

## ETAS INCA V7.5

シリアル(X)ETK による適合のコンセプト —  
リミテッドエミュレーションRAM



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。ご利用および複写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、複写、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 V7.5 | ユーザーガイド R01 JP | 03.2024

## 本書の内容

1	はじめに .....	5
1.1	製品の正しい使用法 .....	5
1.2	対象ユーザー .....	5
1.3	安全に関する注意事項の書式 .....	6
1.4	製品を安全に使用するために .....	6
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	シリアルxETKを用いた適合について .....	8
2.1	システム要件 .....	8
2.2	用語と略語について .....	9
3	基本概念 .....	11
3.1	適合とは? .....	11
3.2	エミュレーションRAMとは? .....	11
3.3	静的エミュレーション（手動で実験準備を行う従来のコンセプト） .....	12
3.3.1	INCAによるエミュレーションRAMの自動設定 .....	13
3.3.2	オペレータによる実験準備作業 .....	14
3.3.3	エミュレーション領域切り替え時の適合データの書き戻し .....	15
3.4	ダイナミックエミュレーション（固定サイズのブロックを使用） .....	16
3.5	ダイナミックエミュレーション（再構成可能な可変サイズのブロックを使用） .....	18
3.6	両コンセプトの比較 .....	19
4	手動での実験準備 .....	21
4.1	手動で実験準備を行うエミュレーションのためのユーザーオプション設定 .....	21
4.2	ワークスペースの準備 .....	21
4.3	実験の実行 .....	22
4.3.1	初期化 .....	22
4.3.2	メモリページ操作の制限事項 .....	22
4.3.3	エミュレートする領域の設定 .....	23
4.3.4	エミュレーションRAMの割り当て状況の確認 .....	27
5	ダイナミックエミュレーション .....	29
5.1	ダイナミックエミュレーションのユーザーオプションの設定 .....	29
5.2	ワークスペースの準備 .....	30
5.3	実験の実行 .....	31
5.3.1	初期化 .....	31

5.3.2	メモリページ間の不一致の解決 .....	31
5.3.3	ダイナミックエミュレーションによる適合の実行 .....	33
5.3.4	エミュレーションRAMの割り当て状況の確認 .....	35
6	制限事項 .....	37
6.1	一般的な制限事項 .....	37
6.2	従来のコンセプト（実験開始時にエミュレーションRAMの設定を行う方法）の制限事項 .....	37
6.3	新しいコンセプト（ダイナミックエミュレーション）の制限事項 .....	37
7	A2Lファイルの記述例 .....	39
7.1	A2Lファイルの記述例 – FixedSizeMoveableEmuRAMを使用する場合 .....	39
7.2	A2Lファイルの記述例 – FixedSizeNonMoveableEmuRAMを使用する場合 .....	40
7.3	A2Lファイルの記述例 – FlashRangeSpecificEmuRAMを使用する場合 .....	41
7.4	A2Lファイルの記述例 – ReconfigurableSizeMoveableEmuRAMを使用する場合 .....	42
8	ご使用上のヒント .....	44
8	お問い合わせ先 .....	46

# 1 はじめに

## 1.1 製品の正しい使用法

リミテッドエミュレーションRAM ("Limited Emulation RAM") はINCA測定・適合ツールの機能のひとつです。マイクロコントローラ上の適合RAMを管理し、ワーキングページとリファレンスページを用いる「ページコンセプト」により適合変数（ROM上のパラメータ）を適合することができます。

本書は『INCA入門ガイド』を補足するものとして、エミュレーションRAMを使用する際に用いられるコンセプトを紹介し、さらに、INCAでそのコンセプトを用いて適合を行うための準備作業を概説します。

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。

ユーザーは、INCAの一般的な操作方法を習得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 製品を安全に使用するために



