

ETAS INCA-MIP V7.5



ユーザーガイド

著作権について

本書のデータをETAS GmbHからの通知なしに変更しないでください。ETAS GmbHは、本書に関してこれ以外の一切の責任を負いかねます。本書に記載されているソフトウェアは、お客様が一般ライセンス契約あるいは単一ライセンスをお持ちの場合に限り使用できます。ご利用および複製はその契約で明記されている場合に限り、認められます。

本書のいかなる部分も、ETAS GmbHからの書面による許可を得ずに、複製、転載、伝送、検索システムに格納、あるいは他言語に翻訳することは禁じられています。

© **Copyright 2024** ETAS GmbH, Stuttgart

本書で使用する製品名および名称は、各社の（登録）商標あるいはブランドです。

MATLABとSimulinkは、The MathWorks, Inc. の登録商標です。その他の商標については [mathworks.com/trademarks](https://www.mathworks.com/trademarks) をご参照ください。

INCA-MIP V7.5 | ユーザーガイド R02 JP | 06.2024

本書の内容

1	はじめに	6
1.1	製品の正しい使用法	6
1.2	対象ユーザー	6
1.3	安全に関する注意事項の書式	6
1.4	本製品使用時の安全に関する注意事項	7
1.5	データ保護	7
1.6	データと情報のセキュリティ	7
2	INCA-MIPの概要	8
2.1	INCA用語集	8
3	INCA-MIPアドオンのインストール	11
3.1	システム要件	11
3.2	インストール	11
3.3	MATLAB® Toolboxディレクトリ用キャッシュの更新	13
3.4	MATLAB® Toolboxディレクトリ用キャッシュの無効化	13
3.5	ソフトウェアのライセンス管理	14
4	API関数	15
4.1	INCA-MIP APIのサンプルファイル	18
4.2	一般的な機能	18
4.2.1	INCA-MIPインターフェースのメッセージID取得	19
4.2.2	スクリプト実行中のメッセージ表示	21
4.2.3	有効なINCA-MIPライセンスの確認（拡張コマンド）	21
4.2.4	インストール済みINCAバージョンに関する情報の取得	22
4.2.5	インストール済みアドオンに関する情報の取得	22
4.2.6	INCAバージョンの取得	23
4.2.7	INCAプロパティの取得（拡張コマンド）	23
4.3	初期化	24
4.3.1	INCAを開く	24
4.3.2	INCAを閉じる（拡張コマンド）	25
4.3.3	データベースを開く	25
4.3.4	データベースのインポート（拡張コマンド）	25
4.3.5	データベースアイテムの読み込み（拡張コマンド）	26
4.3.6	デバイスにプロジェクトとデータセットを割り当てる（拡張コマンド）	27
4.3.7	実験を開く	28
4.3.8	実験のリセット	28

4.3.9	デバイスの取得 (拡張コマンド)	29
4.3.10	デバイスプロパティの取得 (拡張コマンド)	29
4.4	測定と記録	30
4.4.1	測定変数の取得 (拡張コマンド)	31
4.4.2	測定ラスタの取得 (拡張コマンド)	31
4.4.3	測定変数を実験に追加する	32
4.4.4	測定の開始	33
4.4.5	測定終了	34
4.4.6	記録のプロパティの取得 (拡張コマンド)	34
4.4.7	記録のプロパティの設定 (拡張コマンド)	36
4.4.8	記録モードの取得 (拡張コマンド)	37
4.4.9	記録モードの設定 (拡張コマンド)	38
4.4.10	記録の開始	39
4.4.11	記録の終了	39
4.4.12	データの読み込みモードの設定 (オンライン・オフライン)	40
4.4.13	測定データの取得	41
4.4.14	リングバッファのリセット	43
4.4.15	ハードウェアステータスの取得 (拡張コマンド)	43
4.4.16	トリガの設定 (拡張コマンド)	44
4.4.17	手動トリガの発行 (拡張コマンド)	47
4.4.18	記録ステータスの取得 (拡張コマンド)	47
4.4.19	測定変数リストの取得 (拡張コマンド)	47
4.5	適合	48
4.5.1	適合変数の取得 (拡張コマンド)	48
4.5.2	適合変数の追加	49
4.5.3	適合値の取得	49
4.5.4	適合値の変更 (適合の実行)	52
4.5.5	データセットをデバイスに割り当てる (拡張コマンド)	55
4.5.6	デバイスのデータセットリストを取得 (拡張コマンド)	55
4.5.7	適合モードの設定 (拡張コマンド)	56
4.5.8	グループデバイス (拡張コマンド)	56
4.5.9	DCMファイルの書き込み (拡張コマンド)	57
4.6	メモリページマネージャ	57
4.6.1	メモリページの切り替え	57
4.6.2	カレントページの取得 (拡張コマンド)	58
4.6.3	書き込み保護チェック	58
4.6.4	メモリページのダウンロード	59
4.6.5	メモリページのコピー	59
4.6.6	不一致部分のダウンロード	59

4.6.7	メモリページのアップロード（拡張コマンド）	60
4.7	応用例	60
5	コンパイラを用いたスタンドアロン実行ファイルの作成と配布	62
5.1	Mファイルのコンパイル	62
5.2	スタンドアロン実行ファイルの配布	63
6	お問い合わせ先	64
索引		65

1 はじめに

1.1 製品の正しい使用法

INCAとINCAアドオンは、自動車への応用を前提に開発されたものであり、それらのユーザードキュメントに記述された範囲でのみ使用することができます。

INCA-MIPアドオン（INCA MATLAB[®]統合パッケージ）は、INCAを用いてMATLAB[®]の機能を制御するためのAPI（アプリケーションプログラミングインターフェース）です。

INCAとINCAアドオンは、工業用実験室や試験用車両での使用を想定しています。

ETAS GmbHは、誤った使い方や安全情報を守らないことによって生じた損害については責任を負いかねます。

1.2 対象ユーザー

本ソフトウェア製品および本ユーザーガイドは、自動車用ECUの開発・適合に携わる有資格者や、ソフトウェアをインストール・保守・アンインストールするシステム管理者または管理者権限のあるユーザーを対象としています。計測とECUに関する技術的な専門知識が必要とされます。

INCA-MIP APIを使用するには、INCAとMATLAB[®]についての基本知識が必要です。またMATLAB[®]スクリプトの使用方法についてもよく理解している必要があります。

1.3 安全に関する注意事項の書式

以下の「安全に関する注意事項」は、人身事故や物的損害につながる危険性を警告するものです。



危険

記載事項を守らないと死亡または重傷のリスクが高い危険性について説明しています。



警告

記載事項を守らないと死亡または重傷のリスクを招く可能性のある危険性について説明しています。



注意

記載事項を守らないと軽～中程度の負傷のリスクを招く可能性のある危険性について説明しています。

ご注意ください！

記載事項を守らないと物的損害を招く可能性のある状況について説明しています。

1.4 本製品使用時の安全に関する注意事項

INCAとINCAアドオンを用いた作業を行う際には、以下の安全情報を遵守してください。

**警告****予期しない車両の挙動を招く危険があります。**

適合操作は、ECU、およびECUに接続されたシステムの挙動に影響を与えます。

その結果、エンジンが停止したり、予期せぬ車両の挙動（ブレーキング、加速、操舵など）が発生する可能性があります。

適合操作は、製品の使用に関する講習を受け、接続されたシステムの起こり得る反応を評価できる方のみが実施してください。

**警告****予期しない車両の挙動を招く危険があります。**

CAN、LIN、FlexRay、イーサネットなどのバスシステムでメッセージを送信すると、接続されたシステムの動作に影響を与えます。

その結果、エンジンが停止したり、予期せぬ車両の挙動（ブレーキング、加速、操舵など）が発生する可能性があります。

バスシステム経由のメッセージ送信は、各バスシステムの使用に関する十分な知識があり、接続されたシステムの起こり得る反応を評価できる方のみが実施してください。

「ETAS Safety Advice - 安全上のご注意」の指示、およびオンラインヘルプとユーザーガイドに記載されている安全情報を遵守してください。この情報は、INCAのヘルプメニューから安全上のご注意を選択して開くことができます。

1.5 データ保護

製品に個人データを処理する機能が含まれている場合、データ保護およびデータプライバシーに関する法律上の要件は、お客様が遵守するものとします。製品の当該機能に続いて行われる処理は、通常、データ管理者であるお客様が設計するものとします。したがって、保護措置が十分かどうかのチェックもお客様に行っていただく必要があります。

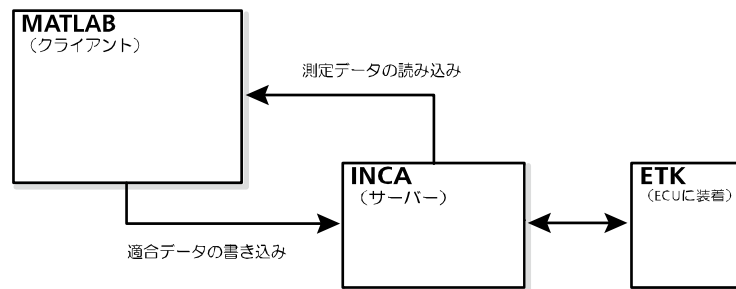
1.6 データと情報のセキュリティ

本製品におけるデータの安全な取り扱いについては、INCAヘルプ内の「データと情報のセキュリティ」セクションを参照してください。

2 INCA-MIPの概要

INCA-MIPアドオン (INCA MATLAB[®]統合パッケージ) は、INCAを用いてMATLAB[®]の機能を制御するためのAPI (アプリケーションプログラミングインターフェース) です。クライアントであるMATLAB[®]がこれらのAPIを用いてサーバーであるINCAのリソースにアクセスします。

下図は、INCA-MIP APIの典型的な使用例を示すものです。INCAとECU間のデータアクセスにはETKが使用されています。



INCA-MIP APIを用いることにより、以下のようなINCAの機能をMATLAB[®]から実行することができます。

メモリページ管理

メモリページの切り替え、ECUへのメモリページのダウンロードなどが行えます。

適合

INCAの実験に含まれるすべての適合変数を適合することができます。値は、変数ごとに読み取って変更でき、カーブやマップが参照する軸ポイントディストリビューションの書き換えも可能です。

測定

INCAの実験に含まれるすべての測定変数を読み取ることができます。測定の開始や停止をMATLAB[®]から行うこともできます。INCAのステータス情報などもMATLAB[®]からアクセスできます。INCA-MIPインターフェースのパフォーマンスは最適化され、高いスループットを実現しています。

INCAは、測定変数と適合変数の値を倍精度の浮動小数点型で送信するので、MATLAB[®]側で値を変換する必要がありません。

MATLAB[®]スクリプト (「Mファイル」と呼ばれます) からINCA-MIP API関数 (以下「コマンド」とも呼ばれます) を呼び出すことにより、INCAの実験の全制御フローを定義することができます。

本書は、INCA-MIP APIの概念、インストール方法、使用方法、さらに各API関数の機能と使用方法について説明するものです。本書にはMATLAB[®]とINCAの基本概念や使用方法は記載されていません。

2.1 INCA用語集

APIの説明では、INCAを使い慣れたユーザーであればすでにご存じの技術用語が使用されています。以下にこれらの用語を簡単に説明します。

適合変数

適合変数は、値の読み込みと書き込みが可能な変数です。適合変数のタイプには、スカラー値、配列、行列、カーブ、マップがあります。カーブとマップが参照する軸ポイントディストリビューションも、読み込みと書き込みが可能です。

データレコード

1つのレコードは1回の測定で生成され、タイムスタンプと、1つのシグナルグループに含まれるすべての測定変数の値で構成されます。1つのシグナルグループの測定データは、測定プロセス全体を通じて生成される複数のレコードで構成されます。

デバイス

測定変数の値を一定の周期で取り込むことのできる測定用ハードウェアを指します。一部の測定デバイスでは適合変数の適合を行うこともできます。たとえばSMBデバイスは測定のみ可能ですが、ETKは測定と適合が可能です。

測定データ

1回の測定作業において各測定ラスタで記録されたすべての測定値を指します。複数のデータレコードから構成されます。

測定ラスタ

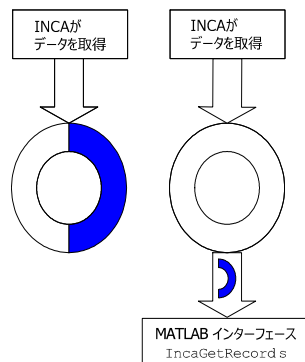
1つのシグナルグループとして一連のシグナルを取り込むためのサンプリングレート（測定周期）を表すものです。

