

# ETAS INCA-SIP 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-SIP V7.5 | ユーザーガイド R01 JP - 03.2024

## 目次

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 .....	8
2	INCA-SIPの概要 .....	9
2.1	Simulink®モデルをINCAに接続する .....	9
2.2	モデルの測定と適合 .....	10
2.3	技術的概要 .....	10
	2.3.1 基本的な処理 .....	10
	2.3.2 INCA-SIPのMATLAB®コンポーネント .....	10
	2.3.3 MATLAB®との接続 .....	11
	2.3.4 INCAとの接続 .....	11
	2.3.5 測定変数の扱い .....	11
	2.3.6 Simulink®とINCAの同期 .....	12
	2.3.7 適合変数の扱い .....	12
3	製品のインストール .....	13
3.1	システム要件 .....	13
	3.1.1 ハードウェア要件 .....	13
	3.1.2 ソフトウェア要件 .....	13
3.2	インストール .....	13
	3.2.1 INCA-SIP V7.5のインストール .....	13
	3.2.2 C++コンパイラの登録 .....	13
	3.2.3 スタートメニュー .....	14
	3.2.4 INCA-SIPのインストールパス .....	14
	3.2.5 MATLAB®スクリプトファイル .....	14
3.3	ソフトウェアのライセンス管理 .....	14
4	INCA-SIPの使用方法 .....	16
4.1	MATLAB®の検索パスの変更 .....	16
4.2	INCAバージョンの切り替え .....	16
4.3	メニューアイテム .....	16
	4.3.1 Connect to INCA .....	16

4.3.2	ECUs	18
4.3.3	Configuration (設定)	22
4.3.4	MDF Configuration	23
4.4	マクロ	26
4.5	命名規則	27
4.5.1	適合変数と測定変数	27
4.5.2	グループ	27
4.6	INCA-SIP のエミュレーションモード	27
4.6.1	リアルタイムエミュレーションモード	27
4.6.2	高速エミュレーションモード	28
4.7	INCA-SIPとブロック削減	28
4.8	シミュレーションモード	28
4.9	参照モデル (Model Referenceブロック) のサポート	29
4.10	適合操作の拒否	29
4.11	ECUのビジー状態	29
4.12	実験の開始と停止	29
4.13	Simulink.SignalとAsap2.Signalのサポート	30
4.14	Simulink.ParameterとAsap2.Parameterのサポート	30
4.15	各種メッセージ (エラー、警告、情報)	30
4.16	DLLモード	30
4.16.1	DLLの作成	31
4.16.2	DLL検索スクリプトの作成	32
4.16.3	DllAsap2ObjectFactoryを用いたオブジェクトの生成	34
4.17	データセットをMDFファイルに書き込む	39
4.18	INCAのリモート操作	41
4.19	カスタムフック	42
4.19.1	setVariableFilterFunctionHandle	43
4.19.2	setVariableCustomizationFunctionHandle	45
4.19.3	setVariableRenamingFunctionHandle	47
4.19.4	setGroupRenamingFunctionHandle	48
4.19.5	MScripts.VariableWrapper	50
4.19.6	MScripts.AccessInfo	55
4.20	サポートされているSimulink®ブロックのタイプ	56
4.21	制限事項とその他の挙動	58
4.21.1	RAM適合	58
4.21.2	他のハードウェアとの同時測定	58

4.21.3	測定の一時的停止 .....	58
5	チュートリアル .....	59
6	お問い合わせ先 .....	61
目次	.....	62

# 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-SIP (INCA Simulink<sup>®</sup>統合パッケージ) は、MATLAB<sup>®</sup>/Simulink<sup>®</sup>でモデリングされたECUファンクションに対してINCAからの測定・適合アクセスを可能にするINCAアドオンです。INCA-SIP アドオンは、Simulink<sup>®</sup>ツールボックスに追加されるものです。

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。その中には、適合操作やINCAやMDAなどのソフトウェア、さらには適合対象システムのソフトウェアアルゴリズムなどが含まれます。

## 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。この情報を参照するには、INCAのヘルプメニューヘルプ > **安全上のご注意** を選択してください。

## 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-SIPの概要

INCA-SIP (INCA Simulink<sup>®</sup>統合パッケージ) は、MATLAB<sup>®</sup>/Simulink<sup>®</sup>でモデリングされたECUファンクションに対してINCAからの測定・適合アクセスを可能にするINCAアドオンです。INCA-SIP アドオンは、Simulink<sup>®</sup>ツールボックスに追加されるものです。

「製品のインストール」(ページ13) の記述に従ってインストールを行うと、Simulink<sup>®</sup>上に新しいメニューアイテムが表示され、使用できるようになります。このメニューアイテムにより、INCAとSimulink<sup>®</sup>モデルとの間の通信が確立されます。

接続処理の一環としてINCA-SIPは、ユーザーモデルに基づきINCAのオブジェクト (.a2l、.s19、INCAワークスペース) を生成します。さらにINCA-SIPは、標準のSimulink<sup>®</sup>ブロックを用いてモデルの測定と適合を行えるように、一連の検索スクリプトを提供します。ユーザー固有のブロックについても測定・適合を行うことができますが、ユーザー固有のブロックを扱うには、カスタマイズされた検索スクリプトを作成する必要があります。

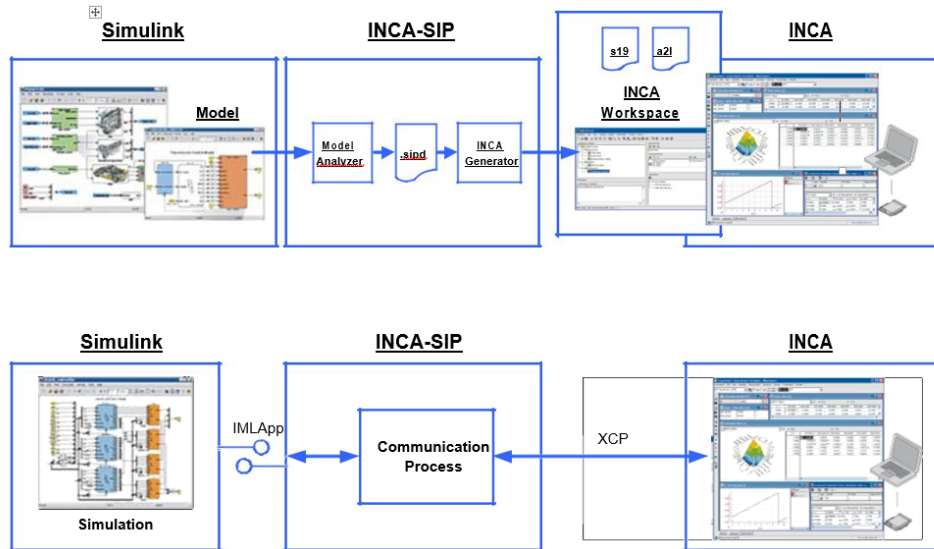
### 2.1 Simulink<sup>®</sup>モデルをINCAに接続する

モデルの接続処理において、INCA-SIPはモデルブロックの分析を行います。「モデルの測定と適合」(次ページ) 内の上側の図を参照してください。カスタマイズされた検索スクリプトによって分析処理をカスタマイズすることにより、ユーザー固有のブロックを扱うこともできます。

接続に成功すると、測定と適合アクセスに必要なすべてのパラメータが決定され、INCAに必要なすべてのアイテムが生成されます。

## 2.2 モデルの測定と適合

INCAの環境において、INCA-SIPアドオンは1つのXCPデバイスとして扱われます。INCAは、XCPを使用してINCA-SIPとのデータ送受信を行い、INCA-SIPはMATLAB®のCOM APIインターフェースを介してモデルとのデータ交換を行います（下側の図を参照してください）。INCAがSimulink®モデルから測定値を受信できるのは、Simulink®モデルの実行中に限られますが、適合は、モデルが実行されていない（つまり停止中や一時停止中である）ときにも行えます。



INCA-SIPによるモデルの分析（上の図）とM/Cアクセス（下の図）

## 2.3 技術的概要

### 2.3.1 基本的な処理

INCA-SIPは、INCAでのSimulink®モデルの計測・適合を可能にするためのもので、アプリケーション、ライブラリ、MATLAB®スクリプトで構成されています。INCA-SIPの機能のほとんどは、MATLAB®スクリプトで実装されています。この構成により、ユーザー固有の適合・測定ブロックを使用するなど、幅広いカスタマイズが可能になっています。MCアプリケーションに対してINCA-SIPは、Simulink®モデルに応じた数の「仮想ECU」を表す1つのASAM MCD-1 XCP V1.1.0サーバーとなります。INCAなどのアプリケーションは、これらの「ECU」が実際の物理デバイスであるかのようにやり取りを行います。その状況においてINCA-SIPは、Microsoft COM APIを使用してINCA-SIPサーバーの実行ファイルとSimulink®モデルとの間の通信を行い、Simulink®モデルとの計測・適合データの交換を行います。

### 2.3.2 INCA-SIPのMATLAB®コンポーネント

INCA-SIPを使用するには、INCA-SIPのインストールディレクトリへのパスをMATLAB®の検索パスリストに追加する必要があります。インストールディレクトリには、Simulink®に対してINCA-SIPのメニューアイテムを **Tools** メニューに追加することを要求する `sl_customization.m` というスクリプトが含まれています。MATLAB®固有の機能のほとんどは、**ETAS SIP**

Simulinkブロックに実装されており、このブロックを適合対象のモデルに追加する必要があります。モデルにこのブロックを追加すると、カスタムSimulinkブロックを含むINCA-SIPモデルがロードされ、ブロックがロードされます。

### 2.3.3 MATLAB<sup>®</sup>との接続

INCA-SIPでは、接続処理は常にSimulink<sup>®</sup>側から開始されます。接続処理は、ユーザーがINCA-SIPのサブメニューから **Connect to INCA** を選択した時点で開始されます。接続処理中は、MATLAB<sup>®</sup>スクリプトが `find_system()` を実行してモデル内の全ブロックを取得し、適合に使用できるブロック（定数、ルックアップテーブルブロックなど）と、測定変数として使用できる名前付き信号を見つけ出します。この解析の結果は3つのファイルに保存されます。

ファイル	説明
*.sipd	SIPDはINCA-SIP固有のXMLファイルで、解析中に見つかったすべてのアイテム（測定変数、適合変数、イベント、グループ、変換メソッド）に関する記述が含まれています。
*.a2l	A2Lは、基本的に同じ情報を含んでいますが、ASAM MCD2 MC (ASAP2) に準拠したMCアプリケーションのほとんどが認識できるフォーマットになっています。INCAなどのアプリケーションは、このファイルを使用して測定変数や適合変数を作成し、変数選択ダイアログボックスに表示します。
*.s19	S19はモトローラS19フォーマットのバイナリファイルで、測定変数と適合変数のメモリオフセットを含みます。

接続が確立されると、その後のやりとり（適合値の送信、可視化用の測定データの要求など）は、INCAなどのMCアプリケーションによって開始されます。接続が確立された後は、測定が開始されるまで、モデルは一時停止状態で起動します。

### 2.3.4 INCAとの接続

A2LファイルとS19ファイルが生成されると、MATLAB<sup>®</sup>スクリプトがINCA-SIP COMサーバーを起動します。サーバーの起動後、スクリプトはINCAを起動し、INCAに必要なすべてのアイテム（フォルダ、ワークスペース、プロジェクト）を作成し、一連のINCA COM APIを実行してINCA実験環境を可視化する準備を行います。A2Lファイルの情報をを用いて、INCA用の「A2Lプロジェクト」が作成され、モデル内の各仮想ECUがINCAのハードウェアコンフィギュレーションエディタにおいて「Ethernet System Simulationデバイス」として表現されます。これらの処理が完了すると、ユーザーがその後の操作（MC変数の選択、実験レイアウトの設定、可視化の開始）を行う準備が整います。そして最後に、接続スクリプトはINCA-SIPサーバー上でCOM APIコールを行い、そのXCPリスナスレッドを開始します。この時点で接続処理が完了し、INCA-SIP XCPサーバー（スレーブ）がINCAからのコマンドを受け付けられる状態になります。

### 2.3.5 測定変数の扱い

ECUを設定する際にECUダイアログボックスで選択するラスタは、INCAがモデルの測定変数の更新を受け取る頻度に反映されます。カスタムラスタがひとつも選択されていない場合は、デフォルトラスタ（100ms）が設定されます。