#### 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 シリアルxETKを用いた適合について

最近のマイクロコントローラはメモリ内蔵型が多くなり、それに伴いRAMエミュレーション機能を内蔵したものが多くなってきました。このようなコントローラではROM（フラッシュEPROM）のアドレス領域をブロック単位でRAMに「オーバーレイする」（重ね合わせて割り当てる）ことができます。オーバーレイさせるブロックはECUの制御レジスタで指定できます。シリアルxETKはこの「エミュレーション」メカニズムを用いて、これらのタイプのコントローラ上で「ワーキングページ」（以下「WP」とも記します）と「リファレンスページ」（以下「RP」とも記します）を実現します。

これらのECUは固定サイズのブロックを限られた数しか持っていないため、適合する変数を選択しておく必要があります。INCA V5.3までは、エミュレーションRAMは静的な方法でしか扱えませんでした。つまり、適合作業を始める前に準備作業が必要でした。

その扱いを簡略化するため、INCA V5.4においてメモリブロックの動的エミュレーションが導入されました。メモリブロックが残っている限り、適合変数をいくつでも実験に追加することができます。実際に値を変更していない変数についてはエミュレーションは行われません。値の変更操作を行った時点で、その変数が含まれるブロックが自動的にエミュレートされます。エミュレーションRAMの全ブロックがエミュレーション用に割り当てられてしまうと、現在エミュレートされていない変数はINCAによって書き込み禁止になります。ユーザーがINCAのワーキングページをECUのフラッシュROMに書き込むことにより、占有されていたエミュレーションメモリが解放され、適合作業を続行することが可能となります。

さらにINCA V7.1では、このダイナミックエミュレーション機能がさらに拡張され、エミュレーションRAM上の各ブロックを異なるサイズに再構成できるようになりました。これによってエミュレーションRAMを効率的に利用できるようになり、同じサイズのプロジェクトでより多くの変数を適合できるようになりました。

ダイナミックエミュレーションを用いたこれらの新しい手法は、操作が簡単で、特にテストベンチでの適合作業に役立ちます。従来の手法（静的エミュレーション）では、INCAでの実験開始前に変数選択ダイアログボックスが開き、手動操作で変数を選択する必要がありますが、この操作はテストベンチからのリモート操作においては弊害となります。

現行バージョンのINCAは新旧の手法をサポートしています。INCAユーザーオプションでいずれか一方（手動で実験準備を行う従来の手法、またはダイナミックエミュレーション）を選択します。ユーザーオプションでダイナミックエミュレーションが選択されている場合、A2Lファイルの定義内容（再構成可能な可変サイズのエミュレーションRAMの定義の有無）とxETKドライバのバージョンに応じて、メモリブロックの使用方法（固定サイズのブロックを使用、または再構成可能な可変サイズのブロックを使用）が決定されます。

### 2.1 システム要件

リミテッドエミュレーションRAM ("Limited Emulation RAM") の機能は、INCAと共にインストールされます。INCAのシステム要件とインストール方法は、『INCAインストールガイド』を参照してください。

リミテッドエミュレーションRAMの機能を使用するには、最新バージョンのHSP（ハードウェアサービスパッケージ）でインターフェースハードウェアをアップデートする必要があります。最新のHSPはETASホームページのダウンロードセンター

（[http://www.etas.com/en/products/download\\_center.php](http://www.etas.com/en/products/download_center.php)）からダウンロードできます。



INCAでサポートされている可変エミュレーションRAMを利用した適合メソッドのどれが実際に使用できるかは、xETKデバイスのタイプとドライババージョンに応じて異なります。また、xETKデバイスのタイプとドライババージョンによってはこの機能がサポートされていない場合があります。ドライバのアップデートについては、ETASのサポート窓口までお問い合わせください。