複数のラスタを「マルチラスタ」として組み合わせて使用することができます。マルチラスタは、使用したいラスタの名前を '+' という文字で連結するだけで簡単に定義することができます（例：'10ms+100ms'）。このような「マルチラスタ」を指定すると、新しい仮想ラスタが生成されます。1つのシグナルの測定は、1つのラスタ、またはマルチラスタのいずれかでのみ行えます。

リングバッファ

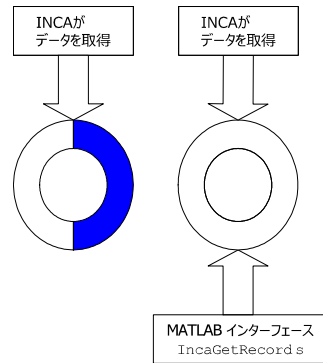
INCAからMATLAB[®]へ確実に測定データを転送するために、各測定ラスタ（シグナルグループ）ごとに専用のリングバッファが使用されます。INCAでオンライン測定または記録が行われている間、取得された測定データは、自動的にリングバッファ内に保存されます。

リングバッファに保存されたタイムスタンプとデータをMATLAB[®]に転送するには、`IncaGetRecords`コマンドを使用します。



リングバッファに保存される測定データはシグナルあたり1Mバイトに制限されており、全体で約8秒間の測定データを保持することができますが、この時間はサンプリングレートに応じて変わります。バッファが満杯になると、データは古い順に上書きされます。データが上書きされて失われてしまうのを防ぐには、周期的にIncaGetRecordsを実行してください。適切な周期は、通常1秒です。

IncaResetRecordsコマンドを実行すると、すべての測定ラスタのリングバッファ内に保存されたタイムスタンプとデータを消去することができます。これにより、リングバッファに保存されていたデータはすべて失われます。



シグナル

シグナルは、INCAで測定される1つの測定変数を指します。各シグナルには、データ型（論理値、整数、浮動小数点）、データ長（1、2、4、8）、および変換式が定義されています。変換式は、物理値から実装値への変換方法を定義したものです。

シグナルグループ

シグナルグループは複数のシグナルで構成されます。1つのシグナルグループ内のすべてのシグナルは1つの測定ラスタで測定されます。各シグナルグループには一意の名前があります。

3 INCA-MIPアドオンのインストール

INCA-MIPは、INCAの機能を拡張するためのアドオン製品です。

MATLAB[®]は、動的にリンクされる関数呼び出し（「MEXファイル」と呼ばれます）を使用して他のアプリケーションと通信します。INCA-MIP APIは一連のMEXファイルからなります。これらのファイルは、インストール時にMATLAB[®]プログラムディレクトリのサブディレクトリにコピーされます。

INCA-MIPに含まれるAPI関数は、2つのセットに分かれています。INCA-MIP Base APIセット（基本セット）は本製品のインストール後すぐに使用できます。INCA-MIP Extended APIセット（拡張セット）に含まれる拡張コマンド（= 拡張API関数）を使用するには、ソフトウェアライセンスキーが必要です。全API関数の一覧、および各関数が属するセット（拡張または基本）は、「API関数」（ページ15）にまとめられています。

3.1 システム要件

INCA-MIPアドオンを使用するには、PCにINCAがインストールされている必要があります。

INCAのシステム要件については、『インストールガイド』を参照してください。

MATLAB[®]にアクセスするためのINCAスクリプトを開発する場合は、MATLAB[®]のフルライセンスが必要です。

INCA V7.5用のINCA-MIPを使用するには以下のプログラムが必要です。

- INCA V7.5 SPx

注記

本バージョンのINCA-MIPをインストールするには、PCにINCA V7.5 がインストールされている必要があります。

INCAのリリース番号がINCA-MIPアドオンパッケージのリリース番号と互換性があることを確認してください。

インストール後は、すべてのINCA V7.x バージョンとともに本バージョンのINCA-MIPを使用することができます（「INCAを開く」（ページ24）を参照）。

- MATLAB[®] 64ビットバージョンの2016a以降（MATLAB[®]への統合インストールを行う場合）

サポートされているMATLAB[®]バージョンについての詳細な情報は、INCAサポート窓口までお問い合わせください。

3.2 インストール

INCA-MIPアドオンのインストール時には、インストールタイプを選択する必要があります。以下のいずれかのインストールタイプから選択できます。

- **MATLAB[®] integrated installation（統合インストール）**

1つの特定のバージョンのMATLAB[®]を使用してMATLAB[®]スクリプトを開発する場合は、このオプションを選択してください。

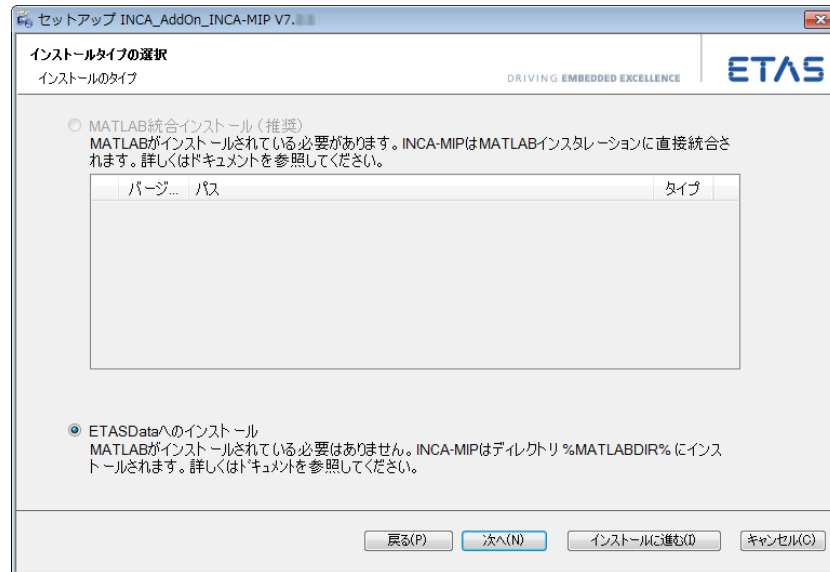
– Installation into ETASData

コンパイル済みのMATLAB[®]スクリプトの実行のみを行う場合、またはPC上にインストールされた複数バージョンのINCA-MIPからMATLAB[®]を使用する場合は、このオプションを選択してください。詳細な説明については、下記を参照してください。

INCA-MIPをインストールするには：

INCA-MIPをインストールする前に、INCAがすでにPCにインストールされており、INCAインストールのリリース番号がアドオンパッケージのリリース番号と互換性があることを確認してください。MATLAB[®]にアクセスするためのINCAスクリプトを開発する場合は、MATLAB[®]がすでにPCにインストールされており、MATLAB[®]インストールのリリース番号がINCA-MIPアドオンパッケージのリリース番号と互換性があることを確認してください。

1. PCで実行されているすべてのプログラムを終了します。
2. 社内規定に応じて、インストーラファイルはDVDまたはネットワークドライブで提供されます。
DVDの場合は、自動的にインストールが開始されます。自動的に開始されない場合は、Autostart.exeというファイルを手動で実行します。
ネットワークドライブからの場合は、setup.exeというファイルを実行します。
3. インストールルーチンの指示に従ってINCA-MIPをPCにインストールします。
4. インストールルーチンでは、希望するインストールタイプを指定するよう求められます。



5. PCにインストールされている1つのバージョンのMATLAB[®]を用いてINCA-MIP用MATLAB[®]スクリプトを開発する場合は、**MATLAB[®] integrated installation**を選択します。

または

以下のいずれかの条件に該当する場合は、**Installation into ETASData**を選択します。

- 複数のバージョンのINCA-MIPでMATLAB[®]を使用する
PCにインストールされた複数のバージョンのMATLAB[®]からINCA-MIPのコマンドを使用するには、各バージョンのMATLAB[®]ツールボックスディレクトリに¥INCA-MIPサブディレクトリをコピーしておく必要があります。MATLAB[®]のパスにディレクトリを追加する方法は、MATLAB[®]のユーザードキュメントを参照してください。
- MATLAB[®]で作成されたMATLAB[®]スタンドアロン実行ファイルの実行のみを行い、スクリプトの開発は行わない

注記

MATLAB[®]を操作するためのINCAコマンドを含むスタンドアロン実行ファイルを実行するには、INCA-MIPをETASDataディレクトリにインストールしてください。この場合、MATLAB[®]のライセンスは必要ありません。ただしこの実行ファイルを作成するには、MATLAB[®]がインストールされた開発環境が必要です。詳しくは「[コンパイラを用いたスタンドアロン実行ファイルの作成と配布](#)」(ページ62)を参照してください。

6. インストールルーチンの指示に従い処理を続行します。

INCA-MIPのライセンスをインストールするには：

拡張セットのAPI関数を使用する場合はソフトウェアライセンスファイルが必要です。

ライセンスについては「[ソフトウェアのライセンス管理](#)」(次ページ)を参照してください。

3.3 MATLAB[®] Toolboxディレクトリ用キャッシュの更新

INCA-MIPインストール時にMATLAB[®]ツールボックスディレクトリ用のキャッシュが有効に設定されていた場合は、MATLAB[®] APIをインストールした後にこのキャッシュを更新する必要があります。MATLAB[®] V6以降のバージョンをデフォルト設定でインストールした場合はキャッシュは有効になっていて、それより前のバージョンでは無効になっています。INCA-MIP APIで使用されるファイルがMATLAB[®]に登録されるように、キャッシュを更新する必要があります。

MATLAB[®]ツールボックスディレクトリ用キャッシュを更新する方法は、MATLAB[®]のユーザードキュメントを参照してください。

3.4 MATLAB[®] Toolboxディレクトリ用キャッシュの無効化

INCA-MIP APIを使用する際は、MATLAB[®]ツールボックスディレクトリ用のキャッシュは無効にしておくことをお奨めします。有効になっていると、INCA-MIP APIや、新しく個別に追加されたスクリプトファイルが認識されなくなる場合があります。

キャッシュを無効にする代わりに、キャッシュが有効な状態で上記のように強制的に更新することは可能です。しかし、予期しない不具合の発生を防ぐため、INCA-MIP APIを使用する際にはキャッシュを無効にしておくことをお奨めします。

MATLAB[®]ツールボックスディレクトリ用キャッシュの有効／無効を切り替える方法は、MATLAB[®]のユーザードキュメントを参照してください。

3.5 ソフトウェアのライセンス管理

ETASのソフトウェアを使用するには、有効なライセンスが必要です。ライセンスは、下記のいずれかの方法で入手できます。

- － ツール管理者
- － ETASホームページのセルフポータルサイト：
www.etas.com/support/licensing
- － ETAS ライセンスマネージャ（ETAS License Manager）

ライセンスをアクティベートする際は、製品のご購入時にETASから入手したアクティベーションIDを入力する必要があります。

ライセンスの管理について詳しくは、[ETAS License Management FAQ](#)またはライセンスマネージャのヘルプを参照してください。

ETASライセンスマネージャは、ETASのソフトウェアをインストールしたコンピューター上で使用可能になります。

1. Windowsのスタートメニューから**E > ETAS > ETAS License Manager**の順に選択します。
ETAS ライセンスマネージャが開きます。
2. ライセンスマネージャの画面をクリックし、**F1**をクリックします。
ライセンスマネージャのヘルプが表示されます。

INCAのリモート操作

インターネットまたはネットワークアプリケーション（Microsoftリモートデスクトップ、その他の端末／デバイスサービスなど）を介してINCAワークステーションライセンス（マシンベースライセンス）を使用することはできません。ただしこの制限は、テストベンチ用通信プロトコル（ASAP 3、MCD-3 MC、iLinkRT）を介したINCAの操作には適用されません。

ユーザーネームライセンスまたはフローティングライセンス（コンカレントライセンス）を使用する場合は、同時に1人のユーザーが1つのライセンスのみを使用することが保証されている限り、Microsoftリモートデスクトップ経由でINCAにアクセスすることができます。

4 API関数

INCA-MIPは、INCAの処理を自動化するためのさまざまなAPI関数を提供するものです。基本的な関数はINCA-MIP Base（基本セット）に含まれ、その他の関数（拡張コマンド）を使用するには、INCA-MIP Extended（拡張セット）のライセンスをご購入いただく必要があります。

注記

INCA-MIP 拡張セットのみに含まれているコマンドは、ソフトウェアライセンスキーで保護されています。有効なソフトウェアライセンスがない状態で拡張API関数を使用すると、MATLAB®スクリプト実行時に例外が発生します。そのため、スクリプト開発時のガイドラインとして、最初に`IncaIsLicenseValid`関数を呼び出し、有効なライセンスが確認された後にINCA-MIP拡張API関数を呼び出すようにしておくことをお勧めします。