これらのラスタ周期はSimulink<sup>®</sup>の内蔵ソルバステップより小さくすることはできず、このソルバステップの倍数でなければなりません。ラスタ周期が短いと、INCAとSimulink<sup>®</sup>間の通信量が増え、モデルの実行が非効率的になります。

モデル実行中、Simulink<sup>®</sup>においてこのラスタがトリガされるたびに、COMイベントがINCA-SIP実行ファイルに送られ、INCA-SIPは、INCAで選択された全変数について、その時点（シミュレーション時刻）における値を取得してキャッシュするようトリガされます。同じ変数に対して同時に複数のラスタがトリガされた場合は、ローカルにキャッシュされた値が使用されます。

変数がSimulink<sup>®</sup> ECUのものである場合は、INCA-SIP実行ファイルはそれぞれの値ごとにCOMを介してSimulink<sup>®</sup>モデルと通信し、最新の値を取得します。DLL ECUの場合は、INCA-SIP実行ファイルはメモリにロードされたDLLを読み取ります。

SIPサーバーは、INCAターゲットサーバーからのXCP呼び出しへの応答として、この測定データを取得してINCAに送ります。このデータはINCAにおいて、物理ECUから受け取ったものであるかのように扱われます。

### 2.3.6 Simulink<sup>®</sup>とINCAの同期

Simulink<sup>®</sup>からのこの「COMイベントアプローチ」の主な目的は、測定データの送信レートを希望のラスタ周期に合わせて抑制することです。Simulink<sup>®</sup>モデルは、デフォルトのラスタ周期である100msよりも桁違いに早いソルバステップで更新され、これはINCAがユーザーインターフェースを更新できる速度よりもはるかに速い速度になります。

INCA-SIP XCPサーバーは、COMイベントを通じてこのような要求を受け取り、特定の時点における同期が確保できるように、シミュレーション時間を進める前にすべての変数を更新する必要があります。

これは、Simulink<sup>®</sup>モデルの信号をオシロスコープに接続することで確認できます。ラスタにより決定される測定レートに合わせてグラフの更新が遅くなります。

### 2.3.7 適合変数の扱い

適合値の変更は、XCPサーバーからの要求としてINCA-SIPに送信されます。これらの変更は、INCAで可視化が行われているときにのみ、モデル内の対応するブロックパラメータに適用されません。要求に応えるためINCA-SIPサーバーは、接続が確立されたときに取得したハンドルを持つ実行中のモデルのSimulink<sup>®</sup>ウィンドウにメッセージをポストします。その後、Simulink<sup>®</sup>はMスクリプトを実行し、メッセージで指定されたブロックパラメータの値をセットします。

適合は、モデルがシミュレーション中でないとき、つまり、停止または一時停止しているときでも行えます。変更した内容は、INCAでの測定が再開された時点でモデルに反映されます。

## 3 製品のインストール

### 3.1 システム要件

INCA-SIPを使用するには、以下の要件が満たされている必要があります。

#### 3.1.1 ハードウェア要件

アドオンのハードウェア要件は、INCAとMATLAB<sup>®</sup>/Simulink<sup>®</sup>のものと同じです。各製品のマニュアルを参照してください。

#### 3.1.2 ソフトウェア要件

INCA-SIP V7.5を使用するには、以下のソフトウェアコンポーネントが必要です。

- － INCA V7.5とその最新のホットフィックス
- － MATLAB<sup>®</sup>/Simulink<sup>®</sup>（対応バージョンはリリースノートを参照してください）

### 3.2 インストール

本項には、インストールの方法と、スタートメニューのエントリに関する補足情報、およびMATLAB<sup>®</sup>パスの変更について記述されています。

#### 3.2.1 INCA-SIP V7.5のインストール

本項では、INCA-SIP V7.5をインストールする方法を説明します。

##### **INCA-SIPをインストールする**

1. ETASのプログラムが開いている場合は、すべて閉じます。
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ず。  
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というファイルを実行します。  
または  
ネットワークドライブからの場合は、Setup\_ServicePack.exeというファイルを実行します。
3. インストールするコンポーネント（INCA V7.5、MDA、INCA-SIP、さらに必要に応じてその他のアドオン）について、「インストール」列のチェックボックスをオンにします。

#### 3.2.2 C++コンパイラの登録

INCA-SIPとともにSimulink<sup>®</sup>のラピッドアクセラレータモードを使用するには、MEXコンパイラ用にC++コンパイラが登録されている必要があります。ラピッドアクセラレータモードでINCA-SIPを使用する前に、MSVC（マイクロソフトVisual Studio）などの外部コンパイラをMEXコンパイラとして登録してください。外部コンパイラを登録するには、INCA-SIPをINCAに接続する前に、任意のバージョンのMATLAB<sup>®</sup>>において"mex -setup"を実行します。サポートされているコンパイラの一覧は、MATLAB<sup>®</sup>のドキュメントを参照してください。

### 3.2.3 スタートメニュー

インストールが正常に終了すると、Windowsのスタートメニューに次のエントリが追加されます。

- **E > ETAS INCA V7.5 > INCA V7.5 Manuals and Tutorials**  
ユーザーマニュアルとその他のサポートドキュメントが保存されたフォルダが開きます。
- **E > ETAS > ETAS License Manager**  
ETASソフトウェア製品のライセンス管理（ライセンスの登録、サーバーライセンスの借用／返却など）を行うETASライセンスマネージャが開きます。

### 3.2.4 INCA-SIPのインストールパス

INCA-SIPのデフォルトのインストールパスは以下のとおりです。

64ビット            <drive>:\Program Files\ETAS\AddOn\_SIP7.5\

### 3.2.5 MATLAB<sup>®</sup>スクリプトファイル

INCA-SIPをインストールすると、MATLAB<sup>®</sup>スクリプトファイルが以下のディレクトリに保存されます。

64ビット            <drive>:\Program Files\ETAS\AddOn\_  
SIP7.5\+MScripts

"+MScripts"フォルダに含まれるMATLAB<sup>®</sup>スクリプトファイルには、INCA-SIPサーバーによって呼び出されるものと、Simulink<sup>®</sup>がINCA-SIPサーバーやINCAとやりとりする際に使用するものがあります。

## 3.3 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](http://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 INCA-SIPの使用法

本章では、INCA-SIPを使用するための準備作業や操作方法について説明します。

### 4.1 MATLAB<sup>®</sup>の検索パスの変更

INCA-SIPを特定バージョンのMATLAB<sup>®</sup>で使用するには、INCA-SIPのインストールパスを検索パスに追加する必要があります。そのためには、MATLAB<sup>®</sup>コマンド`addpath`と`savepath`を続けて実行します。パスについての情報は、[3.2.4「INCA-SIPのインストールパス」](#)（ページ14）を参照してください。

パスが入力されると、Simulink<sup>®</sup>ウィンドウの"Tools"メニューにINCA-SIPというメニューアイテムが表示されます。

### 4.2 INCAバージョンの切り替え

同じPCにINCA-SIP V7.5とその他のINCA-SIP V7.xをインストールすることができますが、複数のバージョンを同時に使用することはできません。

INCAのバージョンを切り替える際には、それに合わせて手動操作でINCA-SIPのパスを変更し、MATLAB<sup>®</sup>を再起動する必要があります。

詳細は、「[MATLAB<sup>®</sup>の検索パスの変更](#)」（上記）を参照してください。

### 4.3 メニューアイテム

INCA-SIPメニューアイテムには以下の機能が含まれます。

- － 「Connect to INCA」（下記）
- － 「ECUs」（ページ18）
- － 「Configuration（設定）」（ページ22）
- － 「MDF Configuration」（ページ23）
- － Help

#### 4.3.1 Connect to INCA

この機能を選択すると、コンフィギュレーション設定に従ってINCA-SIPが所定のモデルの処理を開始します。INCA-SIP内で有効になっている「ECU定義」ごとに、適合変数と測定変数が識別され、処理されます。各ECU定義はA2LファイルとS19ファイルに変換され、変換が完了すると、INCAが自動的に開き、INCAのワークスペースが自動生成されます。さらに各ECUに対応するデバイスがINCA上に作成され、プロジェクトとデータセットも作成されます。

INCA-SIPは、ECUごとにEPKを生成します。

EPKは以下の情報で構成されます。

- － ECUに存在する適合変数と測定変数のためのハッシュ関数
- － ECUへのアクセスに使用するネットワークインターフェースのIPアドレス
- － 使用するTCPポート

これにより、INCA上のデバイスをINCA-SIP内の適切なSimulink<sup>®</sup>モデルに接続することができます。



モデルが変更された後にINCAへの再接続を行うと、INCA-SIPは変更があったことを検出し、自動的にINCAのデータセットを置き換えます。その際、すでにINCAの実験を開いていた場合は、INCA-SIPは実験を閉じる前にその内容を保存するようユーザーに促します。

接続処理が完了すると、INCA-SIPはモデルをノーマルモードおよびアクセラレータモードの一時停止状態にします。これにより、INCAの実験ウィンドウでいつ測定を開始しても高速なデータ収集が行えます。

### 4.3.2 ECUs

この機能を選択すると、**ECUs** ダイアログボックスが開きます。

モデリングされたECUについて、以下の3つのオプションが利用できます。

- **Add (追加)**
- **Remove (削除)**
- **Configure (設定)**

INCA-SIPは2つのタイプのECU（「Simulink<sup>®</sup> ECU」と「DLL ECU」）をサポートします。

「Simulink<sup>®</sup> ECU」は、測定変数と適合変数がSimulink<sup>®</sup>モデル内にブロックとして存在するECUを表します。

「Simulink<sup>®</sup> ECU」はさらに2つのタイプ（「Simulink<sup>®</sup> ECU」と「Referenced Model Group」）に分かれます。

Simulink<sup>®</sup> ECUはルートモデルへのマッピングや、参照モデルへの「1対1」マッピングが可能です。それに対してReferenced Model Groupは、参照モデルへの「1対多」マッピングのみ可能です。

「DLL ECU」は、Sファンクションがサードパーティ製DLLを使用してSファンクションの結果を演算する際に使用されるECUを表します。このようなDLLには、測定変数や適合変数として扱えるグローバル変数を含めることができます。DLL ECUは、これらの変数をINCAにおいて測定変数や適合変数として扱うようにするためのメカニズムを提供するものです。

#### Simulink<sup>®</sup>に新しいECUを追加する

1. **Tools** メニューから **INCA-SIP > ECUs** を選択します。

"ECUs" ダイアログボックスが開きます。

2. "ECUs" ダイアログボックスで、**Add** をクリックします。

"Add ECU" ダイアログボックスが開きます。

3. 以下のフィールドに情報を入力します。

**ECU Name** : ECUの名前。このフィールドではマクロ変数が使用できます。[「マクロ」\(ページ26\)](#) を参照してください。

**ECU Type** : ECUのタイプ (Simulink<sup>®</sup>またはDLL)

4. **Add** をクリックします。

Simulink<sup>®</sup>モデルに新しいECUが追加されます。

DLL ECUを使用する場合、INCA-SIPはSimulink<sup>®</sup>モデルと同じ場所に以下のフォルダを作成します。

- "+<SimulinkModelName>\_DLL" というルートフォルダ
- "+<SimulinkModelName>\_DLL" フォルダ内に、"+<ECUName>" というフォルダを作成します。このECU名は、有効なSimulink<sup>®</sup>パッケージ名です。
- "+<ECUName>" の下に2つのフォルダを作成します。

+Calibrations	このフォルダには、適合変数を追加する検索関数が含まれます。
+Measurements	このフォルダには、測定変数を追加する検索関数が含まれます。

モデルに保存済みコンフィギュレーションがない場合、DLL ECUは自動的に検出されます。保存済みコンフィギュレーションがある場合は、ECUウィンドウから手動でDLL ECUを追加することができます。

### Simulink®内のECUを設定する

1. **Tools** メニューから **INCA-SIP > ECUs** を選択します。  
"ECUs" ダイアログボックスが開きます。
2. ECUを選択します。
3. "ECUs" ダイアログボックスで、**Configure** をクリックします。  
"Configure ECU" ダイアログボックスが開きます。
4. 設定を入力します。  
詳細は、「[各タブの説明](#)」(下記) を参照してください。
5. **Save** をクリックします。