## 2.2 用語と略語について

### 用語の定義

#### エミュレーション

パラメータの固定値を可変値に置き換えるメカニズム

#### エミュレーションRAM

適合変数（パラメータ）の値をエミュレートするため、FLASHメモリに置き換えて（=オーバーレイして）使用される専用のRAM

#### ダイナミックエミュレーション

エミュレーションモードのひとつ。適合変数が、必要に応じて動的にエミュレーションRAMに割り当てられます。

#### 手動操作で実験準備を行うエミュレーション

エミュレーションモードのひとつ。実験開始時に、手動でエミュレートする適合変数（パラメータ）のセットを指定し、エミュレーションRAMに固定的に割り当てます。

#### プロジェクトのフルエミュレーション

エミュレーションモードのひとつ。エミュレーションRAMの容量が、A2Lファイルに定義されたメモリセグメントDATAおよびOFFLINE\_DATAに含まれるすべての適合変数をカバーできる場合は、このモードが使用されます。

#### 実験のフルエミュレーション

エミュレーションモードのひとつ。実験に割り当てられた適合変数をすべて同時にエミュレーションRAMに割り当てるモードです。INCAは、実験開始時にこの状態を確立しようとします。適合変数が追加または削除されても、エミュレーションRAMは更新されません。

### 略語

<b>ECU</b>	Electronic Control Unit : 電子制御ユニット
<b>HDC</b>	Hardware Definition Code : ハードウェア定義コード (xETKファームウェア)
<b>HSP (ハードウェアサービスパック)</b>	Hardware Service Pack : ハードウェアサービスパック (ETASハードウェア用ファームウェアパッケージ)
<b>RAM</b>	Random Access Memory : ランダムアクセスメモリ

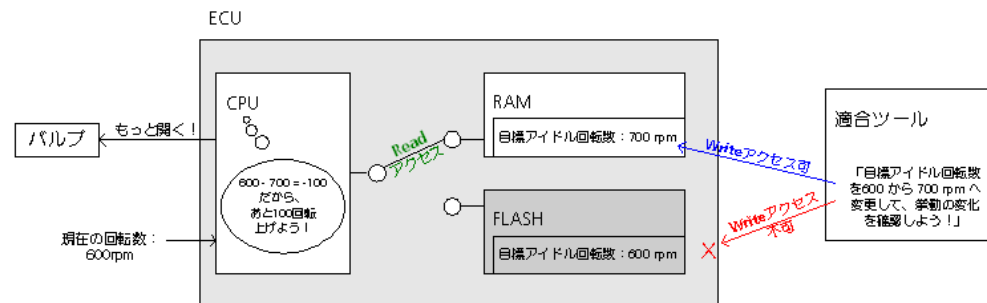
<b>ROM</b>	Read Only Memory : 読み出し専用メモリ (本書ではフラッシュ EPROMを指します)
<b>RP</b>	Reference Page : リファレンスページ
<b>WP</b>	Working Page : ワーキングページ

## 3 基本概念

INCAのリミテッドエミュレーションRAM ("Limited Emulation RAM") の機能を利用すると、ECUのマイクロコントローラ上の**適合RAM**を管理し、ワーキングページとリファレンスページを用いる「ページコンセプト」により適合変数（ROM上のパラメータ）を適合することができます。本章では、「適合」、「エミュレーションRAM」、および「リミテッドエミュレーションRAMでのINCAの管理」について説明します。

### 3.1 適合とは？

適合とは、適合変数の値を、ECUプログラム実行中にINCAからオンラインでリアルタイムに変更し、ECUのシステム特性を理想の状態に近づけることを意味します。これを実現するためには、適合変数が含まれるデータ領域を擬似的にRAM上に配置し、値を変更できるようにする必要があります。



適合時において、適合変数は以下の2種類のいずれかのメモリ上に配置されることになります。

- RAM (Random Access Memory)

読み書き可能なメモリです。

- フラッシュEPROM (本書では「フラッシュメモリ」、「ROM」とも記します)