以下の表に、INCA V7.5のアドオン製品に含まれるすべてのINCA-MIP API関数をアルファベット順に一覧表示します。各関数について以下の内容が記載されています。

- － 関数が含まれているセット（基本、拡張）
- － 関数を使用される目的を示すカテゴリ（初期化、測定、適合、メモリページ管理、一般）
- － 関数についての詳細情報が記載されているページ

関数	基本 ^a	拡張 ^b	カテゴリ	ページ
<code>IncaAddCalibrationElement</code>	○	○	適合	ページ49
<code>IncaAddMeasureElement</code>	○	○	測定	ページ32
<code>IncaBrowseCalibrationElements</code>		○	適合	ページ48
<code>IncaBrowseItemsInFolder</code>		○	初期化	ページ26
<code>IncaBrowseMeasureElements</code>		○	測定	ページ31
<code>IncaClose</code>		○	初期化	ページ25
<code>IncaCopyPageFromTo</code>	○	○	メモリページ管理	ページ59
<code>IncaDatabaseImport</code>		○	初期化	ページ25
<code>IncaDownloadDifferences</code>	○	○	メモリページ管理	ページ59
<code>IncaDownloadPage</code>	○	○	メモリページ管理	ページ59
<code>IncaExecuteManualTrigger</code>		○	測定	ページ47

関数	基本 ^a	拡張 ^b	カテゴリ	ページ
IncaGetCalibrationValue	○	○	適合	ページ49
IncaGetCurrentPage		○	メモリページ管理	ページ58
IncaGetDatasetsForDevice		○	適合	ページ55
IncaGetDeviceProperties		○	初期化	ページ29
IncaGetDevices		○	初期化	ページ29
IncaGetHardwareStatus		○	測定	ページ43
IncaGetInstalledAddOnInfo	○	○	一般	ページ22
IncaGetInstalledProductInfo	○	○	一般	ページ22
IncaGetMeasureRatesForDevice		○	測定	ページ31
IncaGetProperties		○	一般	ページ23
IncaGetRecordingMode		○	測定	ページ37
IncaGetRecordingProperties		○	測定	ページ34
IncaGetRecordingState		○	測定	ページ47
IncaGetRecords	○	○	測定	ページ41
IncaGetRecordStruct		○	測定	ページ47
IncaGetVersion	○	○	一般	ページ23
IncaGroupDevices		○	適合	ページ56
IncaIsLicenseValid		○	一般	ページ21
IncaMessageIds	○	○	一般	ページ19
IncaIsPageWriteProtected	○	○	メモリページ管理	ページ58
IncaOpen	○	○	初期化	ページ24
IncaOpenDatabase	○	○	初期化	ページ25
IncaOpenExperiment	○	○	初期化	ページ28
IncaResetExperiment	○	○	初期化	ページ28
IncaResetRecords	○	○	測定	ページ43
IncaSetCalibrationMode		○	適合	ページ56
IncaSetCalibrationValue	○	○	適合	ページ52
IncaSetDatasetInDevice		○	適合	ページ55

関数	基本 ^a	拡張 ^b	カテゴリ	ページ
IncaSetMeasureReadMode	○	○	測定	ページ40
IncaSetProjectAndDatasetInDevice		○	初期化	ページ27
IncaSetRecordingMode		○	測定	ページ38
IncaSetRecordingProperties		○	測定	ページ36
IncaSetTrigger		○	測定	ページ44
IncaShowMessages	○	○	一般	ページ21
IncaStartMeasurement	○	○	測定	ページ33
IncaStartRecording	○	○	測定	ページ39
IncaStopMeasurement	○	○	測定	ページ34
IncaStopRecording	○	○	測定	ページ39
IncaSwitchPage	○	○	メモリページ管理	ページ57
IncaUploadPages		○	メモリページ管理	ページ60
IncaWriteToFile		○	適合	ページ57

a INCA-MIP API基本セットに含まれています。

b INCA-MIP API拡張セットに含まれています。

本章の以降の部分では、各関数を、以下のカテゴリ順に説明します。

- － 「一般的な機能」 (次ページ)
- － 「初期化」 (ページ24)
- － 「測定と記録」 (ページ30)
- － 「適合」 (ページ48)
- － 「メモリページマネージャ」 (ページ57)

さらに、INCA-MIP APIには各種サンプルファイルが含まれています。この内容は「[INCA-MIP APIのサンプルファイル](#)」 (次ページ) で説明されています。

各関数の実際の使用例は、「[応用例](#)」 (ページ60) を参照してください。

注記

INCA-MIPインターフェースは、常にINCAのユーザーオプション設定に基づいて動作します。INCAユーザーオプションについての詳細は、INCAのドキュメントを参照してください。

4.1 INCA-MIP APIのサンプルファイル

INCA-MIPには、数多くのサンプルファイル集が付属しています。製品をインストールすると、これらのサンプルファイルが自動的にMEXファイルと共にPCにコピーされます。サンプルファイルでは、さまざまな例を用いてINCA-MIP APIの使用方法を紹介しています。

サンプルファイルには、INCA-MIP APIを呼び出すMファイルのサンプルや、サンプルスクリプト内で参照されている変数が定義されたINCAデータベースが含まれています。

サンプルファイルは、インストール時に以下のディレクトリにコピーされます。詳しくは、「インストール」(ページ11)を参照してください。

- － MATLAB[®]にインストールした場合：
 - Mファイル: %MatlabDir%\toolbox\matlab\demos
- － ETASDataディレクトリにインストールした場合：
 - Mファイル: %EtasDataDir%\INCA-MIPx64
 - INCAデモファイル: %EtasDataDir%\Database\db_matlabtest

サンプルファイルを実行するには、まずINCAを起動し、サンプルデータベースを開いておく必要があります。この際、ハードウェアは必要ありません。

サンプルのMファイルの機能は以下のとおりです。

- － tOpen.m - INCAとMATLAB[®]間の接続を確立します。この関数は、各MATLAB[®]セッションを開始する際、他のINCA-MIP API関数を使用する前に、必ず呼び出しておく必要があります。
- － tDummy.m - INCA上に、「VADIテストデバイス」を含むハードウェアコンフィギュレーションが定義された空の実験を開きます。さらにこのスクリプトは、INCA実験内にいくつかの測定変数を割り当てます。
- － tEtkDummy.m - INCA上に、「ETKテストデバイス」を含むハードウェアコンフィギュレーションが定義された空の実験を開きます。さらにこのスクリプトは、INCA実験内にいくつかの測定変数と適合変数を割り当てます。そしてワーキングページとリファレンスページをダウンロードして、測定変数と適合変数の値を読み取り、適合変数の値を変更します。
- － tGetRecords(aGroupName).m - aGroupName という名前のグループ (ラスタ) の測定データを20秒間収集し、さらにそのデータをMATLAB[®]で取得します。ここではテストデバイスのVADIとETKが使用されます。(測定ラスタについては「INCA用語集」(ページ8)を参照してください。)
- － tPrintDB({aFolder{, aFileId}}).m - データベース内のフォルダ aFolder のすべての内容をファイル aFileId に書き込みます。引数なしでこの関数を呼び出すと、階層構造になっているデータベース全体が標準出力に出力されます。
- － tHWStatus.m - 関数IncaGetHardwareStatusの使用例です。MATLAB[®]から、INCA上に開いている実験にアクセスし、最初に見つかった測定デバイス内の最初の測定変数を選択します。続いて、測定を5分間行います。測定中にワーニングやエラーが発生すると、測定は中止され、5秒後に再開されます。

4.2 一般的な機能

一般的な機能を持つAPI関数には以下のようなものがあります。

4.2.1 INCA-MIPインターフェースのメッセージID取得

INCA-MIPインターフェースの各コマンドを使用した際に、エラーが返る場合があります。そのような場合、tryブロックとcatchブロックを用いて、詳細なエラー情報を取得することができます。

使用例

```

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

```

各変数の内容は以下のとおりです。

msgstr	メッセージ文字列
msgid	メッセージ識別子 (=エラーID) 以下のIDがあります。
	— INCA:ParameterError
	— INCA:ReturnParameterError
	— INCA:WrongParameterValue
	— INCA:WrongParameterType
	— INCA:NaN
	— INCA:ExecutionError
	— INCA:ResourceError
	— INCA:RasterFull
	— INCA:ObjectIsWriteProtected
	— INCA:CallSequenceError
	— INCA:LicenseError
	— INCA:RecordingInProgress
	— INCA:NotInstalled
	— INCA:WrongVersion

各エラーIDの意味と対処方法は、以下のとおりです。

INCA:ParameterError	入力引数の数が不正です。 (右側のパラメータ)
INCA :ReturnParameterError	出力引数の数が不正です。 (左側のパラメータ)
INCA:WrongParameterValue	いずれかの入力引数の値が有効範囲を超えています。
INCA:WrongParameterType	いずれかの入力引数のデータ型が不正です。
INCA:NaN	いずれかの入力引数の値が "Not a Number" (非数) です。
INCA:ExecutionError	コマンド実行中に何らかの原因でエラーが発生しました。INCAユーザーインターフェース上での操作によって原因を特定できる可能性があります。INCAを再起動するか、またはPCを再起動してください。
INCA:ResourceError	オペレーティングシステムのリソースを取得できません。INCAを再起動するか、またはPCを再起動してください。
INCA:RasterFull	要求された測定ラスタが満杯のため、測定変数を追加できません。
INCA :ObjectIsWriteProtected	要求された適合変数が書き込み禁止のため、適合できません。
INCA:CallSequenceError	要求されたコマンドの前に所定のコマンドを実行する必要があります。 例：IncaAddMeasureElementを実行するには、IncaOpenExperimentを先に実行しておく必要があります。
INCA:LicenseError	要求されたコマンドを実行するには、ライセンスが必要です。

<code>INCA:RecordingInProgress</code>	現在記録が行われているため、要求されたコマンド (<code>IncaSetRecordingMode</code> によるシグナルの記録の有効化/無効化など) を実行できません。
<code>INCA:NotInstalled</code>	<code>IncaOpen</code> コマンドで指定されたINCAバージョンは、インストールされていないため開くことができません。
<code>INCA:WrongVersion</code>	<p><code>IncaOpen</code>コマンドで指定されたINCAバージョンは、以下のいずれかの理由により開くことができません。</p> <ul style="list-style-type: none"> — すでにINCAが起動していて、そのINCAと異なる 'version' 引数と共に <code>IncaOpen</code>コマンドが実行されました。 — INCA Vx.y用INCA-MIPから、メジャーバージョンがxでない 'version' 引数と共に <code>IncaOpen</code>コマンドが実行されました。

4.2.2 スクリプト実行中のメッセージ表示

コマンド名	<code>IncaShowMessages</code>
機能	スクリプト実行中のMATLAB [®] ウィンドウへの情報表示をオン/オフします。
構文	<code>IncaShowMessages(trueOrFalse)</code>
出力引数	なし
入力引数	<code>trueOrFalse</code> 値が0またはそれ以外である数値パラメータです。パラメータの値が0の場合は情報表示が無効、それ以外の場合は有効 (デフォルト) となります。
例	<pre>IncaShowMessages(0); IncaShowMessages;</pre>

4.2.3 有効なINCA-MIPライセンスの確認 (拡張コマンド)

コマンド名	<code>IncaIsLicenseValid</code>
機能	有効なINCA-MIPライセンスが利用可能であるかどうかを示すステータスを返します。
構文	<code>s = IncaIsLicenseValid</code>

出力引数	s	ライセンスステータス： <ul style="list-style-type: none"> － 0：有効なライセンスがない － 1：有効なライセンスがある
入力引数	なし	
例	status = IncaIsLicenseValid	

4.2.4 インストール済みINCAバージョンに関する情報の取得

コマンド名	IncaGetInstalledProductInfo	
機能	インストール済みのすべてのINCAバージョンに関する情報を返します。 このコマンドはIncaOpen実行前に実行することができます。	
構文	info = IncaGetInstalledProductInfo	
出力引数	info	インストール済みの各INCAバージョン についての以下のエントリを含む MATLAB®構造体
	info.name	製品名
	info.version	製品バージョン（文字列）
	info.hotfixVersion	インストール済みホットフィックスバージョン（文字列）※ホットフィックスがインストールされていない場合は空の文字列になります。
入力引数	なし	
例	i = IncaGetInstalledProductInfo;	