### 各タブの説明

#### General

このタブには、特定のECUのINCAへの接続方法に関する設定が含まれます。

<i>Enabled</i>	オンになっていると、当該ECUが使用されます。オフになっていると、INCA-SIPは当該ECUの処理をスキップします。
<i>Regenerate A2L and S19 file</i>	オフになっていると、INCA-SIPは既存のファイルの内容が最新のものであるとみなし、A2LファイルとS19ファイルを生成しません。これにより、接続時の処理時間を短縮することができます。INCA-SIPは、既存のファイルが最新の形式を使用していないと判断される場合、または足りないファイルがある場合は、このフラグを無視します。この場合、その旨を通知する情報メッセージが発行されます。
<i>ECU Name (read only)</i>	ECUの名前です。このフィールドではマクロ変数を使用できます。詳細は、「 <a href="#">マクロ</a> 」(ページ26) を参照してください。
<i>S19 File Name</i>	生成されるS19ファイル名を含むパスです。このフィールドではマクロ変数を使用できます。詳細は、「 <a href="#">マクロ</a> 」(ページ26) を参照してください。
<i>A2L File Name</i>	生成されるA2Lファイル名を含むパスです。このフィールドではマクロ変数を使用できます。詳細は、「 <a href="#">マクロ</a> 」(ページ26) を参照してください。
<i>SIPD File Name</i>	生成されるSIPDファイル名を含むパスです。SIPDファイルには、ECUのイベント、適合変数、測定変数、グループに関する説明がすべて含まれます。このフィールドではマクロ変数を使用できます。詳細は、「 <a href="#">マクロ</a> 」(ページ26) を参照してください。
<i>Subsystem (Simulink® ECUの場合のみ)</i>	当該ECUが表すモデル内のサブパスです。測定と適合の処理は、このパスとその下位パスに限定されます。このフィールドではマクロ変数を使用できません。詳細は、「 <a href="#">マクロ</a> 」(ページ26) を参照してください。

Referenced Model (Simulink® ECUの場合のみ)	当該ECUが表す参照モデルの名前です。N/Aに設定されていると、ECUはルートモデルを表します。
IPv4 Address	XCPスレーブがリスンするネットワークインターフェースのIPv4アドレスです。設定済みのIPアドレスが有効でなくなると、INCA-SIPは設定をループバックIP (127.0.0.1) に戻します。
XCP TCP Port	ECUのXCPスレーブサービスがリスン (listen) するXCPポートです。-1に設定すると、INCA-SIPは当該ECU用のランダムポートをアロケートし、対応するコンフィギュレーションに保存します。
All/Selection (Referenced Model Group ECUのみ)	<b>All</b> は接続フェーズにおいて存在するすべての参照モデルを含みます。参照モデルのサブセットをECUにマッピングすることもでき、モデルは <b>Model List</b> 機能を用いて選択できます。

## Events

イベントを定義して、ECU内のラスタを作成することができます。ラスタは秒単位で計算されます。これにより、SIPが期待するSimulink®モデルからの更新の頻度が決まります。このウィンドウでは、デフォルトのラスタとして0.1秒が設定されます。

Raster Timeが0の場合、DAQはSimulink®の各ソルバステップにおいてデータを送信します。小さいラスタ値を使用するか、ラスタ値を0にしてSimulink®で非常に小さい固定のソルバ時間を使用することは可能ですが、効率が低下し、実行速度が遅くなることがあります。

ラスタ値を固定ソルバ時間より小さくしたり、ソルバの倍数でないラスタを使用したりすることは不可能であり、INCAとの接続において無効と判定されます。これは、そのようなラスタはすべてトリガすることができず、一部がスキップされるからです。

可変ソルバ時間の使用は推奨されません。ラスタがヒットするかどうかや、Simulink®がどの程度の頻度で更新されるかを判定できないからです。

## Search Functions

2つのテーブルに、当該ECUに使用される検索スクリプトが表示されます。テーブルは、適合変数用と測定変数用のものに分かれます。**SHIFT**キーを押しながら複数のアイテムを同時に選択することができ、選択した一連のアイテムからいずれか1つを除外するには、**CTRL**キーを押しながらそのアイテムを選択します。

Simulink® ECUでは、Simulink®ブロック用の適合変数検索スクリプト (Basic Gain、Basic Lookup 1D、Basic Lookup 2D、Basic Lookup 3D、Table Lookup nD、Direct Lookups nD、Interpolation Blocks nD、Prelookups、Generic S-Function Masked Parameters) がデフォルトでサポートされています。

ここで"nD"は、1D~4Dのすべての次元を表します。

測定変数については、Simulink® ECUはブロックを接続する名前付きラインの検索をサポートします。各ラインにはそのデータソースが続きます。そしてさらに、測定変数を構成する出力ポートのランタイムオブジェクトがデータ収集に使用されます。

このようなECU用のDLL ECU検索スクリプトは、測定／適合されるカスタムDLLに合わせて作成する必要があります。カスタムスクリプトの作成方法の詳細は、「[DLL検索スクリプトの作成](#)」(ページ32)の章を参照してください。

同じ適合変数が複数のECUに存在する場合、1つのECUで適合を行っても、INCA内の他のECUのデータセットは変更されません。そのため、INCAからハードウェアの再初期化を行うとチェックサムエラーが発生します。

### 4.3.3 Configuration (設定)

この機能を選択すると、INCAプロジェクトの作成に関する設定を行えます。

#### **INCAプロジェクトの作成について設定する**

1. **Tools** メニューから **INCA-SIP > Configuration** を選択します。  
"Session Configuration" ダイアログボックスが開きます。
2. 設定を入力します。  
詳細は、「セッションに関する設定内容」(下記) と「INCAリモート操作に関する設定の内容」(次ページ) を参照してください。
3. **OK** をクリックします。

#### セッションに関する設定内容

<i>Compile Model</i>	<p>オフになっていると、<b>Connect to INCA</b> をクリックした際に Simulink<sup>®</sup>モデルのコンパイルが実行されません。</p> <p>測定変数を生成するにはコンパイルを行う必要があるため、Simulink<sup>®</sup> ECUがBasicNameSignalsという検索スクリプトを使用し、さらに Compile Model がオフになっていると、Regenerate A2L and S19 file がオフで、かつファイルがすでに存在する場合を除き、INCA上に測定変数が表示されません。</p>
<i>Open INCA On Connect</i>	<p>オンになっていると、INCAが自動的に開き、ワークスペースが自動的に設定されます。</p> <p>オフになっていると、必要に応じてA2LファイルとS19ファイルが生成され、XCPスレーブが起動しますが、INCAは開かず、設定も行われません。</p>
<i>INCA Database</i>	<p>INCAがINCA-SIP用にINCAデータベースを保存するディレクトリのパス。このフィールドではマクロ変数を使用できます。詳細は、「マクロ」(ページ26) を参照してください。</p>
<i>INCA Folder</i>	<p>INCAデータベース内のINCAフォルダの名前。このフィールドではマクロ変数を使用できます。詳細は、「マクロ」(ページ26) を参照してください。</p>
<i>INCA Workspace</i>	<p>INCAプロジェクトのワークスペースの名前。このフィールドではマクロ変数を使用できます。詳細は、「マクロ」(ページ26) を参照してください。</p>

### INCAリモート操作に関する設定の内容

Remote INCA Connection	オンになっていると、INCA-SIPはDCOM経由でリモートPC上のINCAを開きます。
INCA Machine DNS Name/IP Address	INCAリモートPCのホスト名またはIPアドレス。
Root Network Path	INCA PCとMATLAB <sup>®</sup> PCがアクセスできる共有UCNパス。

参照：「INCAのリモート操作」（ページ41）を参照してください。



#### 注記

デフォルトのSimulink<sup>®</sup> ECUには、A2Lの Emulation Mode Controlsが含まれません。詳細は、「INCA-SIPのエミュレーションモード」（ページ27）を参照してください。

**Show Workspace/DD Variables** がオンになっていると、INCAの**変数選択ダイアログボックス**において、各ECUに新しいグループWorkspaceVariablesが追加されます。このグループには、グローバルワークスペース、ローカルワークスペース、データディクショナリのいずれかに含まれるモデルが使用する変数がすべて含まれています。

#### 4.3.4 MDF Configuration

この機能を選択すると、MDFシミュラスに関する設定を行えます。

##### INCAのMDFシミュラスを設定する

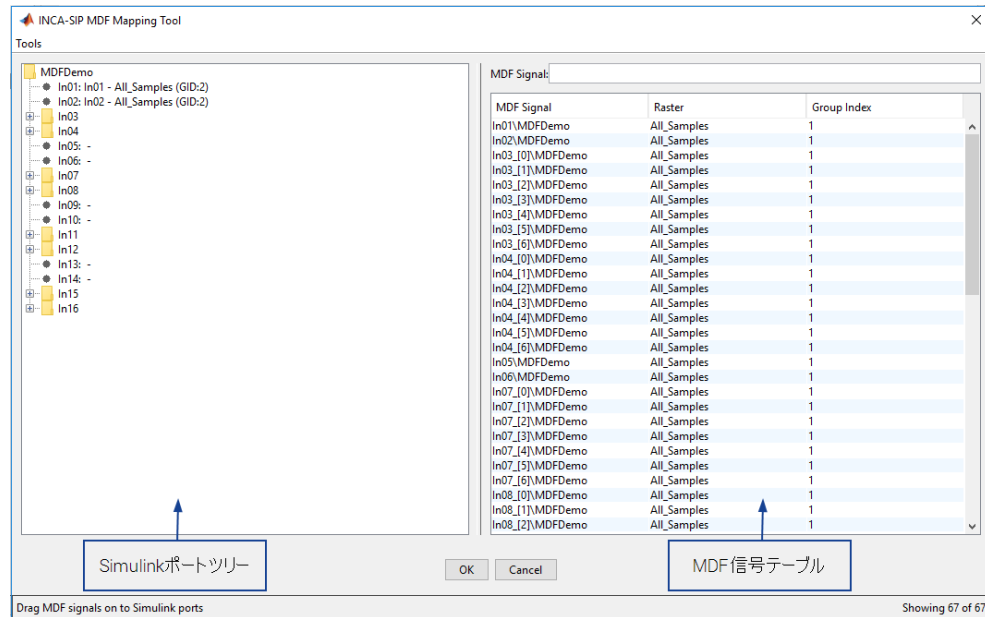
1. **Tools** メニューから **INCA > MDF Configuration** を選択します。  
"MDF Configuration" ダイアログボックスが開きます。
2. 設定を入力します。  
詳細は、「MDFに関する設定内容」（次ページ）を参照してください。
3. **OK** をクリックします。

## MDFに関する設定内容

<i>Use MDF Stimulus</i>	オンになっていると、INCA-SIPはMDFファイルからSimulink®の入力に外部ステイミュラスをロードすることを許可します。接続が確立すると、INCA-SIPはステイミュラスがロードされたことを検証します。
<i>Stimulus File</i>	ステイミュラス用にロードするMDFファイル。このフィールドではマクロ変数を使用できます。詳細は、「マクロ」(ページ26)を参照してください。
<i>Mapping File</i>	ステイミュラスファイルからSimulink®モデルの入力へのMDF信号のマッピングを定義する設定ファイル。ファイルが存在しない場合、INCA-SIPはマッピングツールを使用する際に自動的にファイルを生成します。このフィールドではマクロ変数を使用できません。詳細は、「マクロ」(ページ26)を参照してください。
<i>Mapping Tool</i>	<p>MDFマッピングツールを開きます。このツールで、MDF信号をSimulink®モデルの入力にマッピングすることができます。左側のツリーには、データを割り当てできるSimulink®のポートが表示されます。右側のテーブルには、MDFファイル内のすべてのデータが表示されます。MDFデータを "Mapping Value" 列にドラッグすることにより、その列のデータを入力ポートにマッピングすることができます。</p> <p>INCA-SIPは、選択したMDFデータをtime = 0からtime = &lt;simulation time&gt; までロードします。シミュレーション時間が延長された場合は、再度マッピングツールを利用して、新しく指定されたシミュレーション時間に対してデータをロードする必要があります。</p>



## MDFマッピングツールの使用



### 信号をマッピングする

- "MDF Mapping Tool" ウィンドウで、MDF信号をSimulink®ポートにドラッグ&ドロップします。

Simulink®ポートはこのアイコンで示されています：

または

1. Simulink®ポートを選択します。
2. 目的のMDF信号をダブルクリックしてそのポートにマッピングします。

### MDFマッピングの消去

#### すべてのマッピングを消去する

- "MDF Mapping Tool" ウィンドウで、**Tools > Clear All Mappings** を選択します。

または

- Simulink®ツリーを右クリックして **Clear All Mappings** を選択します。

#### 個々のマッピングを消去する

1. ポートまたはツリーノードを選択します。
2. <DEL> キーを押します。

または

- Simulink®ツリーを右クリックして **Clear Selected** を選択します。  
選択されたノードとそのサブノードがすべて消去されます。

#### 注記

ルートノードにおいては、この操作は行えません。

## Simulink®ポートの更新

MDF信号をSimulink®ポートにマッピングするには、モデルをコンパイルする必要があります。INCA-SIPは、最後に使用されたコンフィギュレーションの情報がある場合はそれを参照し、マッピングツールが開くたびに再コンパイルが行われるのを回避します。ただしこのコンフィギュレーションは、すでに古くなっていて使用できない場合があります。