電氣的にメモリの消去を行えるROMです。消去はブロック単位、またはメモリ全体でのみ行え、消去後は再度電氣的に書き込む (=プログラムする) ことが可能です。

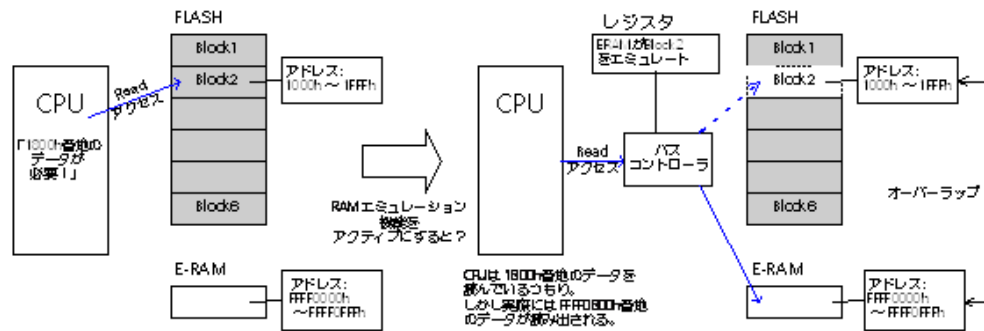
フラッシュEPROMの書き換えには専用のプログラミングツールを使用します。フラッシュ技術の利点は、車両に搭載されたECUであっても、プログラミングツールを使ってフラッシュメモリを書き換えることが可能な点にあります。一般に、ECUメモリには大量のデータを保存する必要があり、かつ製品のライフサイクルにおいてその内容を書き換える必要が生じる可能性があるため、ECUにとってこのテクノロジーは必須のものであるといえます。

### 3.2 エミュレーションRAMとは？

エミュレーションRAMは、そのアドレスをフラッシュメモリのアドレス領域に擬似的にオーバーレイさせる (=重複させる) ことにより、フラッシュメモリに配置された適合変数の値を変更することを可能とするものです。

昨今では多くのマイクロコントローラに内部メモリが内蔵されています。これらのマイクロコントローラの多くには**エミュレーションRAM**が搭載されており、これを用いてxETKがフラッシュメモリ上の値をエミュレートします。

以下に、エミュレーションRAM (図中では「E-RAM」) の役割について簡単に説明します。



通常時においては、ECUの制御プログラムはフラッシュメモリ上のパラメータを読み込んで動作します。しかしエミュレーションRAM機能を有効にすると、内蔵RAMの特定の領域がエミュレーションRAMとして使用され、この領域がフラッシュメモリの一部をブロック単位でエミュレートします。エミュレーションRAMの有効化/無効化は、ECU内の制御レジスタによって制御されます。

一般的なマイクロコントローラでは、エミュレーションRAMは複数のブロックで構成されており、フラッシュメモリ上の複数のブロックを同時にエミュレートすることが可能です。

エミュレーションRAMのブロック数と各ブロックのサイズ、エミュレートできるフラッシュメモリの領域などは、マイクロコントローラのタイプに応じて異なりますので、各マイクロコントローラの説明書をご参照ください。また、各xETKがサポートしているマイクロコントローラのタイプについては、ETASのホームページ (<http://www.etas.com>) でxETKの製品情報のページをご参照ください。

INCAで適合を行う際は、内蔵フラッシュメモリがリファレンスページ（「RP」）として使用され、内蔵エミュレーションRAMがワーキングページ（「WP」）として使用されます。このような割り当て設定を行うために、INCAに接続されたxETKは、マイクロコントローラのデバッグポートを経由してエミュレーションRAMの制御レジスタをセットします。制御レジスタの数と使用法は、マイクロコントローラによって異なります。

### 3.3 静的エミュレーション（手動で実験準備を行う従来のコンセプト）

実験の開始時、またはハードウェアを明示的に初期化する際、INCAは、すべての適合変数をエミュレーションRAM上でエミュレートできるかどうかを確認します。この確認処理は、以下のステップで行われます。

1. 実験に割り当てられているすべての適合変数を一度にエミュレートできるかどうかを調べ、可能な場合は、エミュレーションRAMの割り当てを設定します。自動的に実行できる処理はすべてユーザーの介入なしに実行されます。
2. エミュレーションRAMのサイズが足りない場合、INCAはオペレータに対して適合する変数を減らすことを要求します。このステップは、指定された適合変数をすべてエミュレートできるようになるまで繰り返されます。
3. INCAがECUに対して必要な設定を行います。

