51
IncaSetCalibrationMode	67
IncaSetCalibrationValue	62
IncaSetDatasetInDevice	66
IncaSetMeasureReadMode	47
IncaSetProjectAndDatasetInDevice	31
IncaSetRecordingMode	45
IncaSetRecordingProperties	41
IncaSetTrigger	52
IncaShowMessages	23
IncaStartMeasurement	38
IncaStartRecording	46
IncaStopMeasurement	39
IncaStopRecording	46
IncaSwitchPage	69
IncaUploadPages	72
IncaWriteToFile	68
Informations des Contacts	76
M	
mcc	74
Measuring	9
R	
Raster de mesure	10
S	
scripts MATLAB	9
Signal	11
V	
Variable de calibration	10