### Simulink®ポートツリーを更新する

1. "MDF Mapping Tool" ウィンドウで、**Tools** をクリックします。
2. "Mapping" ダイアログボックスで、Simulink®ポートを再ロードします。

## MDF信号データの更新

パフォーマンスの向上のため、MDFデータはモデルワークスペースの一部として保存されます。このデータは古くなってしまふ可能性があるため、マニュアル操作で更新する必要があります。これには、MDFファイルのパスと名前は変わらずに、内容が変更されたり、他のものと入れ替わったりするようなケースが考えられます。

### MDF信号を更新する

1. "MDF Mapping Tool" ウィンドウで、**Tools** をクリックします。
2. "Mapping" ダイアログボックスで、MDFデータを再ロードします。



### 注記

INCAに接続すると、以下のような場合にデータが自動的に再ロードされます。

- － INCAがMDFデータをロードすると、シミュレーション時間が考慮され、必要な量のデータのみロードされます。シミュレーション時間が長くなると、データが自動的に再ロードされます。
- － MDFコンフィギュレーションウィンドウにおいてMDFのパスが変更されると、キャッシュされていたMDFデータが削除されるため、接続時に自動的に再ロードされます。
- － Simulink®ポートが変更され、マッピングが無効になると、そのポートのMDFデータは削除され、接続時において、再マッピングされていれば自動的に再ロードされます。

## 4.4 マクロ

INCA-SIPは文字列内で使用できるマクロ変数をサポートし、これによって動的な文字列を定義することができます。以下のようなマクロ変数が使用できます。

- － `$(SystemName)` : モデル名
- － `$(ECUName)` : ECU名 ("Configure ECUs" ダイアログボックスでのみ使用可能)
- － `$(INCADataDir)` : INCAデータディレクトリのパス
- － `$(INCASIPDir)` : INCA-SIPのインストールパス

INCA-SIPでは、マクロ変数内で環境変数（例：`%(APPDATA)%`）を使用することができます。

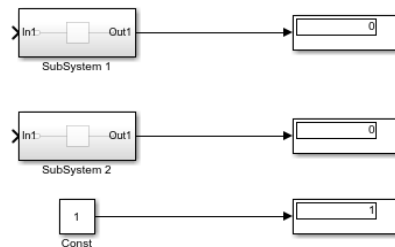
## 4.5 命名規則

### 4.5.1 適合変数と測定変数

INCA-SIPがECUを処理する際には、A2Lファイルを生成します。そのため、測定変数と適合変数の名前は、すべてA2Lの命名規則に準拠している必要があり、準拠していない名前は適切に変更されます。また重複する名前については、INCA-SIPはその名前にシーケンス番号（\_x）を付加します。番号の範囲は1から無限大です。

### 4.5.2 グループ

測定変数と適合変数をグループ内に存在するものとして定義することにより、INCAの変数の選択ダイアログボックス内にそれらの変数を同じグループのものとしてまとめて扱うことができます。同じECU内に重複するグループ名があると、INCA-SIPはその名前にシーケンス番号（\_x）を付加します。番号の範囲は1から無限大です。



"ETAS\_SIP" というグループには、INCA-SIPのエミュレーションモードに関連した制御変数が含まれます。

## 4.6 INCA-SIP のエミュレーションモード

INCA-SIPのモデルシミュレーションには2つのモードがあります。

- － リアルタイムエミュレーションモード
- － 高速エミュレーションモード

このモードは、"Default Simulink ECU"のA2Lファイル内の"ETAS-SIP"グループに含まれる適合変数で制御することができます。

### 4.6.1 リアルタイムエミュレーションモード

リアルタイムエミュレーションモードにおいてINCA-SIPアドオンは、実時間にほぼ等しい速度でのシミュレーションを試みます。モデルが実時間よりも高速にシミュレートされている場合は、INCA-SIPは速度を調整して実時間に近づけます。一方、モデルシミュレーションの速度が実時間よりも遅い場合は、INCA-SIPは何も調整を行いません。

このモードは、シミュレーション中にモデルを適合し、シミュレーションに対する適合の効果を視覚化するような場合に適しています。

このモードを有効にするには、INCAの実験に`RealTimeBlock.mode`という適合変数を追加し、その値として"Real Time"を選択します。加速係数は、`RealTimeBlock.realTimeMultiplier`という適合変数の値により以下のように調整できます。

1	実時間になります。
1未満の値	実時間よりも遅くなります。
1より大きい値	実時間よりも速くなります。
0以下の値	INF（無限大）と同じ意味を持ち、高速エミュレーションモードと同等のものになります。

これらの適合変数は、"Default Simulink ECU"の"ETAS\_SIP"グループに含まれています。

#### 4.6.2 高速エミュレーションモード

高速エミュレーションモードにおいてINCA-SIPアドオンは、モデルシミュレーション時に、データ損失が発生しない範囲において可能な限り高速でINCAにデータを送信します。

そのためこのモードは、データの視覚化を行うには実用的ではありません。一方でこのモードは、オフラインで適合変数の値を変更し、それに基づきモデルを一定期間シミュレートするような場合に適しています。モデルシミュレーションは比較的短時間で終了しますが、シミュレーションで得られた測定データは、MDA（Measure Data Analyzer：測定データアナライザ）を使用してオフラインで分析できます。

このモードを有効にするには、INCAの実験に`RealTimeBlock.mode`という適合変数を追加し、その値として"Fast Emulation"を選択します。これらの適合変数は、"Default Simulink ECU"の"ETAS\_SIP"グループに含まれています。

#### 4.7 INCA-SIPとブロック削減

MATLAB<sup>®</sup>は、デフォルトにおいて「ブロック削減」をサポートしています。シミュレーションに影響を与えないと見なされる適合変数と測定変数は、シミュレーションから削除して最適化することができます。INCA-SIPはこれらの適合変数と測定変数を無視します。この「ブロック削減」は、シミュレーション設定において有効／無効を選択することができます。詳細は、MATLAB<sup>®</sup>のドキュメントを参照してください。

#### 4.8 シミュレーションモード

MATLAB<sup>®</sup>にはいくつかのシミュレーションがあります。その中でINCA-SIPがサポートしているモードは、以下のとおりです。

- － ノーマルモード
- － アクセラレータモード
- － ラピッドアクセラレータモード

### 注記

ラピッドアクセラレータモードを使用するには、有効なC++コンパイラが登録されている必要があります。詳細は、「[C++コンパイラの登録](#)」(ページ13)を参照してください。

### 注記

MATLAB®の新バージョンでの機能変更に伴い、ラピッドアクセラレータモードをサポートしているのは2018b以前のMATLAB®のみとなっています。

## 4.9 参照モデル (Model Referenceブロック) のサポート

参照モデルは、Simulink®における標準モデルです。これらのモデルは親モデルから参照され、親モデルは参照モデルの入力と出力にリンクすることができます。ECUを設定する際に、参照するモデルの名前をコンボボックスから選択することができます。

参照モデルは、INCA上で個別のECUとして表示されます。

## 4.10 適合操作の拒否


MATLAB®/Simulink®は、INCAから送信された適合値を拒否する場合があります。一例として、ある適合変数をマップの値データおよび軸データとして同時に使用するケースが挙げられます。軸データは単調に増加する必要がありますが、値データにはそのような制限はありません。そのため、その変数が値データとして変更されて単調増加性が失われると、MATLAB®はこの適合操作を拒否します。そのような場合は、XCPによって返されるエラーコードCALMEM\_REQUESTNOTVALIDがINCAモニタウインドウに表示されます。

## 4.11 ECUのビジー状態


処理時間の長い要求を行うと、ECUがビジー状態になる場合があります。一例としては、大きな配列のすべての要素の値を同時にインクリメント/デクリメントするようなケースが挙げられます。このような場合、ECUはこの要求を実行するためにビジー状態になり、ECUがこの状態から解放されて次のコマンドを受け取れる状態になるまで、INCAはダイアログボックスを表示します。

## 4.12 実験の開始と停止

### 実験を開始する

1. INCAの実験ウインドウで、 をクリック、または **F11** キーを押します。  
Simulink®におけるシミュレーションの実行状態が、以下のように変化します。  
シミュレーションが停止状態であった場合は、シミュレーションが開始されます。  
または  
シミュレーションが一時停止状態であった場合は、シミュレーションが続行されます。

### 実験を停止する

1. INCAの実験ウィンドウで、 をクリック、または **F9** キーを押します。  
Simulink®におけるシミュレーションの実行状態が、以下のように変化します。  
ノーマルモードまたはアクセラレータモードにおいては、一時停止します。  
または  
ラピッドアクセラレータモードにおいては、停止します。



#### 注記

シミュレーション状態が上記のように変化するのは、INCAの実験に測定変数が割り当てられている場合に限りです。

## 4.13 Simulink.SignalとAsap2.Signalのサポート

Simulink® ECUにおいて、行の名前にSimulink.SignalまたはAsap2.Signalが使用されている場合、測定変数の"Description"、"Unit"、"Min"、"Max"は、Simulink.SignalまたはAsap2.Signalの変数から取得されます。

## 4.14 Simulink.ParameterとAsap2.Parameterのサポート

Simulink® ECUにおいて、ブロック内にSimulink.ParameterまたはAsap2.Parameterというタイプのワークスペース変数の内容を持つパラメータがある場合、適合変数の"Description"、"Unit"、"Min"、"Max"は、Simulink.ParameterまたはAsap2.Parameterの変数から取得されます。

## 4.15 各種メッセージ（エラー、警告、情報）

INCA-SIPは、接続とシミュレーションの処理中に各種メッセージ（エラー、警告、情報）を発行します。

開いているSimulink®モデルに関するメッセージが、Simulink®のDiagnostic Viewerに出力されます。

### Simulink®のDiagnostic Viewerを開く

1. Simulink®エディタで、**View > Diagnostic Viewer** をクリックします。  
エラーメッセージが発行された時には、MATLAB®でも診断ビューアーが開きます。

## 4.16 DLLモード

MATLAB®では、実行中にDLLを内部的に使用するSファンクションを作成することができます。INCA-SIPのDLLモードは、DLL内のグローバル変数の適合／測定を行う機能を提供するものです。

### 4.16.1 DLLの作成


以下の場所に、シンプルなVisual Studioプロジェクトが保存されています。

<drive>:\ETASData\INCA7.5\Demo\AddOn\_INCA-SIP7.5\SIPDllDemo

"SIPDllDemo"フォルダには以下のサブフォルダが含まれます。

"legacy_alg"	INCAから適合と測定を行うためのDLLのサンプルが含まれます。ソリューションのビルド時に、DLLファイルとマップファイルが"x64"フォルダにコピーされます。マップファイルは、モジュールと変数のオフセットを決定する際に使用されます。
"x64"	MATLAB®の64ビットバージョンで使用され、ここにモデルとDLL検索スクリプトが保存されます。このプロジェクトを使用するには、 <code>mex legacy_alg_sfcn.c</code> を用いて <code>legacy_alg_sfcn.c</code> をコンパイルする必要があります。

ソースコード内には、測定または適合を行えるグローバル変数がいくつか含まれています。

 **注記**

SIPとDLLとの間に同期メカニズムは存在しません。そのため、INCA-SIPのみが適合を行うのではない場合、競合状態が発生する可能性があります。

INCA-SIPのDLLモードは以下の型をサポートしています。

INCA-SIP DLLの型	Cの型	サイズ	注記
<code>MScripts.DllMode. DllAsap2Object.UINT8</code>	<code>uint8_t</code>	8	
<code>MScripts.DllMode. DllAsap2Object.UINT16</code>	<code>uint16_t</code>	16	
<code>MScripts.DllMode. DllAsap2Object.UINT32</code>	<code>uint32_t</code>	32	
<code>MScripts.DllMode. DllAsap2Object.UINT64</code>	<code>uint64_t</code>	64	適合変数のみ
<code>MScripts.DllMode. DllAsap2Object.INT8</code>	<code>int8_t</code>	8	
<code>MScripts.DllMode. DllAsap2Object.INT16</code>	<code>int16_t</code>	16	
<code>MScripts.DllMode. DllAsap2Object.INT32</code>	<code>int32_t</code>	32	