4.2.5 インストール済みアドオンに関する情報の取得

コマンド名	IncaGetInstalledAddOnInfo	
機能	指定された製品に対してインストールされているすべてのアドオンに関する情報を返します。 このコマンドはIncaOpen実行前に実行することができます。	
構文	info = IncaGetInstalledAddOnInfo(productName, productVersion)	
出力引数	info	インストール済みの各アドオンについての以下のエントリを含むMATLAB®構造体
	info.name	アドオンの名前
	info.version	アドオンのバージョン（文字列）

入力引数	productName	製品名
	productVersion	製品バージョン（文字列） 完全なバージョン文字列を指定してください。
例	<pre>i = IncaGetInstalledAddOnInfo('INCA', 'V7.5.0'); i = IncaGetInstalledAddOnInfo('INCA', 'V7.5.1 Build 100');</pre>	

**注記**

入力引数productNameおよびproductVersionには、必ずIncaGetInstalledProductInfoで返される名前とバージョンを使用してください。

4.2.6 INCAバージョンの取得

コマンド名	IncaGetVersion
機能	INCAのバージョンを返します。
構文	IncaGetVersion
出力引数	version INCAのバージョン（文字列）
入力引数	なし
例	IncaGetVersion;

4.2.7 INCAプロパティの取得（拡張コマンド）

コマンド名	IncaGetProperties
機能	INCAの各種プロパティを返します。
構文	p = IncaGetProperties

出力引数	p	INCAのプロパティについての以下のエントリを含むMATLAB®構造体 <ul style="list-style-type: none"> — p.databasePath - 現在INCA上に開いているデータベースが保存されているパス。INCAデータベースが開いていない場合は、空の文字列を返します。 — p.dataPath : INCAデータディレクトリのパス — p.installationPath : INCAのインストールディレクトリのパス — p.tempPath : ETASアプリケーションで使用される一時ファイルが格納されるパス
入力引数	なし	
例	p = INCAGetProperties;	

4.3 初期化

INCAで実行される測定および適合処理は、「実験」というデータベースアイテムに基づいて実行されます。実験を開くには、適切なハードウェアコンフィギュレーションが定義された「ワークスペース」を作成し、そのワークスペースに実験を割り当てておく必要があります。

INCA-MIP APIを使用するには、有効なワークスペースとハードウェアコンフィギュレーションにリンクされた空の実験がデータベースに存在する必要があります。実験は、MATLAB®から開くことができます。

INCAで実験を行う際の一連の初期化処理には、以下のAPI関数を使用します。

4.3.1 INCAを開く

コマンド名	IncaOpen	
機能	INCAを起動し、MATLAB®とINCAとの接続を初期化します。	
構文	IncaOpen IncaOpen(version)	
出力引数	なし	
入力引数	version	起動するINCAのバージョン (オプション) 構文 : < MajorVersion>.<MinorVersion>. INCA-MIP x.y用のINCAの場合、接続できるINCAのバージョンは x (つまりメジャーバージョンが同じもの) だけです。
例	IncaOpen; IncaOpen('7.3');	

4.3.2 INCAを閉じる（拡張コマンド）

コマンド名	IncaClose	
機能	IncaOpenでINCAを開いた後、INCAとの接続を切断するにはこのコマンドを使用します。さらに、INCAを閉じるか、それとも起動したままにしておくかを選択できます。	
構文	IncaClose IncaClose(isDisconnectOnly)	
出力引数	なし	
入力引数	isDisconnectOnly	MATLAB [®] をINCAから切断した後にINCAを閉じるかどうかを指定します。値は以下のいずれかです。 — 0: INCAを閉じる（デフォルト） — 1: INCAを閉じない
例	INCAclose; INCAclose(1);	

4.3.3 データベースを開く

コマンド名	IncaOpenDatabase	
機能	指定のディレクトリに保存されているINCAデータベースを開きます。	
構文	IncaOpenDatabase({pathName})	
出力引数	なし	
入力引数	pathName	開くべきデータベースが保存されているディレクトリ。ディレクトリが指定されていない場合は、カレントデータベース（前回開いていたデータベース）が開きます。
例	IncaOpenDatabase;% open current database IncaOpenDatabase('c:\etasdata\mydatabase');	

4.3.4 データベースのインポート（拡張コマンド）

コマンド名	IncaDatabaseImport	
機能	エクスポートファイル (*.exp) にエクスポートされているデータベースをINCAにインポートします。この際、INCA上に開いている同名のデータベースアイテムは必ず上書きされます。	
構文	IncaDatabaseImport(path) name = IncaDatabaseImport(path) [name, type] = IncaDatabaseImport(path)	

出力引数	name	<p>インポートされたすべてのデータベースアイテムの名前（フルパス）の配列</p> <p>配列の要素にアクセスするには、deblank（）を使用してください。例： name2 = deblank(name(2, :))</p>
	type	<p>インポートされたすべてのデータベースアイテムのタイプの配列。</p> <p>値は以下のいずれかです。</p> <ul style="list-style-type: none"> － Folder：データベースフォルダ － Experiment：実験 － Workspace：ワークスペース － Asap2Project：ASAP2プロジェクト － MeasurementCatalog：測定カタログ（ASAP2フォーマット） － CanDB：CAN DBファイル（ASAP2フォーマット） <p>配列の要素にアクセスするには、deblank（）を使用してください。例： type2 = deblank(type(2, :))</p>
入力引数	path	インポートする*.expファイルの名前（フルパス）
例	<pre>names = IncaDatabaseImport('D:\ ETASData\INCA7.5\export\Project0815.exp')</pre>	

4.3.5 データベースアイテムの読み込み（拡張コマンド）

コマンド名	IncaBrowseItemsInFolder	
機能	検索パターンに一致するデータベースアイテムを、指定のデータベースフォルダから検索して読み込みます。	
構文	[name, type] = IncaBrowseItemsInFolder (pattern, folderName)	
出力引数	name	データベースアイテム名のリスト
	type	<p>データベースアイテムのタイプのリスト：</p> <ul style="list-style-type: none"> － Folder：データベースフォルダ － Experiment：実験 － Workspace：ワークスペース － Asap2Project：ASAP2プロジェクト

入力引数	<code>pattern</code>	データベースアイテムを指定するための検索文字列です。'*' は0または1つ以上の任意の文字を表します。'#' は1つの任意の文字を表します。これらのワイルドカードで指定されるもの以外の文字は、すべて正確に一致する必要があります。検索文字列内の大文字と小文字は区別されません。
	<code>folderName</code>	データベースアイテムの読み込み先のデータベースフォルダフォルダの階層は '\' で区切ります。最上位レベルに読み込む場合は、この引数は空にしておいてください。
例	<pre>[n,t]=IncaBrowseItemsInFolder('*','DEFAULT\MyProject'); [name,type]=IncaBrowseItemsInFolder('Prj*_##','');</pre>	

4.3.6 デバイスにプロジェクトとデータセットを割り当てる（拡張コマンド）

コマンド名	<code>IncaSetProjectAndDatasetInDevice</code>	
機能	指定のワークスペース内のデバイスに、プロジェクトとデータセットを割り当てます。これは、現在どの実験も開かれていない場合にのみ可能です。	
構文	<code>IncaSetProjectAndDatasetInDevice(workspace, device, project, dataset)</code>	
出力	なし	
入力引数	<code>workspace</code>	ワークスペースのデータベースパス
	<code>device</code>	デバイス名
	<code>project</code>	プロジェクトのデータベースパス
	<code>dataset</code>	データセットのデータベースパス
例	<pre>IncaSetProjectAndDatasetInDevice('DEFAULT\workspace','ETK:1','DEFAULT\Prj0815','Ds4711\Ds47-11_3')</pre>	

4.3.7 実験を開く

コマンド名	<code>IncaOpenExperiment</code>	
機能	指定の実験（実験環境）を開きます。INCA-MIP APIを使用して測定変数や適合変数を実験に割り当てるには、このコマンドで実験を開いておく必要があります。	
構文	<pre>IncaOpenExperiment ({closeAllViewsFlag})</pre> <p>または</p> <pre>IncaOpenExperiment (expFolderName, experimentName, workspaceFolderName, workspaceName {, closeAllViewsFlag})</pre>	
出力引数	なし	
入力引数	<code>expFolderName</code>	実験が保存されているフォルダ名
	<code>experimentName</code>	実験名
	<code>workspaceFolderName</code>	ワークスペースが保存されているフォルダ名
	<code>workspaceName</code>	ワークスペース名
	<code>closeAllViewsFlag</code>	指定の実験に含まれるすべての測定／適合ウィンドウを閉じます。以下の設定が有効です。 <ul style="list-style-type: none"> — 1 : 全ウィンドウを閉じる（デフォルト） — 0 : ウィンドウを開いたままの状態にする
例	<pre>IncaOpenExperiment ('ExpFolder', 'MyExperiment', 'WorkspaceFolder', 'MyWorkspace');</pre>	



注記

`IncaOpenExperiment`関数を呼び出す時点ですでに実験が開いている場合は、環境を指定する入力引数は省略できます。

実験がまだ開いていない場合は、`IncaOpenExperiment`の前に`IncaOpenDatabase`を実行しておく必要があります。

4.3.8 実験のリセット

コマンド名	<code>IncaResetExperiment</code>
機能	現在開いている実験をリセットして閉じます。実験から変数をすべて削除します。個々の変数の削除はサポートされていません。
構文	<code>IncaResetExperiment</code>

出力引数	なし
入力引数	なし
例	<code>IncaResetExperiment;</code>

**注記**

MATLAB[®]コマンドではなくINCA上での手動操作により実験を開いていた場合、`IncaResetExperiment`コマンドを発行すると、実験はリセットされますが、実験ウィンドウは開いたままの状態となります。この状態において再度MATLABから実験にアクセスするには、`IncaOpenExperiment`コマンドを発行する必要があります。

4.3.9 デバイスの取得（拡張コマンド）

コマンド名	<code>IncaGetDevices</code>				
機能	実験内のすべてのデバイス名を返します。				
構文	<code>[name, type] = IncaGetDevices</code>				
出力引数	<table> <tr> <td><code>name</code></td> <td>デバイス名のリスト</td> </tr> <tr> <td><code>type</code></td> <td> デバイスタイプのリスト： <ul style="list-style-type: none"> — <code>WorkbaseDevice</code>: データセットが割り当てられたデバイス — <code>MeasurementDevice</code>: 測定デバイス </td> </tr> </table>	<code>name</code>	デバイス名のリスト	<code>type</code>	デバイスタイプのリスト： <ul style="list-style-type: none"> — <code>WorkbaseDevice</code>: データセットが割り当てられたデバイス — <code>MeasurementDevice</code>: 測定デバイス
<code>name</code>	デバイス名のリスト				
<code>type</code>	デバイスタイプのリスト： <ul style="list-style-type: none"> — <code>WorkbaseDevice</code>: データセットが割り当てられたデバイス — <code>MeasurementDevice</code>: 測定デバイス 				
入力引数	なし				
例	<code>[name,type]=IncaGetDevices;</code>				

4.3.10 デバイスプロパティの取得（拡張コマンド）

コマンド名	<code>IncaGetDeviceProperties</code>						
機能	デバイスのプロパティを返します。						
構文	<code>p = IncaGetDeviceProperties (deviceName)</code>						
出力引数	<table> <tr> <td><code>p</code></td> <td>MATLAB[®]構造体で表されたデバイスプロパティ。以下のエントリで構成されます。</td> </tr> <tr> <td><code>p.name</code></td> <td>デバイス名</td> </tr> <tr> <td><code>p.descriptionFile</code></td> <td>デバイスに割り当てられているプロジェクトのディスクリプションファイルのパス名を指定します。デバイスにプロジェクトが割り当てられていない場合は空の文字列を返します。</td> </tr> </table>	<code>p</code>	MATLAB [®] 構造体で表されたデバイスプロパティ。以下のエントリで構成されます。	<code>p.name</code>	デバイス名	<code>p.descriptionFile</code>	デバイスに割り当てられているプロジェクトのディスクリプションファイルのパス名を指定します。デバイスにプロジェクトが割り当てられていない場合は空の文字列を返します。
<code>p</code>	MATLAB [®] 構造体で表されたデバイスプロパティ。以下のエントリで構成されます。						
<code>p.name</code>	デバイス名						
<code>p.descriptionFile</code>	デバイスに割り当てられているプロジェクトのディスクリプションファイルのパス名を指定します。デバイスにプロジェクトが割り当てられていない場合は空の文字列を返します。						