INCA-SIP DLLの型	Cの型	サイズ	注記
MScripts.DllMode. DllAsap2Object.INT64	int64_t	64	
MScripts.DllMode. DllAsap2Object.BOOL	uint8	8	
MScripts.DllMode. DllAsap2Object.FLOAT	float	32	IEEE-754
MScripts.DllMode. DllAsap2Object.DOUBLE	double	64	IEEE-754



### 注記

int64とuint64は適合変数でのみ使用できます。測定変数で使用すると、変数はINCAの変数選択ダイアログボックスに表示されますが、実験に割り当てることができません。

## 4.16.2 DLL検索スクリプトの作成

以下の場所にサンプルスクリプトが保存されています。

```
<drive>:\ETASData\INCA7.5\Demo\AddOn_INCA-
SIP7.5\SIPDLLDemo\x64\+SipDllDemo_DLL\+DllDemo
```

各検索スクリプトは、以下の関数シグネチャで定義される関数を使用して定義されます。

```
<Function Name>(dataModel, sRootSystem)
```

オブジェクト"dataModel"に、適合変数と測定変数を追加します。"sRootSystem"は、モデル名の文字列です。

### DataModel関数コール

関数	addCalibration (calibration)	
パラメータ	calibration	Calibration Description Object (適合変数ディスクリプションオブジェクト)
戻り値	なし	
説明	INCAに表示するデータモデルに指定のCalibration Description Objectを追加します。適合変数に割り当てられている軸は、すべて自動的に追加されます。	
関数	addCalibrationAxis (axis)	



パラ メータ	axis	Calibration Axis Description Object (適合軸ディスクリプションオブ ジェクト)
戻り 値	なし	
説明	INCAに表示するデータモデルに指定のCalibration Axis Objectを追加します。この関数は、どの適合オブジェクトにも割り当てられていない独立した適合軸を表示する場合にのみ使用します。	
関数	addMeasurement (measurement)	
パラ メータ	measurement	Measurement Description Object (測定変 数ディスクリプションオブジェクト)
戻り 値	なし	
説明	INCAに表示するデータモデルに指定のMeasurement Description Objectを追加します。	

DLLの適合変数を定義してプロパティをすべて設定した後に、それをデータモデルに追加します。これは以下のように行います。

```
dataModel.addCalibration(calibration);
```

### 注記

1つの軸オブジェクトは1つの適合変数にのみリンクできます。同じモジュールオフセットを軸として別の適合変数にリンクするには、軸のインスタンスを別に生成する必要があります。

適合変数にすでに軸がリンクされている場合は軸を追加する必要はありません。軸は自動的に追加されます。

適合変数にリンクされていない軸は、以下のように追加します。

```
dataModel.addCalibrationAxis(axis);
```

DLLの測定変数を定義してプロパティをすべて設定した後に、それをデータモデルに追加します。これは以下のように行います。

```
dataModel.addMeasurement(measurement);
```

### 4.16.3 DllAsap2ObjectFactoryを用いたオブジェクトの生成

#### 適合変数の生成

MScripts.DllMode.DllAsap2ObjectFactoryは、適合変数を生成するための以下のメソッドを提供します。

関数	createScalarCalibration(name, moduleOffset, cDataType, dllName, groupPath)	
パラ	name	INCAで使用される名前
メータ	moduleOffset	DLL内の変数の、モジュールオフセットアドレスのHEX文字列
	cDataType	モジュールオフセットのデータ型
	dllName	モジュールオフセットが含まれるDLLの名前
	groupPath	INCAにおいて適合変数が含まれるグループ
戻り値	Calibration Description Object (適合変数ディスクリプションオブジェクト)	
説明	INCAにおいて単純な適合変数として扱われるスカラー適合変数を生成します。	
関数	createTableCalibration(name, moduleOffset, cDataType, dllName, groupPath)	
パラ	name	INCAで使用される名前
メータ	moduleOffset	DLL内の変数の、モジュールオフセットアドレスのHEX文字列
	cDataType	モジュールオフセットのデータ型
	dllName	モジュールオフセットが含まれるDLLの名前
	groupPath	INCAにおいて適合変数が含まれるグループ
	arraySize	ベクトルフォーマットで記述された配列のサイズ (例: [1 3], [2 3 4])
戻り値	Calibration Description Object (適合変数ディスクリプションオブジェクト)	
説明	INCAにおいて適合変数として扱われるn次元の適合変数を生成します。	

関数	<code>createFixedAxis(offset, distance, numberOfAxisPoints, asap2Description)</code>	
パラ	<code>offset</code>	固定軸の先頭の値
メータ	<code>distance</code>	インクリメントサイズ
	<code>numberOfAxisPoints</code>	軸ポイント（軸に含まれる要素）の数
戻り値	Fixed Axis Description Object（固定軸ディスクリプションオブジェクト）	
説明	固定軸を生成します。この軸は、1つのn次元の適合変数にのみリンクできます。注記：すべてのn次元の適合変数には、デフォルトで固定軸が割り当てられています。	

関数	<code>createCalibrationAxis(name, moduleOffset, cDataType, dllName, groupPath, arraySize)</code>	
パラ	<code>name</code>	INCAで使用される名前
メータ	<code>moduleOffset</code>	DLL内の変数の、モジュールオフセットアドレスのHEX文字列
	<code>cDataType</code>	モジュールオフセットのデータ型
	<code>dllName</code>	モジュールオフセットが含まれるDLLの名前
	<code>groupPath</code>	INCAにおいて適合変数が含まれるグループ
	<code>arraySize</code>	ベクトルフォーマットで記述された配列のサイズ（例：[1 3]、[2 3 4]）
戻り値	Calibration Axis Description Object（適合軸ディスクリプションオブジェクト）	
説明	軸を生成します。この軸は、n次元の適合変数にリンクするか、テーブル適合変数にリンクせずにデータモデルに追加することができます。	

関数	<code>createCalibrationAxisWithMeasurementKey (name, moduleOffset, cDataType, dllName, groupPath, arraySize, sMeasurmentKey, hDataModel)</code>	
パラ	<code>name</code>	INCAで使用される名前
メータ	<code>moduleOffset</code>	DLL内の変数の、モジュールオフセットアドレスのHEX文字列
	<code>cDataType</code>	モジュールオフセットのデータ型
	<code>dllName</code>	モジュールオフセットが含まれるDLLの名前
	<code>groupPath</code>	INCAにおいて適合変数が含まれるグループ
	<code>arraySize</code>	ベクトルフォーマットで記述された配列のサイズ (例: [1 3]、[2 3 4])
	<code>sMeasurmentKey</code>	この軸に割り当てられる測定変数の測定キー
	<code>hDataModel</code>	データモデルのハンドル
戻り値	Calibration Axis Description Object (適合軸ディスクリプションオブジェクト)	
説明	軸を生成します。この軸は、n次元の適合変数にリンクするか、テーブル適合変数にリンクせずにデータモデルに追加することができます。	



### 注記

モジュールオフセットと配列サイズの妥当性は、ユーザーが責任を持って確認する必要があります。

### 適合変数に軸をリンクする

DLL Axis CalibrationをDLL Table Calibrationにリンクするには、Calibration Description Objectsに対して以下を行います。

関数	<code>setAxis (axis, index)</code>	
パラ	<code>axis</code>	Calibration Axis Description Object (適合軸ディスクリプションオブジェクト)
メータ	<code>index</code>	軸のインデックス (例: 1=X軸、2=Y軸、3=Z軸、4=W軸)
戻り値	なし	
説明	軸をCalibration Description Objectに割り当てます。	

## 測定変数の生成

MScripts.DllMode.DllAsap2ObjectFactoryは、測定変数を生成するための以下のメソッドを提供します。

関数	<code>createScalarMeasurement(name, moduleOffset, cDataType, dllName, groupPath, arraySize)</code>	
パラメータ	<code>name</code>	INCAで使用される名前
メータ	<code>moduleOffset</code>	DLL内の変数の、モジュールオフセットアドレスのHEX文字列
	<code>cDataType</code>	モジュールオフセットのデータ型
	<code>dllName</code>	モジュールオフセットが含まれるDLLの名前
	<code>groupPath</code>	INCAにおいて適合変数が含まれるグループ
戻り値	Measurement Description Object (測定変数ディスクリプションオブジェクト)	
説明	INCAで単純な測定変数として扱われるスカラー測定変数を生成します。	
関数	<code>createArrayMeasurement(name, moduleOffset, cDataType, dllName, groupPath, arraySize)</code>	
パラメータ	<code>name</code>	INCAで使用される名前
メータ	<code>moduleOffset</code>	DLL内の変数の、モジュールオフセットアドレスのHEX文字列
	<code>cDataType</code>	モジュールオフセットのデータ型
	<code>dllName</code>	モジュールオフセットが含まれるDLLの名前
	<code>groupPath</code>	INCAにおいて適合変数が含まれるグループ
	<code>arraySize</code>	ベクトルフォーマットで記述された配列のサイズ (例: [1 3]、[2 3 4])
戻り値	Measurement Description Object (測定変数ディスクリプションオブジェクト)	
説明	INCAにおいて単純な測定変数として扱われるn次元の測定変数を生成します。	

**i** 注記

測定変数と適合変数の名前はASAP2の命名規則に従う必要があります。従っていない場合は、INCA-SIPによって適切な名前に変更されます。ECU内で同じ名前がすでに使用されている場合は、名前にシーケンス番号（\_x）が付加されます。番号の範囲は1から無限大です。

**i** 注記

モジュールオフセットと配列サイズの妥当性は、ユーザーが責任を持って確認する必要があります。

## 一般オプション

以下の関数呼び出しにより、オブジェクトに属性値（説明文、最小値、最大値など）を持たせることができます。

関数	setAsap2Description (sDescriptionText)	
パラメータ	sDescriptionText	説明文
戻り値	なし	
説明	オブジェクトの説明文を設定します。	

関数	setAsap2MinValue (dMin)	
パラメータ	dMin	最小値
戻り値	なし	
説明	最小値を設定します。	

関数	set2AsapMaxValue (dMax)	
パラメータ	dMax	最大値
戻り値	なし	
説明	最大値を設定します。	

関数	addGroupPath (sGroupPath)	
パラメータ	sGroupPath	INCAにおいて適合変数が含まれるグループ
戻り値	なし	
説明	INCAにおいてオブジェクトが含まれるグループのパスを追加します。	

#### 4.17 データセットをMDFファイルに書き込む

MDFの書き込み機能により、あらゆるSimulink<sup>®</sup>データセット（Simulink<sup>®</sup>信号のロギングにより生成されたものなど）をMDFファイルに出力することができます。

##### データセットをMATLAB<sup>®</sup>コマンドラインでMDFファイルに書き込む

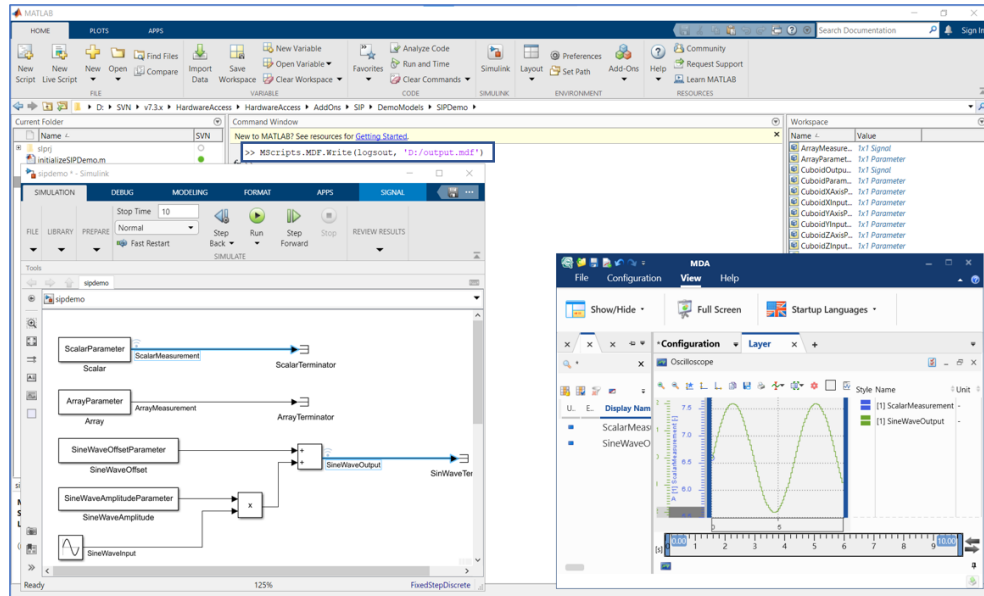
- コマンド `MScripts.MDF.Write (parameter1, 'parameter2', parameter3)` を呼び出します。

<i>parameter1</i>	MDFファイルに書き込みたいMATLAB <sup>®</sup> /Simulink <sup>®</sup> データセット
<i>parameter2</i>	書き込みたいMDFファイルのパス
<i>parameter3</i>	このパラメータは、MDFライターに対して、データセット内の各信号の最新の値を最新のタイムスタンプにコピーすることを指示するものです。追加されるこのデータポイントは、MDFビューア（ETAS MDAなど）が1つのポイントだけでなく1行のデータを表示する際に役立ちます。設定できる値はtrue/falseです。 これらのパラメータはオプションです。

##### 例：

```
MScripts.MDF.Write(logsout, 'D:/output.mdf', false)
```

以下のスクリーンショットは、MATLAB<sup>®</sup>/Simulink<sup>®</sup>とMDAにおけるMDF書き込み機能の利用例を表しています。信号の記録には、Simulink<sup>®</sup>の信号ロギング機能を使用します。その後、MDF書き込みコマンド `MScripts.MDF.Write (logsout, 'D:/output.mdf')` を実行し、その出力をMDAにロードします。



## 注記

**Sample Time** を **-1** にすることにより、Simulink<sup>®</sup> がチェックごとに信号を反復値と共にロギングするように設定することができます。

モデルのライフタイムコールバックを使用すると、モデルの一時停止、終了、保存などが行われるたびに、Simulink<sup>®</sup> が自動的にMDF書き込み機能呼び出すようにセットアップすることができます。モデルコールバックの設定方法については、MATLAB<sup>®</sup> の公式ドキュメントを参照してください。

## GUIを使用してMDFファイルを書き込む

1. **Tools** メニューから、**INCA-SIP > MDF Configuration** を選択します。  
"MDF Configuration" ダイアログボックスが開きます。
2. MDF Export領域で、**Use Signal Logging** をオンにします。  
または  
**Change Settings** をクリックし、Simulink<sup>®</sup> Configuration Parameterダイアログボックスの **Data Import/Export** タブで、信号ロギング設定をオンにします。
3. **Export File** フィールドで、MDFファイルのパスと名前を指定します。
4. 信号値が変化するまでMDFエクスポートファイル内の値を最後の信号値にしておくには、**Hold Last Value.** をオンにします。
5. Simulink<sup>®</sup> で **Stop** ボタンをクリックしたときに自動的にMDFエクスポートファイルが作成されるようにするには、**Automatically Export on Stop** をオンにします。
6. MDFエクスポートファイル名にインクリメントカウンタが自動的に付加されるようにするには、**Auto Increment Filename** をオンにします。

ファイル名の衝突があった場合は、INCA-SIPはファイル名の末尾にカウンタ `_xx` を付加します。新しいセッションにおいて、またはファイル名が変更された場合は、INCA-SIPはカウンタを `_01` にリセットします。



 **注記**

このオプションがオフになっていると、作成済みの同名のMDFエクスポートファイルは、上書きされてしまいます。

7. すぐにMDFファイルをエクスポートするには、**Export Now** をクリックします。
8. **適用** をクリックします。
9. **OK** をクリックします。

## 4.18 INCAのリモート操作

INCAとSimulink<sup>®</sup>がそれぞれ別のPCにインストールされている場合、INCAからSimulink<sup>®</sup>モデルの適合を行うには、INCAのリモートアクセスが必要になります。

INCAをリモートで実行するには、INCA-SIPとINCAが、INCA PCとリモートPCにインストールされている必要があります。

### INCAのリモート接続のセットアップ

INCAのリモート接続をセットアップするには、以下のように操作します。

- 「Windowsコンポーネントサービスを設定する」(下記)
- 「ユーザーまたはグループをINCAのDCOM構成に追加する」(下記)
- 「MATLAB<sup>®</sup>内でINCA-SIPを設定する」(次ページ)

#### Windowsコンポーネントサービスを設定する

1. Windows管理ツールの **コンポーネントサービス** を開きます。
2. **Component Services** フォルダを展開します。
3. **マイコンピュータ** を右クリックして、ショートカットメニューから **プロパティ** を選択します。
4. **既定のプロパティ** タブで、以下のように設定します。
  - **このコンピューター上で分散COMを有効にする** をオンにします。
  - **既定の認証レベル** を **接続** にします。
  - **既定の偽装レベル** を **識別する** にします。
5. **適用** をクリックします。
6. **OK** をクリックします。

#### ユーザーまたはグループをINCAのDCOM構成に追加する

1. Windows管理ツールの **コンポーネントサービス** を開きます。
2. **コンポーネントサービス** フォルダを展開します。
3. **DCOMの構成** フォルダ内の **INCA7xServer** を右クリックし、ショートカットメニューから **プロパティ** を選択します。
4. **セキュリティ** タブを選択します。
5. **起動とアクティブ化のアクセス許可** オプションの **カスタマイズ** をオンにします。
6. **編集** をクリックします。

7. **グループ名またはユーザー名** リストの **追加** をクリックし、新しいユーザーまたはグループを追加します。
8. **OK** をクリックします。
9. **アクセス許可: SYSTEM** リストで、すべてのアイテムについて **許可** をオンにします。
10. **OK** をクリックします。
11. **適用** をクリックします。
12. **OK** をクリックします。

ユーザーまたはグループを追加する際には、以下の点が守られていることを確認してください。

- － 追加したユーザーとMATLAB<sup>®</sup>を使用するユーザーが同じであること
- － 追加したグループにMATLAB<sup>®</sup>を使用するユーザーが含まれること

ファイアウォールを介したDCOM通信を許可するには、ファイアウォールの設定を変更する必要があります。詳細は、マイクロソフトのドキュメントを参照してください。

#### **MATLAB<sup>®</sup>内でINCA-SIPを設定する**

1. **Tools** メニューから **INCA-SIP > Configuration** を選択します。
2. 以下の設定をオンにします。
  - **Open INCA On Connect**
  - **Remote INCA Connection**
3. **INCA Machine DNS Name/IP Address** フィールドに、INCAリモートPCのホスト名またはIPアドレスを入力します。
4. **Root Network Path** フィールドに、INCA PCとMATLAB<sup>®</sup> PCがアクセスできる共有UCNパスを入力します。

**Root Network Path** は、INCAデータベース、A2Lファイル、S19ファイルを保存する際に使用されます。相対パスを指定した場合は、データはリモートネットワークパスに保存されます。その際、パスは設定されたUCNパスに相対して解決されます。絶対パスを指定した場合は、INCAリモートPCがINCAデータベースパスにアクセスできることをユーザーの責任において確認してください。A2LファイルとS19ファイルのパスは、両方のPCから同じUCNを使用してアクセスされます。

#### **注記**

各ECUにはそれぞれTCPポートがあります。各ECUポートは、ファイアウォールによってMATLAB<sup>®</sup>マシンの着信ポートとして許可される必要があることに注意してください。さらに各ECUには、INCAがXCPスレーブへの接続に使用するIPアドレスがあります。選択したアドレスがINCA PCと同じサブネット上にない場合、INCA-SIPはINCA PCと同じサブネット上にあるインターフェースを見つけるを試みます。インターフェースが見つからない場合は、手動操作で適切なIPアドレスを特定する必要があります。

## 4.19 カスタムフック

INCA-SIPは以下の機能を持つフックを提供します。

- － カスタムフィルタリング
- － 測定変数と適合変数のカスタムネーミング
- － グループのカスタムネーミング
- － 変数のカスタマイズ

上記の機能により以下のようなことが行えます。

- － 不要な適合変数と測定変数のフィルタリング
- － 測定変数、適合変数、グループの独自の命名規則を作成
- － ディスプレイID（表示用の名前）を設定

この「カスタムフック」は、MATLAB<sup>®</sup>関数への関数ポインタです。

MScripts.Configuration内のこれらのフックをセットするには、以下を呼び出します。

- － 「setVariableFilterFunctionHandle」（下記）
- － 「setVariableCustomizationFunctionHandle」（ページ45）
- － 「setVariableRenamingFunctionHandle」（ページ47）
- － 「setGroupRenamingFunctionHandle」（ページ48）

MScripts.Configuration内のこれらのフックをリセットするには、以下を呼び出します。

- － clearGroupRenamingFunctionHandle
- － clearVariableCustomisationFunctionHandle
- － clearVariableFilterFunctionHandle
- － clearVariableRenamingFunctionHandle

MScripts.Configurationクラスはシングルトン（singleton）であり、MATLAB<sup>®</sup>インスタンスに対してグローバルです。

### **MScripts.configurationクラスのインスタンスを取得する**

1. MScripts.Configuration.getInstance()を実行します。

構成クラスに加えられた変更は、MATLAB<sup>®</sup>を閉じる際に失われます。そのため、カスタムフックはMATLAB<sup>®</sup>を開くたびに設定する必要があります。

#### 4.19.1 setVariableFilterFunctionHandle

カスタムフックが設定されていると、データモデルに追加された各適合変数と測定変数がカスタムフックに渡されます。カスタム関数に受け付けられた適合変数はデータモデルに追加されますが、受け付けられなかったものはスキップされ、その適合変数または測定変数に関して残りのカスタムフックは呼び出されません。

## フック関数のシグネチャ

関数	<code>bKeep = variableCustomFilter(type, variableInfoWrapper)</code>	
パラメータ	<code>bkeep</code> が返ります。	
	<code>true</code>	適合変数または測定変数が保持されています。
	<code>false</code>	適合変数または測定変数が除外されています。
	<code>type char[]</code>	以下のいずれかの定数を使用できます。 <code>MScripts.DataModelBase.SimulinkWorkspaceCalibration</code> <code>MScripts.DataModelBase.DLLCalibrationAxis</code> <code>MScripts.DataModelBase.DLLCalibration</code> <code>MScripts.DataModelBase.DLLMeasurement</code> <code>MScripts.DataModelBase.SimulinkCalibrationAxis</code> <code>MScripts.DataModelBase.SimulinkCalibration</code> <code>MScripts.DataModelBase.SimulinkMeasurement</code>
	<code>variableInfoWrapper</code>	<code>MScripts.VariableWrapper</code> のインスタンス <code>variableInfoWrapper</code> は、現在処理されている適合または測定を判定するために使用されます。 <code>variableInfoWrapper</code> の詳細な使用方法については、 <a href="#">「MScripts.VariableWrapper」(ページ50)</a> を参照してください。

 注記

このフックにおいては、`getters`のみを`variableInfoWrapper`で使用してください。この適合変数または測定変数に対する変更内容は、`setVariableCustomisationFunctionHandle`フック内で設定することができます。

"SIPDEMO"フォルダ内の`CustomHooks variableCustomFilter.m script`に、このフックのサンプルが含まれています。

カスタムフックを使用する前の変数の選択

All Sources (Not Filtered, 7/7 Visible)		
	Name	Role
	<u>_1_D_Lookup_Table</u>	
	CalibrationToFilter	
	LineToFilter	
	OriginalCalibrationName	
	OriginalLineName	
	RealTimeBlock.mode	
	RealTimeBlock.realTimeMultiplier	

カスタムフックを使用した後の変数の選択

All Sources (Not Filtered, 5/5 Visible)		
	Name	Role
	<u>_1_D_Lookup_Table</u>	
	OriginalCalibrationName	
	OriginalLineName	
	RealTimeBlock.mode	
	RealTimeBlock.realTimeMultiplier	

#### 4.19.2 setVariableCustomizationFunctionHandle

フィルタ関数を通してすべての適合変数または測定変数は、このカスタムフックでカスタマイズすることができます。カスタマイズできる内容についての詳しい情報は、[「MScripts.VariableWrapper」](#)（ページ50）を参照してください。