<code>p.binaryFile</code>	デバイスに割り当てられているプロジェクトのバイナリファイルのパス名を指定します。デバイスにプロジェクトが割り当てられていない場合は空の文字列を返します。
<code>p.projectDBPath</code>	デバイスに割り当てられているプロジェクトのINCAデータベース内のパス名を指定します。デバイスにプロジェクトが割り当てられていない場合は空の文字列を返します。
<code>p.isWriteProtected</code>	<ul style="list-style-type: none"> — 0: デバイ스에 메모리 페이지가 존재하지 않거나, 또는 현재의 페이지가 쓰기 금지되어 있지 않습니다. — 1: 현재의 페이지는 쓰기 금지입니다.
<code>p.isActive</code>	<ul style="list-style-type: none"> — 0: デバイ스가接続されていないか、または稼働状態になっていません。 — 1: デバイ스가接続されていて、稼働状態になっています。
<code>p.isWorkbaseDevice</code>	<ul style="list-style-type: none"> — 0: デ바이스에 데이터 세트가 할당되어 있지 않습니다. — 1: 데바이스에 데이터 세트가 할당되어 있습니다.
入力引数	<code>deviceName</code> デバイス名
例	<code>p = IncaGetDeviceProperties ('Device');</code>

4.4 測定と記録

シグナル (= 測定変数) は、ある測定デバイス用に定義された「測定ラスタ」の一部として取り込まれます。1つの測定変数は、いずれか1つの測定ラスタに属します。実験の設定を行うには、最初に測定変数を個々の測定ラスタに割り当てる必要があります。

注記

変数、デバイス、シグナル、測定ラスタの名前は、大文字と小文字が区別されます。

4.4.1 測定変数の取得（拡張コマンド）

コマンド名	IncaBrowseMeasureElements	
機能	実験に割り当てられている測定変数の名前とタイプを、検索パターンとデバイスを指定して取得します。	
構文	<pre>[name, type] = IncaBrowseMeasureElements (pattern, {deviceName}) [name] = IncaBrowseMeasureElements(pattern, {deviceName})</pre>	
出力引数	name	測定変数名のリスト
	type	測定変数のタイプのリスト: <ul style="list-style-type: none"> - Scalar: スカラ値 - Array: 配列 - Matrix: 行列
入力引数	pattern	測定変数を指定するための検索文字列です。'*' は0または1つ以上の任意の文字を表します。'#' は1つの任意の文字を表します。これらのワイルドカードで指定されるもの以外の文字は、すべて正確に一致している必要があります。検索文字列内の大文字と小文字は区別されません。
	deviceName	デバイス名
例	<pre>[n,t]=IncaBrowseMeasureElements('ign*', 'Device'); [name,type]=IncaBrowseMeasureElements('*');</pre>	

4.4.2 測定ラスタの取得（拡張コマンド）

コマンド名	IncaGetMeasureRatesForDevice	
機能	指定されたデバイスのすべての測定ラスタを返します。	
構文	<pre>[name] = IncaGetMeasureRatesForDevice (deviceName)</pre>	
出力引数	name	測定ラスタ名のリスト
入力引数	deviceName	デバイス名
例	<pre>n=IncaGetMeasureRatesForDevice('Device'); name=IncaGetMeasureRatesForDevice('Dev');</pre>	

4.4.3 測定変数を実験に追加する

コマンド名	IncaAddMeasureElement	
機能	測定変数（測定ラスタの指定あり／なし）を実験に追加します。	
構文	<pre>IncaAddMeasureElement(deviceName, groupName, signalName {, displayMode}) groupName = IncaAddMeasureElement(deviceName, [], signalName{, displayMode})</pre>	
出力引数	なし	
入力引数	deviceName	デバイス名
	groupName	測定ラスタの名前を指定します。 ラスタ名を '+' という文字で連結することにより、複数のラスタを指定できます (例: '10ms+100ms')。このような「マルチラスタ」を指定すると、新しい仮想ラスタが生成されます。 1つのシグナルの測定は、1つのラスタ、またはマルチラスタのいずれかでのみ行えます。 グループ名を [] (空白) にすることもできます (以下の注記を参照してください)。
	signalName	測定シグナルの名前です。スカラタイプの場合には変数名のみで、配列タイプの場合には変数名にインデックス ([n] または [n,m]) を付加します。1番目の要素のインデックスは0 (ゼロ) です。
	displayMode	変数の表示モード - 1: 表示 (デフォルト) - 0: 非表示
例	<pre>IncaAddMeasureElement('MyDevice', '10ms', 'Channel01', 0); IncaAddMeasureElement('ETK:1', '1.0ms', 'Matrix[2,1]'); group = IncaAddMeasureElement('CalcDev', [], 'MyCalcSig1');</pre>	



注記

測定ラスタがフル (=満杯状態) になると、測定変数はその測定ラスタに追加されません。

i 注記

入力引数groupNameを[]（空）にすると、以下のような方法でシグナルグループが確定されます。

- すでに実験に含まれているシグナルの場合は、実験で指定されているシグナルグループ名が使用されます。
- 実験に含まれていないシグナルの場合は、使用可能なシグナルグループのいずれかが使用されます。演算シグナル（デバイス名：CalcDev）またはCANモニタリングの場合、そのシグナル用に定義されているシグナルグループが使用されます。

IncaGetRecords、IncaGetRecordStruct、IncaGetRecordCountを呼び出す際にはシグナルグループ名が必要となるので、オプション引数としてシグナルグループが返されます。

例:

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

i 注記

追加できるシグナルの総数は、デバイスやプロトコルに応じて異なります。シグナル数は、ターゲットサーバーがアロケートするバッファの空き容量によって制限されます。各シグナルに必要なバッファメモリのサイズはサンプリングレートに依存します。

例:

サンプリングレートが0.1msのシグナルの場合は、3メガバイト以上のデータメモリを必要とするため、追加できるシグナルの総数は400～600となります。サンプリングレートを遅くすれば、より多くのシグナルを追加できます。

4.4.4 測定の開始

名前	IncaStartMeasurement
機能	INCAの測定処理を開始します。
構文	IncaStartMeasurement
出力引数	なし
入力引数	なし
例	IncaStartMeasurement;

4.4.5 測定終了

コマンド名	IncaStopMeasurement	
機能	INCAの測定処理を終了し、記録が行われている場合は記録も終了します。	
構文	IncaStopMeasurement { (mdfFileName) }	
出力引数	なし	
入力引数	mdfFileName	測定とともに記録が終了した場合に、記録されたデータが保存されるMDFファイルの名前。ファイルへの完全なパスを指定してください (例 : 'c:\mydata\store1.dat') 。
例	IncaStopMeasurement ('c:\mydata\store1.dat');	



注記

長時間の測定によりデータが失われるのを避けるため、データ量が多くなってきたら、IncaStopMeasurement (mdfFileName) コマンドで一旦、記録を終了してください。その後、続けて残りのデータをIncaGetRecordsコマンドでMATLAB[®]に転送してください。

4.4.6 記録のプロパティの取得 (拡張コマンド)

コマンド名	IncaGetRecordingProperties	
機能	デフォルトレコーダのメインの出力ファイルのプロパティや、第2の記録フォーマットなどを返します。	
構文	properties = IncaGetRecordingProperties	
出力引数	properties	記録のプロパティについての以下のエントリを含むMATLAB [®] 構造体
	properties.fileName	メインのファイル拡張子を含む記録ファイル名
	properties.directory	測定ファイルのディレクトリ

<code>properties.fileFormat</code>	第2の測定ファイルのフォーマットを表す以下のいずれかの文字列 <ul style="list-style-type: none"> — ETASBinary — #DiademATF — ETASAscii — ETASGroupAscii — ETASMATLABMFILE — ETASGroupMatlabM — FamosRecord — ETAS MDF — ETAS MDF4
<code>properties.autoIncrement</code>	測定ファイル名の自動インクリメントモード <ul style="list-style-type: none"> — 0 : インクリメントしない — 1 : 自動的にインクリメントする
<code>properties.comment</code>	測定ファイルのヘッダ部分のコメント。使用できる文字数は、1024からデフォルトコメントの文字数を引いた数です。
<code>properties.defaultComment</code>	INCAによって測定ファイルのヘッダに挿入されるデフォルトコメント
<code>properties.company</code>	測定ファイルのヘッダに挿入される会社名
<code>properties.project</code>	測定ファイルのヘッダに挿入されるプロジェクト名
<code>properties.user</code>	測定ファイルのヘッダに挿入されるユーザー名
<code>properties.vehicle</code>	測定ファイルのヘッダに挿入される車両名

入力 なし

引数

例 `properties = IncaGetRecordingProperties;`

4.4.7 記録のプロパティの設定（拡張コマンド）

コマンド名	IncaSetRecordingProperties	
機能	次回の記録処理に適用されるプロパティを設定します。	
構文	IncaSetRecordingProperties (properties)	
出力	なし	
引数		
入力引数	properties	記録のプロパティに関する以下のエンタリで構成されるMATLAB構造体
	properties.fileName	データを記録するメイン測定ファイルの名前
	properties.directory	測定ファイルのディレクトリ
	properties.fileFormat	第2の測定ファイルのフォーマットを表す以下のいずれかの文字列
		<ul style="list-style-type: none"> — ETASBinary — #DiademATF — ETASAscii — ETASGroupAscii — ETASMATLABMFILE — ETASGroupMatlabM — FamosRecord — ETAS MDF — ETAS MDF4
	properties.autoIncrement	測定ファイル名の自動インクリメントモード
		<ul style="list-style-type: none"> — 0 : インクリメントしない — 1 : 自動的にインクリメントする
	properties.comment	測定ファイルのヘッダ部分のコメント※使用できる文字数は、1024 からデフォルトコメントの文字数を引いた数です。
	properties.company	測定ファイルに挿入される会社名
	properties.project	測定ファイルに挿入されるプロジェクト名

<code>properties.user</code>	測定ファイルに挿入されるユーザー名
<code>properties.vehicle</code>	測定ファイルに挿入される車両名

例

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

注記

`IncaSetRecordingProperties`で記録のプロパティを設定する際は、`IncaStopRecording`で記録を終了しないでください。代わりに`IncaSetTrigger`を使用してください。任意のトリガ条件を指定して実験を終了することができます。

例 :

一定の時間経過後、記録を終了する :

```
TIMEDURATION_SECONDS = 25;
IncaSetTrigger('none', 'none', 'none', 'none',
TIMEDURATION_SECONDS);
IncaStartRecording;
% Recording automatically stops after TIMEDURATION_
SECONDS seconds
```

手動トリガによって記録を終了する :

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

4.4.8 記録モードの取得 (拡張コマンド)

コマンド名 `IncaGetRecordingMode`

機能 指定したシグナルの記録モード (デフォルトレコーダで記録するかどう
か) を返します。

構文 `IncaGetRecordingMode(deviceName, signalName)`

出力引数

デフォルトレコーダでの記録モード :

- 0 : 指定のシグナルはデフォルトレコーダで記録されません。
- 1 : 指定のシグナルはデフォルトレコーダで記録されます。

入力引数	deviceName	デバイス名
	signalName	測定シグナルの名前です。スカラタイプの場合は変数名のみで、配列タイプの場合は変数名にインデックス ([n] または [n, m]) を付加します。1番目の要素のインデックスは0 (ゼロ) です。
例	<pre>m = IncaGetRecordingMode('ETK:1', 'hfm'); mode = IncaGetRecordingMode('CalcDev', 'MyCalcSig1');</pre>	

注記

IncaGetRecordingModeを使用するには、あらかじめIncaAddMeasureElementでシグナルを実験に割り当てておく必要があります。

4.4.9 記録モードの設定 (拡張コマンド)

コマンド名	IncaSetRecordingMode	
機能	<p>指定したシグナルの記録モード (デフォルトレコーダで記録するか) を設定します。記録できるのは、INCAの実験に割り当てられているシグナルのみです。</p> <p>このコマンドを実行するには、あらかじめIncaAddMeasureSignalでシグナルを実験に割り当てておく必要があります。</p>	
構文	IncaSetRecordingMode(deviceName, signalName, recordingMode)	
出力引数	なし	
入力引数	deviceName	デバイス名
	signalName	測定シグナルの名前です。スカラタイプの場合は変数名のみで、配列タイプの場合は変数名にインデックス ([n] または [n, m]) を付加します。1番目の要素のインデックスは0 (ゼロ) です。
	recordingMode	<p>デフォルトレコーダでの記録モード：</p> <ul style="list-style-type: none"> — 0 : 指定のシグナルがデフォルトレコーダから削除されます。 — 1 : 指定のシグナルがデフォルトレコーダに追加されます。
例	<pre>IncaSetRecordingMode('ETK:1', 'hfm', 1); IncaSetRecordingMode('CalcDev', 'MyCalcSig1', 0);</pre>	