## フック関数のシグネチャ

関数 `variableCustomization(type, variableInfoWrapper)`

パラメータ	<code>type char[]</code>	以下のいずれかの定数を使用できます。
メータ		<code>MScripts.DataModelBase.SimulinkWorkspaceCalibration</code>
		<code>MScripts.DataModelBase.DLLCalibrationAxis</code>
		<code>MScripts.DataModelBase.DLLCalibration</code>
		<code>MScripts.DataModelBase.DLLMeasurement</code>
		<code>MScripts.DataModelBase.SimulinkCalibrationAxis</code>
		<code>MScripts.DataModelBase.SimulinkCalibration</code>
		<code>MScripts.DataModelBase.SimulinkMeasurement</code>
variableInfoWrapper		<code>MScripts.VariableWrapper</code> のインスタンス variableInfoWrapperは、現在処理されている適合または測定を判定するために使用されます。 variableInfoWrapperの詳細な使用方法については、「 <a href="#">MScripts.VariableWrapper</a> 」(ページ50)を参照してください。

"SIPDEMO"フォルダ内のCustomHooks variableCustomization.m scriptに、このフックのサンプルが含まれています。

カスタムフックを使用する前の変数の選択

Name	Role
_1_D_Lookup_Table	Referenced
CalibrationToFilter	Referenced
LineToFilter	Referenced
OriginalCalibrationName	Referenced
OriginalLineName	Referenced
RealTimeBlock.mode	Referenced
RealTimeBlock.realTimeMultiplier	Referenced

カスタムフックを使用した後の変数の選択

Name [Display ID]	Role
_1_D_Lookup_Table.BreakpointsForDimension1	Referenced
_1_D_Lookup_Table_sDisplayIdentifier	Referenced
CalibrationToFilter_sDisplayIdentifier	Referenced
LineToFilter_sDisplayIdentifier	Referenced
OriginalCalibrationName_sDisplayIdentifier	Referenced
OriginalLineName_sDisplayIdentifier	Referenced
RealTimeBlock_sDisplayIdentifier	Referenced
RealTimeBlock_sDisplayIdentifier	Referenced

### 4.19.3 setVariableRenamingFunctionHandle

フィルタ関数を通過するすべての適合変数または測定変数は、このカスタムフックで名前を変更することができます。show workspace/data dictionaries variablesで追加された適合変数は名前を変更できないため、このフックには渡されません。

#### フック関数のシグネチャ

---

関数	mappedName = variableNameMapper(incaSIPName, type, variableInfoWrapper)	
----	---	--

---

パラ	mappedName	新しいマッピング名。空にすることはできません。
メー	incaSIPName	INCA-SIPが提案する名前。
タ	type char[]	以下のいずれかの定数を使用できます。 MScripts.DataModelBase.DLLCalibrationAxis MScripts.DataModelBase.DLLCalibration MScripts.DataModelBase.DLLMeasurement MScripts.DataModelBase.SimulinkCalibrationAxis MScripts.DataModelBase.SimulinkCalibration MScripts.DataModelBase.SimulinkMeasurement
	variableInfoWrapper	MScripts.VariableWrapperのインスタンス variableInfoWrapperは、現在処理されている適合または測定を判定するために使用されます。 variableInfoWrapperの詳細な使用方法については、 <a href="#">「MScripts.VariableWrapper」</a> （ページ50）を参照してください。





---

### 注記

- 要求された名前がASAP2に準拠していない場合は、INCA-SIPがそれを修正して準拠させます。
- 要求された名前がすでに使用されている場合は、INCA-SIPは\_xを付加します (xは1~infです)。
- incaSIPNameは推奨される名前で、ここでは一意でない名前やASAP2非準拠の名前にすることもできます。
- このフックにおいては、ゲッターのみをvariableInfoWrapperで使用してください。この適合変数または測定変数に対する変更内容は、setVariableCustomizationFunctionHandleフック内で設定することができます。

"SIPDEMO"フォルダ内のCustomHooks variableNameMapper.m scriptに、このフックのサンプルが含まれています。

カスタムフックを使用する前の変数の選択

All Sources (Not Filtered, 7/7 Visible)		
	Name	Role
	_1_D_Lookup_Table	
	CalibrationToFilter	
	LineToFilter	
	OriginalCalibrationName	
	OriginalLineName	
	RealTimeBlock.mode	
	RealTimeBlock.realTimeMultiplier	

カスタムフックを使用した後の変数の選択

CustomHooks (Not Filtered, 7/7 Visible)		
	Name [Display ID]	Role
	_1_D_Lookup_Table	Referenced
	CalibrationToFilter	Referenced
	CustomCalibrationName	Referenced
	CustomLineName	Referenced
	LineToFilter	Referenced
	RealTimeBlock.mode	
	RealTimeBlock.realTimeMultiplier	

#### 4.19.4 setGroupRenamingFunctionHandle

フィルタ関数を通してすべての適合変数または測定変数の各グループパスは、このカスタムフックで名前を変更することができます。一意の各パスが、フックに1回だけ渡されます。



## フック関数のシグネチャ

関数	<code>mappedName = groupNameMapper(incaSIPName, type, metadata)</code>	
パラ	<code>mappedName</code>	新しいマッピング名。空にすることはできません。
メータ	<code>incaSIPName</code>	INCA-SIPが提案する名前。
	<code>type char[]</code>	以下のいずれかの定数を使用できます。 <code>MScripts.DataModelBase.SimulinkGroup</code> <code>MScripts.DataModelBase.DLLGroup</code>
	<code>metadata</code>	グループのフルパス (例: <code>/model/subsystem/subsubsystem</code> )。 <code>metadata</code> を用いて現在処理されているグループが判定されま す。

 注記

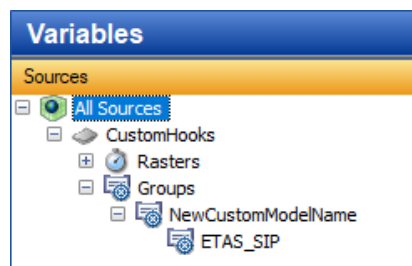
- － 要求された名前がASAP2に準拠していない場合は、INCA-SIPがそれを修正して準拠させます。
- － 要求された名前がすでに使用されている場合は、INCA-SIPは\_xを付加します (xは1~infです)。
- － `incaSIPName`は推奨される名前で、ここでは一意でない名前やASAP2非準拠の名前にすることもできます。

"SIPDEMO"フォルダ内のCustomHooks groupNameMapper.m scriptに、このフックのサンプルが含まれています。

カスタムフックを使用する前の変数の選択



カスタムフックを使用した後の変数の選択



### 4.19.5 MScripts.VariableWrapper

このクラスは、適合変数と測定変数をラップし、そのプロパティにアクセスしてその内容を変更する際に使用されます。

#### 変更できるプロパティと機能

プロパティ	sDisplayIdentifier
説明	A2LディスプレイIDのエントリ
変数タイプ別	MScripts.DataModelBase.DLLCalibration
の使用法	MScripts.DataModelBase.DLLMeasurement
	MScripts.DataModelBase.SimulinkCalibration
	MScripts.DataModelBase.SimulinkMeasurement
	MScripts.DataModelBase.SimulinkWorkspaceCalibration
フィールドタイプ	char[]
プロパティ	dMin
説明	A2L変数の最小値。INCAにより使用されます。
変数タイプ別	MScripts.DataModelBase.DLLCalibrationAxis
の使用法	MScripts.DataModelBase.DLLCalibration
	MScripts.DataModelBase.DLLMeasurement
	MScripts.DataModelBase.SimulinkCalibrationAxis
	MScripts.DataModelBase.SimulinkCalibration
	MScripts.DataModelBase.SimulinkMeasurement
	MScripts.DataModelBase.SimulinkWorkspaceCalibration
フィールドタイプ	number

プロパティ	dMax
説明	A2L変数の最大値。INCAにより使用されます。
変数タイプ別の使用方法	MScripts.DataModelBase.DLLCalibrationAxis MScripts.DataModelBase.DLLCalibration MScripts.DataModelBase.DLLMeasurement MScripts.DataModelBase.SimulinkCalibrationAxis MScripts.DataModelBase.SimulinkCalibration MScripts.DataModelBase.SimulinkMeasurement MScripts.DataModelBase.SimulinkWorkspaceCalibration
フィールドタイプ	number
プロパティ	sDescription
説明	A2L変数のディスクリプションエントリ。
変数タイプ別の使用方法	MScripts.DataModelBase.DLLCalibrationAxis MScripts.DataModelBase.DLLCalibration MScripts.DataModelBase.DLLMeasurement MScripts.DataModelBase.SimulinkCalibrationAxis MScripts.DataModelBase.SimulinkCalibration MScripts.DataModelBase.SimulinkMeasurement MScripts.DataModelBase.SimulinkWorkspaceCalibration
フィールドタイプ	char[]

プロパティ	numOfDimensions
説明	変数が表すデータの次元数。
変数タイプ別の使用方法	MScripts.DataModelBase.DLLCalibrationAxis MScripts.DataModelBase.DLLCalibration MScripts.DataModelBase.DLLMeasurement MScripts.DataModelBase.SimulinkCalibrationAxis MScripts.DataModelBase.SimulinkCalibration MScripts.DataModelBase.SimulinkMeasurement MScripts.DataModelBase.SimulinkWorkspaceCalibration
フィールドタイプ	number
プロパティ	dataAccessInfo
説明	この変数ラッパーが表す適合変数または測定変数を決定します。
変数タイプ別の使用方法	MScripts.DataModelBase.DLLCalibrationAxis MScripts.DataModelBase.DLLCalibration MScripts.DataModelBase.DLLMeasurement MScripts.DataModelBase.SimulinkCalibrationAxis MScripts.DataModelBase.SimulinkCalibration MScripts.DataModelBase.SimulinkMeasurement MScripts.DataModelBase.SimulinkWorkspaceCalibration
フィールドタイプ	MScripts.AccessInfo

関数	addGroupPath (<char[] newPath>)	
説明	変数用のA2Lグループを追加します。	
変数タイプ別の使用方法	MScripts.DataModelBase.DLLCalibrationAxis MScripts.DataModelBase.DLLCalibration MScripts.DataModelBase.DLLMeasurement MScripts.DataModelBase.SimulinkCalibrationAxis MScripts.DataModelBase.SimulinkCalibration MScripts.DataModelBase.SimulinkMeasurement MScripts.DataModelBase.SimulinkWorkspaceCalibration	
フィールドタイプ	MScripts.AccessInfo	
パラメータ	newPath	この変数を含める新しいA2Lグループ。このパスは、Simulink®モデル内に存在している必要はありません。どのようなパスでも使用できます。
関数	makeComAxis	
説明	A2Lの軸を強制的にCOM軸にします。	
変数タイプ別の使用方法	MScripts.DataModelBase.DLLCalibrationAxis MScripts.DataModelBase.SimulinkCalibrationAxis	
フィールドタイプ	MScripts.AccessInfo	

プロパティ	<code>xAxisAccessInfo</code>
説明	この変数エレメントが表す適合軸を決定します。
変数タイプ別 の使用方法	<code>numOfDimensions &gt;= 1</code> の場合は、以下の変数タイプで使用できます。 <code>MScripts.DataModelBase.DLLCalibration</code> <code>MScripts.DataModelBase.SimulinkCalibration</code> <code>MScripts.DataModelBase.SimulinkWorkspaceCalibration</code>
フィールドタイプ	<code>MScripts.AccessInfo</code>
プロパティ	<code>yAxisAccessInfo</code>
説明	この変数エレメントが表す適合軸を決定します。
変数タイプ別 の使用方法	<code>numOfDimensions &gt;= 2</code> の場合は、以下の変数タイプで使用できます。 <code>MScripts.DataModelBase.DLLCalibration</code> <code>MScripts.DataModelBase.SimulinkCalibration</code> <code>MScripts.DataModelBase.SimulinkWorkspaceCalibration</code>
フィールドタイプ	<code>MScripts.AccessInfo</code>
プロパティ	<code>zAxisAccessInfo</code>
説明	この変数エレメントが表す適合軸を決定します。
変数タイプ別 の使用方法	<code>numOfDimensions &gt;= 3</code> の場合は、以下の変数タイプで使用できます。 <code>MScripts.DataModelBase.DLLCalibration</code> <code>MScripts.DataModelBase.SimulinkCalibration</code> <code>MScripts.DataModelBase.SimulinkWorkspaceCalibration</code>
フィールドタイプ	<code>MScripts.AccessInfo</code>

プロパティ	<code>wAxisAccessInfo</code>
説明	この変数エレメントが表す適合軸を決定します。
変数タイプ別の使用方法	<code>numOfDimensions = 4</code> の場合は、以下の変数タイプで使用できます。 <code>MScripts.DataModelBase.DLLCalibration</code> <code>MScripts.DataModelBase.SimulinkCalibration</code> <code>MScripts.DataModelBase.SimulinkWorkspaceCalibration</code>
フィールドタイプ	<code>MScripts.AccessInfo</code>