 **注記**

IncaSetRecordingModeを使用するには、あらかじめ
IncaAddMeasureElementでシグナルを実験に割り当てておく必要があります。

4.4.10 記録の開始

コマンド名	IncaStartRecording
機能	INCAでの測定データの記録処理を開始します。このコマンドを実行する際には、IncaStartMeasurementを実行する必要はありません。 測定または記録処理が開始されると、MATLAB [®] から測定データを読み込むことが可能となります。
構文	IncaStartRecording
出力引数	なし
入力引数	なし
例	IncaStartRecording;

4.4.11 記録の終了

コマンド名	IncaStopRecording
機能	INCAでの測定データの記録処理を終了します。測定はそのまま続行されるので、測定処理の終了はIncaStopMeasurementコマンドで明示的に行う必要があります。このコマンドを利用して、測定中の記録の開始／終了を任意に制御することができます。
構文	IncaStopRecording (mdfFileName)
出力引数	なし
入力引数	mdfFileName 記録されたデータが保存されるMDFファイルの名前を指定します。ファイルへの完全なパスを指定してください (例： 'c:\mydata\store1.dat')。
例	IncaStopRecording('c:\mydata\store1.dat');

4.4.12 データの読み込みモードの設定（オンライン・オフライン）

コマンド名 IncaSetMeasureReadMode

機能 どのデータソース（オンライン／オフライン）の測定データをMATLAB[®]に送信するかを指定します。ここでは、INCAが測定したデータをオフラインデータとして蓄積し、後からMATLAB[®]に転送するか、または各測定デバイスのバッファから直接MATLABにオンラインデータとして転送するかを選択できます。

ただしES1303やES6xx 測定デバイスなどのように、測定データの画面表示のみが行われている時にはオフラインデータを蓄積できないデバイスがあります。これらのデバイスの測定時には、オンラインデータを取り込むようにしてください。

測定データの記録時にはオンラインデータとオフラインデータのいずれかを使用することができます。どちらの場合も、最適化された送信機能による特徴的な処理の違いがみられます。オンラインデータは高負荷時に内容が不完全なものになることがありますが、オフラインデータは必ず完全なデータとして記録されます。ただし高負荷時におけるオフラインデータの送信には、一定の遅れが生じます。測定データを記録している間は、オフラインデータのみを使用することをお勧めします。

構文 IncaSetMeasureReadMode (measureReadMode)

出力引数 なし

入力引数 measureReadMode データソースを指定する数値。以下の設定が有効です。

- － 1: オフラインデータ
- － 0: オンラインデータ (デフォルト)

例 IncaSetMeasureReadMode (0) ;

4.4.13 測定データの取得

コマンド名	IncaGetRecords	
機能	<p>測定データをMATLAB®に転送します。各シグナルグループの測定データは、それぞれ専用のリングバッファに保存され、各リングバッファは最大約30秒間分の測定データを保持することができます。測定データはグループごとにMATLAB®に転送されます。測定データを取り込んだ後はMATLAB®スクリプトの実行を停止するようにしてください。一度に転送されるデータが多いほどデータ転送効率が向上します。この関数は、指定された測定ラスタについて指定された数のデータレコードを転送します。</p> <p>リングバッファについては、「INCA用語集」(ページ8)で詳しく説明されています。</p>	
構文	<pre>[time, data {, state}] = IncaGetRecords (deviceName, groupName, maxRecords { {, latest {, exact}}})</pre>	
出力引数	time	<p>転送されたレコードのタイムスタンプが格納される配列。要素の最大数をmとすると、必ず $m \leq \text{maxRecords}$ でなければなりません。</p>
	data	<p>各測定変数の値が含まれる2次元配列。含まれる変数の順は <code>IncaAddMeasureElement</code> で実験に追加された順と同じです。この2次元配列では、mが受信するレコードの数、nが測定ラスタの数です。</p>
	state	<p>以下の状態を表すオプション引数：</p> <ul style="list-style-type: none"> — 0: 成功。データレコードを正しく受信しました。 — 1: 測定が実行されていません。データレコードを受信しませんでした。 — 2: データレコードが足りません。データレコードをまったく受信していません。この値を返すのは <code>exact = 1</code> の場合のみです。
入力引数	deviceName	デバイス名
	groupName	<p>測定ラスタ名</p> <p>ラスタ名を '+' という文字で連結することにより、複数のラスタを指定できます (例: '10ms+100ms')。このような「マルチラスタ」を指定すると、新しい仮想ラスタが生成されます。</p> <p>1つのシグナルの測定は、1つのラスタ、またはマルチラスタのいずれかでのみ行えます。</p>

maxRecords	受信するレコードの最大数（下記の引数 exact も参照してください）。ここにはtimeまたはdataのサイズ m を指定してください。mが maxRecords の値より大きい場合、一部のレコードが読み込まれず、リングバッファがオーバーフローしてしまう可能性があります。
latest	最新データから受け取るか、または最も古いデータから受け取るかを指定します。
一致	リングバッファ内のデータレコード数が maxRecordsより少ない場合でもデータを受け取るようにするかどうかを指定します。

例

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

より大きなサイズのデータを転送する場合は、「例2」（ページ60）を参照してください。

注記

IncaGetRecordsに渡すラスタは、以下の例のように必ず IncaAddMeasureElementに渡したラスタまたはマルチラスタと同じものを使用してください。

例：

```
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)
```

ラスタに割り当てられているシグナルは、IncaGetRecordStructを使用して以下のように確認できます。

例：

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

注記

2つの入力引数latestおよびexactの組み合わせにより、以下のように処理されます。

- latest = 0, exact = 0: (default)
リングバッファ内の最も古いレコードから順にmaxRecords個のレコードを返します。それより新しいレコードは保持されます。
- latest = 1, exact = 0:
リングバッファ内の最新のレコードから順にmaxRecords個のレコードを返します。それより古いレコードは意図的に破棄されます。
- latest = 0, exact = 1:
リングバッファ内の最も古いレコードから順にmaxRecords個のレコードを返します。それより新しいレコードは保持されます。リングバッファ内のレコード数がmaxRecordsより少ない場合、何も受信しません。
- latest = 1, exact = 1:
リングバッファ内の最新レコードから順にmaxRecords個のレコードを返します。それより古いレコードは意図的に破棄されます。リングバッファ内のレコード数がmaxRecordsより少ない場合、何も受信しません。

4.4.14 リングバッファのリセット

コマンド名	IncaResetRecords
機能	すべてのシグナルグループ用のリングバッファをリセットします。測定中であっても、このコマンドを実行してすべてのリングバッファをリセットすることができます。測定または記録を開始した時点においてリングバッファは自動的にリセットされるので、通常は、この関数を使用する必要はありません。 リングバッファについては、「 INCA用語集 」(ページ8)で詳しく説明されています。
構文	IncaResetRecords
出力引数	なし
入力引数	なし
例	IncaResetRecords;

4.4.15 ハードウェアステータスの取得 (拡張コマンド)

コマンド名	IncaGetHardwareStatus
機能	測定中または記録中のハードウェアステータスを返します。
構文	[status, message] = IncaGetHardwareStatus
出力引数	status ハードウェアの現在のステータス <ul style="list-style-type: none"> - 0: OK - 1: ワーニング状態 - 2: エラー状態

message statusの値が1または2の場合は、ワーニングまたはエラーの内容を示すテキストを返します。

statusの値が1または2の場合は、測定または記録を終了させてからでないと、再度IncaGetHardwareStatusを実行することはできません。

使用例は、「[INCA-MIP APIのサンプルファイル](#)」(ページ18)に記述されているサンプルスクリプトtHWStatus.mを参照してください。

入力引数 なし

例 [s, m] = IncaGetHardwareStatus;

4.4.16 トリガの設定 (拡張コマンド)

コマンド名 IncaSetTrigger

機能 記録の開始/終了トリガを設定します。この設定はIncaStartMeasurementまたはIncaStartRecordingコマンドで測定または記録を開始する前に行います。

構文 IncaSetTrigger(startTrigger{, stopTrigger{, preTriggerTime{, postTriggerTime{, duration}}}))

出力引数 なし

入力引数

startTrigger	開始時のトリガ条件。 <ul style="list-style-type: none"> — manual : 手動トリガで開始 — none : トリガを使用しない
stopTrigger	終了時のトリガ条件。 <ul style="list-style-type: none"> — manual : 手動トリガで終了 — none : トリガを使用しない (デフォルト)
preTriggerTime	プリトリガ時間 (単位: 秒) <ul style="list-style-type: none"> — none : 設定しない (デフォルト)
postTriggerTime	ポストトリガ時間 (単位: 秒) <ul style="list-style-type: none"> — none : 設定しない (デフォルト)
duration	記録を行う時間 (単位: 秒) <ul style="list-style-type: none"> — none : 設定しない (デフォルト) — この場合、記録時間は無制限になります。

例

```
IncaSetTrigger('nmot\ETK:1 > 2000', 'none',
2.0, 3.0)
IncaSetTrigger('none', 'none', 'none',
'none', 360)
```

次の表に、入力引数の値として有効な組み合わせを示します。他の組み合わせを使用すると例外が発生します。

入力引数の組み合わせ

トリガの種類	startTrigger	stopTrigger	preTriggerTime	postTriggerTime	duration
記録時間を指定	'none'	'none'	'none'	'none'	value
手動トリガで記録開始、プリトリガ時間を指定、手動トリガで記録終了	'manual'	'manual'	value	'none'	'none'
手動トリガで記録開始、プリトリガ時間とポストトリガ時間を指定	'manual'	'none'	value	value	'none'
手動トリガで記録開始、プリトリガ時間と終了トリガ条件を指定	'manual'	value	value	'none'	'none'
手動トリガで記録終了	'none'	'manual'	'none'	'none'	'none'
開始トリガ条件と記録時間を指定	value	'none'	'none'	'none'	value
開始トリガ条件を指定し、手動トリガで記録終了	value	'manual'	'none'	'none'	'none'
開始トリガ条件とプリトリガ時間を指定し、手動トリガで記録終了	value	'manual'	value	'none'	'none'
開始トリガ条件を指定し、プリトリガ時間とポストトリガ時間を指定	value	'none'	value	value	'none'
開始トリガ条件を指定し、プリトリガ時間と終了トリガ条件を指定	value	value	value	'none'	'none'

4.4.17 手動トリガの発行（拡張コマンド）

コマンド名	IncaExecuteManualTrigger	
機能	記録を開始／終了するための手動トリガを発行します。ただしこのコマンドが有効なのは、前もってIncaSetTriggerコマンドでstartTriggerパラメータまたはstopTriggerパラメータの値がmanualに設定されている場合のみです。	
構文	IncaExecuteManualTrigger (type)	
出力引数	なし	
入力引数	type	トリガタイプ <ul style="list-style-type: none"> － start：記録開始の手動トリガを発行 － stop：記録終了の手動トリガを発行
例	IncaExecuteManualTrigger ('start')	

4.4.18 記録ステータスの取得（拡張コマンド）

コマンド名	IncaGetRecordingState	
機能	現在の記録ステータスを返します。	
構文	result = IncaGetRecordingState	
出力引数	result	記録のステータス <ul style="list-style-type: none"> － 0：非アクティブ － 1：トリガ待ち状態、または記録中
入力引数	なし	
例	s = IncaGetRecordingState	

4.4.19 測定変数リストの取得（拡張コマンド）

コマンド名	IncaGetRecordStruct	
機能	測定または記録用として割り当てられている測定変数のリストを返します。リストに含まれる測定変数の並び順は、IncaAddMeasureElementコマンドで測定変数を割り当てた際の並び順と同じです。	
構文	list = IncaGetRecordStruct (device, groupName)	
出力引数	なし	
入力引数	device	デバイス名
	groupName	測定ラスタ名 ラスタ名を '+' という文字で連結することにより、複数のラスタを指定できます (例：'10ms+100ms')。
例	<pre>l = IncaGetRecordStruct ('ETK:1', '10ms'); list = IncaGetRecordStruct ('device1', 'Syncro');</pre>	

4.5 適合

適合 (=値の編集) は、「スカラ値」、「特性カーブ」、「特性マップ」、さらにカーブやマップが参照する「軸ポイントディストリビューション」について行うことができます。1つの実験には任意の数の適合変数を割り当てることができます。



注記

適合変数名は、大文字と小文字が区別されます。

4.5.1 適合変数の取得 (拡張コマンド)