### 注記

すべての`axisAccessInfo`について、そのインデックスの軸が編集不可能な軸（固定軸）である場合にも例外がスローされることがあります。

- 間違っただ変数タイプで呼び出しが行われると、例外がスローされます。
- この変数ラッパーが表す適合変数または測定変数を決定するには、`MScripts.AccessInfo`を使用します。

## 4.19.6 MScripts.AccessInfo

このクラスには、どの適合変数または測定変数が表されているかを判定するための2つのパラメータが含まれています。これらのパラメータは、`accessParameter1`と`accessParameter2`です。記述をわかりやすくするため、変数のタイプに応じたゲッターを使用することができます。

変数のタイプ	<code>MScripts.DataModelBase.SimulinkCalibrationAxis</code> <code>MScripts.DataModelBase.SimulinkCalibration</code>
使用できるゲッター	<code>getBlockPath</code> ブロックパスは、ブロックが配置されているモデル内のパスです。 <code>getPropertyName</code> プロパティ名は、このキャリブレーションが表すブロック内のプロパティです。
注記	アクセス情報を使用してブロックの名前を取得するヘルパー関数： <pre>blockName = MScripts.AccessInfo.getBlockName(&lt;hAccessInfo&gt;);</pre>

---

変数 `MScripts.DataModelBase.SimulinkWorkspaceCalibration`  
のタイプ

---

使用 `getBlockPath` ブロックパスモデル名  
でき

---

るゲッター `getPropertyName` プロパティ名はワークスペース変数の名前です。

---



---

変数 `MScripts.DataModelBase.SimulinkMeasurement`  
のタイプ

---

使用 `getBlockPath` ブロックパスは、ブロックが配置されているモデル内のパスです。  
でき

---

るゲッター `getPortNumber` ポート番号は、ブロックの出力ポートの行のインデックスです。

---

注記 アクセス情報を使用してブロックの名前を取得するヘルパー関数：  
`LineName = MScripts.AccessInfo.getLineName`  
`(<hAccessInfo>);`

---



---

変数 `MScripts.DataModelBase.DLLCalibrationAxis`  
のタイプ  
`MScripts.DataModelBase.DLLCalibration`  
`MScripts.DataModelBase.DLLMeasurement`

---

使用 `getDllName` DLL名は、DLLモードで使用されているDLLの名前です。

---

できるゲッター `getModuleOffset` モジュールオフセットは、DLL内の変数の位置を表すメモリオフセットです。詳細は、「[DLLモード](#)」(ページ30)を参照してください。

---

## 4.20 サポートされているSimulink<sup>®</sup>ブロックのタイプ

INCA-SIPは以下のタイプのSimulink<sup>®</sup>ブロックをサポートしています。

ブロックタイプ	Constant
ブロックタイプ	Gain
ブロックタイプ	Lookup
ブロックタイプ	Lookup2D
ブロックタイプ	Lookup3D



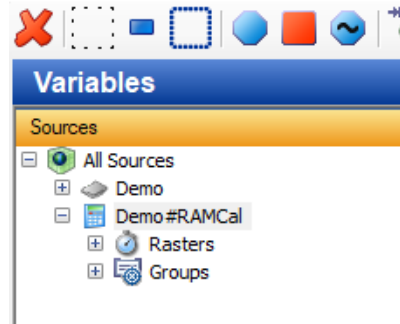
ブロックタイプ	sfun_lookupnd
ブロックタイプ	Stateflow
必須条件	パラメータ - Workspace Variables
ブロックタイプ	LookupNDDirect
必須条件	NumberOfTableDimensions → 1~4
ブロックタイプ	S-FunctionとM-S-Function
必須条件	Tunable → on
	Visible → on
	Enabled → on
	ReadOnly → off
ブロックタイプ	Interpolation_n-DとS-Function (FunctionName → sfun_kflookupnd)
必須条件	TableSource → Dialog
	NumberOfTableDimensions/numDimsPopupSelect → 1~4
ブロックタイプ	ManualSwitch
ブロックタイプ	PreLookup DとS-Function (FunctionName → sfun_idxsearch)
ブロックタイプ	Lookup_n-D
必須条件	NumberOfTableDimensions → 1~4

## 4.21 制限事項とその他の挙動

本項には、INCA-SIPの制限事項と挙動に関するヒントと情報が含まれています。

### 4.21.1 RAM適合

INCA-SIPはRAM適合をサポートしていません。RAM適合は、測定変数の適合を行うことができる機能です。この機能の有効化／無効化は、INCAユーザーオプションで設定します。この機能により生成された適合変数は、**変数の選択** ダイアログボックスに以下のように表示されます。



### 4.21.2 他のハードウェアとの同時測定

INCA-SIPを使用する時には、INCAに接続されたデバイスを用いた測定／適合は行えません。

### 4.21.3 測定の一時停止

INCAにおいて記録の一時停止を行った場合は、記録された測定ファイルをMDAで分析すると、データが存在しない、または補間されたデータを含む一時停止期間が表示されます。実際のハードウェアによる測定の場合は、一時停止の期間は1秒のギャップとして表示されます。

## 5 チュートリアル

このチュートリアルは、INCA-SIPの一連の操作方法を簡単にご紹介するものです。ここでは、以下のディレクトリに保存されている"SIPDemo"というモデルを使用します。

<drive>:\ETASDATA\INCA7.5\Demo\AddOn\_INCA-SIP7.5.

このチュートリアルでは、Simulink®モデルをINCAに接続し、実験を実行して、Simulink®モデルに変更を加えます。

ここでは以下のアクションを行います。

- 「INCA-SIPをMATLAB®に追加する」(下記)
- 「INCA-SIPをINCAに接続する」(下記)
- 「INCAでモデルの適合/測定を行う」(下記)
- 「エミュレーションモードを変更する」(次ページ)

### INCA-SIPをMATLAB®に追加する



あるバージョンのMATLAB®を初めてINCA-SIPと併用するには、INCA-SIPをMATLAB®パスに追加する必要があります。詳細は、「INCA-SIPのインストールパス」(ページ14)を参照してください。

1. MATLAB®を開きます。
2. "SIPDEMO"フォルダにナビゲートします。
3. **sipdemo.mdl** をダブルクリックしてSimulink®モデルを開きます。  
Simulink®エディタが開き、モデルが表示されます。


### INCA-SIPをINCAに接続する

1. MATLAB®の**Tools**メニューで、**INCA-SIP > Connect to INCA**を選択します。  
"ETAS SIP"ブロックがモデルに追加されます。INCAが起動し、"INCASIP"データベース内のフォルダに、ワークスペースとECUプロジェクトが作成されます。


### INCAでモデルの適合/測定を行う

1. INCAデータベースマネージャの"データベースアイテム"フィールドで、INCASIP/sipdemoフォルダを開きます。
2. ワークスペース  "sipdemo\_wsp"をダブルクリックします。  
INCAの実験環境が開きます。
3. **変数 > 変数の選択** を選択します。  
または  
 をクリックするか、**SHIFT + F4**を押します。  
変数の選択ダイアログボックスが開きます。
4. 変数"SineWaveOutput"を選択します。
5. 変数を右クリックし、ショートカットメニューから **任意のウィンドウに追加 > レイヤ\_1 > 新しいウィンドウ > YTオシロスコープ** を選択します。
6. 変数"SineWaveAmplitude"を選択します。
7. **OK** をクリックします。

適合変数と測定変数が新しいYTオシロスコープに追加されます。

8.  をクリック、または**F11**キーを押して、実験を開始します。


Simulink<sup>®</sup>でのシミュレーションとINCAでの測定が開始され、測定値がYTオシロスコープに表示されます。

9.  をクリック、または**F9**キーを押して、実験を終了します。

### 注記

INCAの実験において"SineWaveAmplitude"の値を変更すると、オシロスコープにその効果が反映されます。

### エミュレーションモードを変更する

- INCAの実験環境で、**変数 > 変数の選択** を選択、または  をクリックするか、**SHIFT + F4**を押します。  
変数の選択ダイアログボックスが開きます。
- 適合変数"RealTimeBlock.mode"と"RealTimeBlock.realTimeMultiplier"を選択します。  
スカラー適合変数用のデフォルトエディタ（基本適合エディタまたはスカラーテーブルエディタ）が開きます。デフォルトエディタは、INCAユーザーオプションで選択できます。
- "RealTimeBlock.realTimeMultiplier"の値を4に変更します。  
シミュレーションが実時間の4倍の速度で実行されます。  
または  
"RealTimeBlock.realTimeMultiplier"の値を0.5に変更します。  
シミュレーションが実時間より遅い速度で実行されます。  
または  
"RealTimeBlock.mode"の値をFast Emulation Modeに変更します。  
シミュレーションが最高レベルの速度で実行されます。

## 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](http://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](http://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 :	<a href="http://www.etas.com">www.etas.com</a>

## 目次

C	
C++コンパイラ	13
D	
DLL	
型	31
DLL検索スクリプト	32
DLLモード	30
E	
ECU	
ビジー状態	29
新規追加	18
ETAS	
お問い合わせ先	61
I	
INCA-SIP	
MATLABに追加	59
インストール	13
エミュレーションモード	27
挙動	58
制限事項	58
ブロック削減	28
M	
MATLAB	
INCA-SIPを追加	59
検索パス	16
シミュレーションモード	28
スクリプトファイル	14
適合の拒否	29
ブロック削減	28
MDF	24
MScripts.AccessInfo	55
MScripts.VariableWrapper	50
S	
setGroupRenamingFunctionHandle	48
setVariableCustomisationFunctionHandle	45
setVariableRenamingFunctionHandle	47
Simulink	
参照モデル	29
接続 INCAへ	9
い	
インストール	
INCA-SIP	13

インストールパス .....	14
え	
エミュレーションモード .....	27
高速 .....	28
変更方法 .....	60
リアルタイム .....	27
お	
お問い合わせ先 .....	61
オブジェクト	
生成 .....	34
オブジェクトの生成 .....	34
か	
カスタムフック .....	42
関数	
addGroupPath .....	39
createCalibrationAxis .....	35
createCalibrationAxisWith MeasurementKey .....	36
createFixedAxis .....	35
set2AsapMaxValue .....	38
setAsap2Description .....	38
setAsap2MinValue .....	38
setAxis .....	36
createArrayMeasurement .....	37
createScalarMeasurement .....	37
createScalarCalibration .....	34
createTableCalibration .....	34
き	
技術的概要 .....	10
接続 .....	11
測定変数の扱い .....	11
適合変数の扱い .....	12
同期 .....	12
規則	
命名 .....	27
基本的な処理 .....	10
く	
グループ .....	27
こ	
高速エミュレーションモード .....	28
さ	
作成	
DLL検索スクリプト .....	32
サポート	
Simulink.ParameterとAsap2.Parameter .....	30
Simulink.SignalとAsap2.Signal .....	30

参照モデルのサポート .....	29
し	
システム要件 .....	13
ソフトウェア .....	13
ハードウェア .....	13
実験	
開始 .....	29
終了 .....	29
シミュレーションモード .....	28
せ	
生成	
オブジェクト .....	34
適合 .....	34
接続	
SimulinkからINCA .....	9
そ	
測定	
一時停止 .....	58
他のハードウェアと同時に .....	58
モデル .....	10
測定／適合 モデル .....	59
測定変数と適合変数 .....	27
測定変数の扱い .....	11
ソフトウェア .....	13
た	
他のハードウェアとの同時測定 .....	58
ち	
チュートリアル .....	59
て	
ディレクトリ	
インストーラファイル .....	14
ディレクトリとファイル .....	14
データ .....	8
RAM適合 .....	58
適合 .....	34
RAM .....	58
拒否 .....	29
生成 .....	34
モデル .....	10
適合操作の拒否 .....	29
適合変数と測定変数 .....	27
適合変数の扱い .....	12
と	
同期 .....	12



登録	
C++コンパイラ .....	13
は	
ハードウェア .....	13
ひ	
ビジー状態 ECU .....	29
ふ	
ファイル	
MATLABスクリプト .....	14
ブロック削減 .....	28
ま	
マクロ .....	26
め	
命名	
規則 .....	27
メニューアイテム .....	16
Configuration (設定) .....	41
接続 INCAへ .....	16
設定 .....	22-23
も	
モード	
DLL .....	30
モデル	
測定 .....	10
適合 .....	10
モデルの適合/測定 .....	59
り	
リアルタイムエミュレーションモード .....	27