コマンド名	<code>IncaBrowseCalibrationElements</code>				
機能	実験に割り当てられている適合変数の名前とタイプを、検索パターンとデバイスを指定して取得します。				
構文	<pre>[name, type] = IncaBrowseCalibrationElements (pattern, {deviceName}) name = IncaBrowseCalibrationElements (pattern, {deviceName})</pre>				
出力引数	<table border="0"> <tr> <td><code>name</code></td> <td>適合変数名のリスト</td> </tr> <tr> <td><code>type</code></td> <td> 適合変数のタイプのリスト : <ul style="list-style-type: none"> - <code>Distribution</code>: 軸ポイントディストリビューション - <code>OneDTable</code>: カーブ - <code>TwoDTable</code>: マップ - <code>Scalar</code>: スカラ値 - <code>Array</code>: 配列 - <code>Matrix</code>: 行列 </td> </tr> </table>	<code>name</code>	適合変数名のリスト	<code>type</code>	適合変数のタイプのリスト : <ul style="list-style-type: none"> - <code>Distribution</code>: 軸ポイントディストリビューション - <code>OneDTable</code>: カーブ - <code>TwoDTable</code>: マップ - <code>Scalar</code>: スカラ値 - <code>Array</code>: 配列 - <code>Matrix</code>: 行列
<code>name</code>	適合変数名のリスト				
<code>type</code>	適合変数のタイプのリスト : <ul style="list-style-type: none"> - <code>Distribution</code>: 軸ポイントディストリビューション - <code>OneDTable</code>: カーブ - <code>TwoDTable</code>: マップ - <code>Scalar</code>: スカラ値 - <code>Array</code>: 配列 - <code>Matrix</code>: 行列 				
入力引数	<table border="0"> <tr> <td><code>pattern</code></td> <td> 適合変数を指定するための検索文字列です。'*' は0または1つ以上の任意の文字を表します。'#' は1つの任意の文字を表します。これらのワイルドカードで指定されるもの以外の文字は、すべて正確に一致している必要があります。検索文字列内の大文字と小文字は区別されません。 </td> </tr> <tr> <td><code>deviceName</code></td> <td>デバイス名</td> </tr> </table>	<code>pattern</code>	適合変数を指定するための検索文字列です。'*' は0または1つ以上の任意の文字を表します。'#' は1つの任意の文字を表します。これらのワイルドカードで指定されるもの以外の文字は、すべて正確に一致している必要があります。検索文字列内の大文字と小文字は区別されません。	<code>deviceName</code>	デバイス名
<code>pattern</code>	適合変数を指定するための検索文字列です。'*' は0または1つ以上の任意の文字を表します。'#' は1つの任意の文字を表します。これらのワイルドカードで指定されるもの以外の文字は、すべて正確に一致している必要があります。検索文字列内の大文字と小文字は区別されません。				
<code>deviceName</code>	デバイス名				
例	<pre>[n,t]=IncaBrowseCalibrationElements ('MAP*', 'Device'); [name,type]= IncaBrowseCalibrationElements ('*');</pre>				

4.5.2 適合変数の追加

コマンド名	IncaAddCalibrationElement	
機能	現在開いている実験に、適合変数を1つ追加します。適合変数として使用できるのは、スカラー値、および特性カーブ/マップ（軸ポイントディストリビューションを含む）です。1つの実験には任意の数の適合変数を割り当てることができます。軸ポイントディストリビューションとグループ軸ポイントディストリビューションに対しても、このコマンドを発行できます。	
構文	<pre>IncaAddCalibrationElement (deviceName, calibrationName {, displayMode})</pre>	
出力引数	なし	
入力引数	deviceName	デバイス名
	calibrationName	適合変数の名前
	displayMode	変数の表示モード
		<ul style="list-style-type: none"> — 2：適合変数を表示し、常に内容を更新します（デフォルト） — 1：表示しますが、更新は行いません — 0：非表示
		<p>高速転送時は、1に設定することによりスループットが向上します。</p>
例	<pre>IncaAddCalibrationElement ('anEtk', 'Scalar'); IncaAddCalibrationElement ('anEtk', 'Curve'); IncaAddCalibrationElement ('anEtk', 'Map');</pre>	

注記

グループ軸を使用するカーブとマップの補間は行われません。

4.5.3 適合値の取得

コマンド名	IncaGetCalibrationValue
機能	適合変数の現在の値、または適合変数が参照する軸ポイントディストリビューションの現在の値を返します。
構文	<pre>value = IncaGetCalibrationValue (deviceName, calibrationName {, start, size} {, valueType})</pre>

出力引数	value	適合変数の現在の値。変数のタイプに応じて以下の型のデータを返します。 <ul style="list-style-type: none">－ スカラ値の場合：(1,1)の行列－ カーブの場合：(x,1)の行列－ マップの場合：(x,y)の行列－ 軸ポイントディストリビューションの場合： (x,1)の行列
入力引数	deviceName	デバイス名
	calibrationName	適合変数の名前
	start	先頭のインデックスを指定します。データタイプに応じて以下のように指定します。 <ul style="list-style-type: none">－ カーブと軸ポイントディストリビューションの場合、先頭のインデックス x を指定します。 ($x \geq 1$)－ マップの場合、先頭のインデックス $[x, y]$ を指定します。 ($x, y \geq 1$)

size	読み込む値の数を指定します。データタイプに応じて以下のように指定します。 <ul style="list-style-type: none"> － カーブと軸ポイントディストリビューションの場合、値の数nを指定します。 $n \geq 1$ － マップの場合、値の数 $[n,m]$ を指定します。 $n,m \geq 1$
valueType	出力引数として受け取る適合変数の値の種類を文字列で指定します。この関数は、適合変数の値（デフォルト）または軸ポイントディストリビューションの値を返します。以下の設定が有効です。 <ul style="list-style-type: none"> － v : 出力値 － x : カーブ/マップのX軸ポイント値 － y : マップのY軸ポイント値

例

```

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 適合値の変更 (適合の実行)

コマンド名	IncaSetCalibrationValue
機能	適合変数、または適合変数が参照する軸ポイントディストリビューションに対して値を書き込みます。
構文	<pre>IncaSetCalibrationValue(deviceName, calibrationName, value) IncaSetCalibrationValue(deviceName, calibrationName, value, valueType) IncaSetCalibrationValue(deviceName, calibrationName, value, start) IncaSetCalibrationValue(deviceName, calibrationName, value, start, valueType) result = IncaSetCalibrationValue(deviceName, calibrationName, value) result = IncaSetCalibrationValue(deviceName, calibrationName, value, valueType) result = IncaSetCalibrationValue(deviceName, calibrationName, value, start) result = IncaSetCalibrationValue(deviceName, calibrationName, value, start, valueType)</pre>

出力引数	result	<p>適合結果が返ります（エラー時のみ）。いずれのビットも1でない場合、適合値が正しく変更されたことを示します。ビット5～8のいずれかが1の場合、適合値が条件付きで変更されたことを示します。ビット0～4のいずれかが1の場合、適合値が変更されていません。</p> <ul style="list-style-type: none"> － ビット0：適合が実行されませんでした － ビット1：値が下限値（ソフトリミット）を超えています。 － ビット2：値が上限値（ソフトリミット）を超えています。 － ビット3：値が下限値（ハードリミット）を超えています。 － ビット4：値が下限値（ハードリミット）を超えています。 － ビット5：値が下限値（ソフトリミット）に設定されました。 － ビット6：値が下限値（ソフトリミット）に設定されました。 － ビット7：値が上限値（ソフトリミット）に設定されました。 － ビット8：値が下限値（ハードリミット）に設定されました。 <p>適合が実行されない理由はいくつかあります。たとえば、要求された値が、現在設定されている適合モードに応じた上下限値を超えている場合があります。その場合は、ビット1～4によってその詳細が示されます。そのほか、適合変数やアクティブページが書き込み禁止である場合や、X/Y軸のポイント値のモニターが守られていない（値が順に増加していない）場合などがあります。これらの場合はすべて、ビット0のみが1にセットされません。</p>
入力引数	deviceName calibrationName	デバイス名 適合変数の名前

value	<p>適合変数の値を指定します。変数のタイプに応じて以下の型のデータを使用します。</p> <ul style="list-style-type: none"> — スカラ値の場合：(1,1) の行列 — カーブの場合：(x,1) の行列 — マップの場合：(x,y) の行列 — 軸ポイントディストリビューションの場合：(x,1) の行列
start	<p>インデックスの初期値。データタイプに応じて以下のように指定します。</p> <ul style="list-style-type: none"> — カーブと軸ポイントディストリビューションの場合、先頭のインデックス x を指定します。 $(x \geq 1)$ マップの場合、先頭のインデックス $[x, y]$ を指定します。 $x, y \geq 1$
valueType	<p>適合する値の種類を文字列で指定します。この関数は、適合変数の値（デフォルト）または軸ポイントディストリビューションの値を変更します。以下の設定が有効です。</p> <ul style="list-style-type: none"> — v : 出力値（デフォルト） — x : カーブ/マップの軸ポイントのX座標値 — y : マップのY軸ポイント値

例

```

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 データセットをデバイスに割り当てる（拡張コマンド）

コマンド名	IncaSetDatasetInDevice	
機能	現在開いている実験内のデバイスにデータセットを割り当てます。	
構文	IncaSetDatasetInDevice(device, dataset)	
出力引数	なし	
入力引数	device	デバイス名
	dataset	データセットのデータベースパス
例	IncaSetDatasetInDevice ('ETK:1', 'Ds4711\Ds4711_3')	

4.5.6 デバイスのデータセットリストを取得（拡張コマンド）

コマンド名	IncaGetDatasetsForDevice	
機能	指定のデバイスに割り当てられているデータセットのリストを取得します。	
構文	name = IncaGetDatasetsForDevice(device) [name, properties] = IncaGetDatasetsForDevice(device)	
出力引数	name	検索されたすべてのデータセット名（フルパス）のリスト
	properties	データセットの属性を表す文字列リストです。値は以下のいずれかです。 <ul style="list-style-type: none"> — " (empty string): 読み書き可能なマスターデータセット — r: 読み込みのみ可能なデータセット — m: 読み書き可能なマスターデータセット — mr: 読み込みのみ可能なマスターデータセット
入力引数	device	デバイス名
例	l = IncaGetDatasetsForDevice('ETK:1')	

4.5.7 適合モードの設定（拡張コマンド）

コマンド名	IncaSetCalibrationMode	
機能	このコマンドの実行後にIncaSetCalibrationValueコマンドで実行されるすべての適合処理に適用される共通の適合モード（上下限値を超える値が指定された際の処理モード）を設定します。設定されたモードは、実験を一度閉じて再度開いた後も継続して有効となります。ただしMATLAB [®] インターフェースの起動時には、上限値と下限値の両方にデフォルトモードであるrejectWeakBoundViolationが設定されます。	
構文	IncaSetCalibrationMode(lowerLimitMode, upperLimitMode)	
出力引数	なし	
入力引数	lowerLimitMode	下限値よりも小さい値についての処理
	upperLimitMode	上限値よりも大きな値についての処理
		<ul style="list-style-type: none"> — rejectWeakBoundViolation: ソフトリミット値の範囲を超える値を設定しようとする、それ以降の適合コマンドがすべて拒否されます。（デフォルト） — limitToWeakBound: ソフトリミット値の範囲を越える値を設定しようとする、設定しようとした値の代わりにソフトリミット値（上限値または下限値）が設定されます。 — rejectHardBoundViolation: ハードリミット値の範囲を超える値を設定しようとする、それ以降の適合コマンドがすべて拒否されます。 — limitToHardBound: ハードリミット値の範囲を超える値を設定しようとする、設定しようとした値の代わりにハードリミット値（上限値または下限値）が設定されます。
例	<pre>IncaSetCalibrationMode ('rejectHardBoundViolation', 'limitToHardBound')</pre>	

4.5.8 グループデバイス（拡張コマンド）

コマンド名	IncaGroupDevices
機能	デバイスのグルーピングをアクティブまたは非アクティブにします。
構文	IncaGroupDevices(onOff)

出力引数	なし	
入力引数	onOff	<ul style="list-style-type: none"> — 0: デバイスのグルーピングをオフ（無効）にします。 — 1: デバイスのグルーピングをオン（有効）にします。
例	IncaGroupDevices (1)	

4.5.9 DCMファイルの書き込み（拡張コマンド）

コマンド名	IncaWriteToFile	
機能	現在開いている実験のデータをDCMファイルに書き込みます。	
構文	IncaWriteToFile(format, file, device, calibs {, options})	
出力引数	なし	
入力引数	format	ファイルフォーマットの識別子： — 'DCM': DCM フォーマット
	file	ファイル名（フルパス）
	device	書き込む適合変数が属するデバイスの名前
	calibs	書き込む適合変数のリスト（セル配列）
	options	指定のファイルフォーマットへの書き込み時のオプション
例	<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 メモリページマネージャ

ここまでに説明されているすべてのAPI関数は、デバイス上で現在アクティブになっているページについてのみ機能します。基本的に、適合アクセスはワーキングページについてのみ可能です。ただし、INCAデータベースとETKとの間でワーキングページのチェックサムが一致しない場合、ETKのワーキングページはロックされ、書き込み禁止となります。

以下に、メモリページ管理に関する機能を持つAPI関数を紹介します。

4.6.1 メモリページの切り替え

コマンド名	IncaSwitchPage
機能	指定のメモリページをアクティブにします。
構文	IncaSwitchPage(deviceName, pageName)

出力引数	なし	
入力引数	deviceName	デバイス名
	pageName	ページ名
		－ wp : ワーキングページ
		－ rp : リファレンスページ
例	<pre>IncaSwitchPage('MyDevice', 'wp'); IncaSwitchPage('Dev', 'rp');</pre>	

4.6.2 カレントページの取得（拡張コマンド）

コマンド名	IncaGetCurrentPage	
機能	現在アクティブになっているページを取得します。	
構文	pageName = IncaGetCurrentPage(deviceName)	
出力引数	pageName	アクティブなメモリページの名前： － wp : ワーキングページ － rp : リファレンスページ
入力引数	deviceName	デバイス名
例	<pre>p = IncaGetCurrentPage('MyDevice');</pre>	

4.6.3 書き込み保護チェック

コマンド名	IncaIsPageWriteProtected	
機能	指定のメモリページが書き込み禁止になっているかどうかを調べます。	
構文	isRW = IncaIsPageWriteProtected(deviceName, pageName)	
出力引数	isRw	－ 0 : 書き込み可能 － ≠0 : 書き込み禁止
入力引数	deviceName	デバイス名
	pageName	ページ名
		－ wp : ワーキングページ
		－ rp : リファレンスページ
例	<pre>isETK1RW = IncaIsPageWriteProtected('ETK:1', 'wp'); isETK2RW = IncaIsPageWriteProtected('ETK:2', 'rp');</pre>	

4.6.4 メモリページのダウンロード

コマンド名	IncaDownloadPage	
機能	指定のメモリページを制御デバイスにダウンロードします。	
構文	IncaDownloadPage (deviceName, pageName)	
出力引数	なし	
入力引数	deviceName	デバイス名
	pageName	ダウンロードするページ名 <ul style="list-style-type: none"> - wp : ワーキングページ - rp : リファレンスページ
例	IncaDownloadPage ('ETK:1', 'wp'); IncaDownloadPage ('ETK:1', 'rp');	

4.6.5 メモリページのコピー

コマンド名	IncaCopyPageFromTo	
機能	指定のメモリページをコピーします。現時点では、リファレンスページからワーキングページへのコピーしかサポートされておらず、ソースとターゲットの他の組み合わせはサポートされていません。	
構文	IncaCopyPageFromTo (deviceName, sourcePageName, destinationPageName)	
出力引数	なし	
入力引数	deviceName	デバイス名
	sourcePageName	コピー元のページ : <ul style="list-style-type: none"> - wp : ワーキングページ - rp : リファレンスページ
	destinationPageName	コピー先のページ : <ul style="list-style-type: none"> - wp : ワーキングページ - rp : リファレンスページ
例	IncaCopyPageFromTo ('ETK:1', 'rp', 'wp');	

4.6.6 不一致部分のダウンロード

コマンド名	IncaDownloadDifferences
機能	ワーキングページとリファレンスページの差分をECUにダウンロードします。この処理は、ECUのワーキングページとリファレンスページの内容がINCAのリファレンスページの内容と一致している場合にのみ実行されます。
構文	IncaDownloadDifferences (deviceName)

出力引数	なし
入力引数	deviceName デバイス名
例	IncaDownloadDifferences('ETK:1');

4.6.7 メモリページのアップロード（拡張コマンド）

コマンド名	IncaUploadPages
機能	リファレンスページとワーキングページをアップロードして、新しいデータセットを作成します。作成されたデータセットは自動的にそのデバイスに割り当てられます。
構文	IncaUploadPages(device{, referencePage, workingPage})
出力引数	なし
入力引数	device デバイス名
	referencePage リファレンスページ用のデータセット名。 指定がない場合はデフォルト名が使用されます。
	workingPage ワーキングページ用のデータセット名。 指定がない場合はデフォルト名が使用されます。
例	IncaUploadPages('ETK:1'); IncaUploadPages('ETK:1', 'ref_1', 'work_1');

4.7 応用例

例1

```
% Check if working page is write-protected and
% download the page if it is write-protected
if(IncaIsPageWriteProtected('anEtk', 'wp'))
IncaDownloadPage('anEtk', 'wp');
end

% Switch to the working page
IncaSwitchPage('anEtk', 'wp');
```

例2

以下に、各API関数を利用して、デバイス MyDevice の測定ラスタ 10ms から測定値を読み取る例を示します。この例に示された処理を行うには、前もってINCA上で、MyDevice という名前のデバイスが割り当てられた実験を開いておく必要があります。

```
% Measure the following signals(測定変数の割り当て)
IncaAddMeasureElement('MyDevice', '10ms', 'Chan1');
IncaAddMeasureElement('MyDevice', '10ms', 'Chan2');
```

```

IncaAddMeasureElement( 'MyDevice', '10ms', 'Chan3' );
IncaAddMeasureElement( 'MyDevice', '10ms', 'Chan4' );
% Now measure (測定開始)
data = [];
time = [];
IncaShowMessages(0);
IncaSetMeasureReadMode(0)
IncaStartMeasurement;
deltaT = 0;
% Measure for 20 seconds (合計で20秒間測定を行う)
while( deltaT < 20 )
    % Pause for 0.1 seconds to have more than one
    % record -- saves processor time. (0.1秒間ポーズ)
    pause(0.1)
    % Get up to 500 records for group 10ms (10msラスタの
データを最大500レコード取得)
    [ t, d ]=IncaGetRecords( 'MyDevice', '10ms', 500
);
    % Append t and d to time and data (データを保存)
    data = [data; d];
    time = [time; t];
    if( length(time) )
        % Calculate time measured (測定時間の計算)
        deltaT = time( length(time)) - time(1);
    end
end
IncaStopMeasurement;
IncaShowMessages(1);
% Plot the results (データの視覚化)
plot(time, data);

```

上記の例では、1つのラスタのみを使用して測定を行っています。実際には、複数のラスタに測定変数を割り当て、MATLAB[®]から独立して各ラスタからデータを取得することもできます。

5 コンパイラを用いたスタンドアロン実行ファイルの作成と配布

INCA-MIPを使用すると、MATLAB[®]の機能が実装されたINCA-MIP API関数を含むMファイルを作成してコンパイルすることができます。作成されたスタンドアロンファイルは、MATLAB[®]がインストールされていない環境でも実行可能です。

スタンドアロン実行ファイルを作成するには、PCにMATLAB[®]がインストールされている必要があります。作成された実行可能ファイルにはMATLAB[®]とETASのDLLのコピーが含まれるため、ターゲットPCにMATLAB[®]をインストールせずに実行することができます。

ただし、コンパイル済み実行ファイルをターゲットPCで実行する際には、MATLAB[®]ランタイムライブラリが必要になる場合があります。

MATLAB[®]ランタイムライブラリをインストールして使用方法については、MATLAB[®]ユーザードキュメントの「Distributing Stand-Alone Applications」の項を参照してください。

5.1 Mファイルのコンパイル

MATLAB[®]コンパイラを使用してMファイルをコンパイルするには：

1. Inca*.dllファイルとInca*.mexw64ファイルをすべて現在の作業ディレクトリにコピーします。

".dll"ファイルと".mexw64"ファイルが格納されているフォルダは、インストール時に選択したインストールタイプにより異なります。

例：

–MATLAB[®]にインストールした場合：

```
<drive:\>Program Files\MATLAB\Ryyyy\bin\win64
```

–ETASDataディレクトリにインストールした場合：

```
<drive:\>ETASData\INCAx.y\INCA-MIPx64
```

MATLAB[®]コンパイラとINCA-MIPのパスがすべてWindowsの環境変数に登録されていることを確認してください。

環境変数の設定方法は、Windowsの "システムの詳細設定" に関するユーザードキュメントを参照してください。

以下のコマンドを実行します。

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

例：

以下のコマンドを実行すると、testCase2.mというファイルからスタンドアロン実行ファイルが生成されます。

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

この結果、testCase2.exeというファイルが生成されます。

MATLAB[®]コンパイラは、コンパイルされたMATLAB[®]スクリプトを実行するために必要なすべてのMEX関数DLLとそれに関連するDLLを含むコンテナを作成します。このコンテナには、スクリプトで使用されるすべてのInca*.dllファイルとincaRci2x64Matlab.dllが含まれている必要があります。INCA MEX関数DLLの拡張子は、*.mexw64です。

コンパイルされたスクリプトの実行時には、PC上にDLLファイルが存在している必要はありません。

使用できるMATLAB[®]コンパイラのバージョンや設定などについての詳しい情報は、MATLAB[®]ユーザードキュメントを「MATLAB[®] Compiler」または「`mcc`」というキーワードで検索してください。

i 注記

INCAは、一度に1つのMATLAB[®]セッションからしか操作できません。INCAやスタンドアロン実行ファイルの複数インスタンスから同時にMATLAB[®]を操作しようすると、その処理は中断され、エラーメッセージが発行されます。

i 注記

MATLAB[®] R2016a以降では、INCA MEX関数DLLの拡張子は*.mexw64です。

5.2 スタンドアロン実行ファイルの配布

MATLAB[®]コンパイラでコンパイルされたスタンドアロン実行ファイルを実行するには、その実行ファイル以外のファイルは必要ありません。MATLAB[®]がインストールされている必要はなく、MATLAB[®]ライブラリのコピーも必要ありません。

MATLAB[®]コンパイラでコンパイルされたスタンドアロン実行ファイルを配布するには：

- スタンドアロン実行ファイルをターゲットPCにコピーします。

これだけで実行ファイルを実行できるようになるので、特別な操作は必要ありません。

6 お問い合わせ先

テクニカルサポート

各国支社の営業やテクニカルサポートについての情報は、ETASウェブサイトをご覧ください。

www.etas.com/ja/hotlines.php



ETASでは、お客様向けに製品トレーニングを提供しています。

www.etas.com/academy

ETAS本社

ETAS GmbH

Borsigstraße 24	電話 :	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	インターネット :	www.etas.com

索引

E

ETAS

お問い合わせ先 64

I

IncaAddCalibrationElement 49

IncaAddMeasureElement 32

IncaBrowseCalibrationElements 48

IncaBrowseItemsInFolder 26

IncaBrowseMeasureElements 31

IncaClose 25

IncaCopyPageFromTo 59

IncaDatabaseImport 25

IncaDownloadDifferences 59

IncaDownloadPage 59

IncaExecuteManualTrigger 47

IncaGetCalibrationValue 49

IncaGetCurrentPage 58

IncaGetDatasetsForDevice 55

IncaGetDeviceProperties 29

IncaGetDevices 29

IncaGetHardwareStatus	43
IncaGetInstalledAddOnInfo	22
IncaGetInstalledProductInfo	22
IncaGetMeasureRatesForDevice	31
IncaGetProperties	23
IncaGetRecordingMode	37
IncaGetRecordingProperties	34
IncaGetRecordingState	47
IncaGetRecords	41
IncaGetRecordStruct	47
IncaGetVersion	23
IncaGroupDevices	56
IncaIsLicenseValid	21
IncaIsPageWriteProtected	58
IncaOpen	24
IncaOpenDatabase	25
IncaOpenExperiment	28
IncaResetExperiment	28
IncaResetRecords	43
IncaSetCalibrationMode	56
IncaSetCalibrationValue	52
IncaSetDatasetInDevice	55
IncaSetMeasureReadMode	40

IncaSetProjectAndDatasetInDevice	27
IncaSetRecordingMode	38
IncaSetRecordingProperties	36
IncaSetTrigger	44
IncaShowMessages	21
IncaStartMeasurement	33
IncaStartRecording	39
IncaStopMeasurement	34
IncaStopRecording	39
IncaSwitchPage	57
IncaUploadPages	60
IncaWriteToFile	57
M	
mcc	62
MEXファイル	11
お	
お問い合わせ先	64
さ	
サンプルファイル	18
し	
シグナル	10

シグナルグループ	10
す	
スタンドアロン実行ファイル	62
て	
データ	7
データレコード	9
デバイス	9
め	
メモリページ管理	8
り	
リングバッファ	9
漢字	
測定	8
測定データ	9
測定ラスタ	9
適合	8
適合変数	9