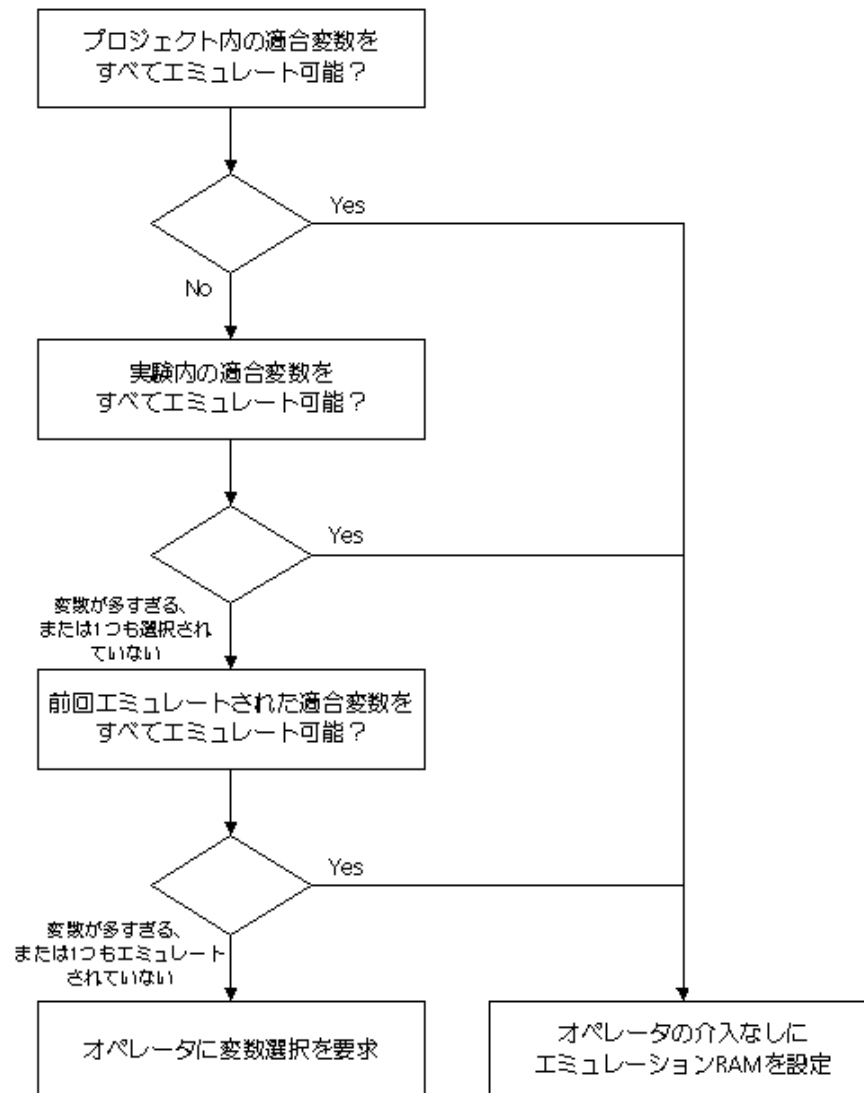
上記の実験準備作業が終わると、通常どおりに実験を行うことができます。

**i 注記**

エミュレーションRAMを使用する際に手動で実験準備を行うようにするには、A2Lファイル内にCALIBRATION\_METHOD（適合メソッド）としてFixedSizeMoveableEmuRAMが定義されている必要があります。詳しくは、「[A2Lファイルの記述例](#)」（ページ39）を参照してください。また、xETKデバイスのタイプとドライババージョンによってはこの機能がサポートされていない場合があります。ドライバのアップデートについては、ETASのサポート窓口までお問い合わせください。

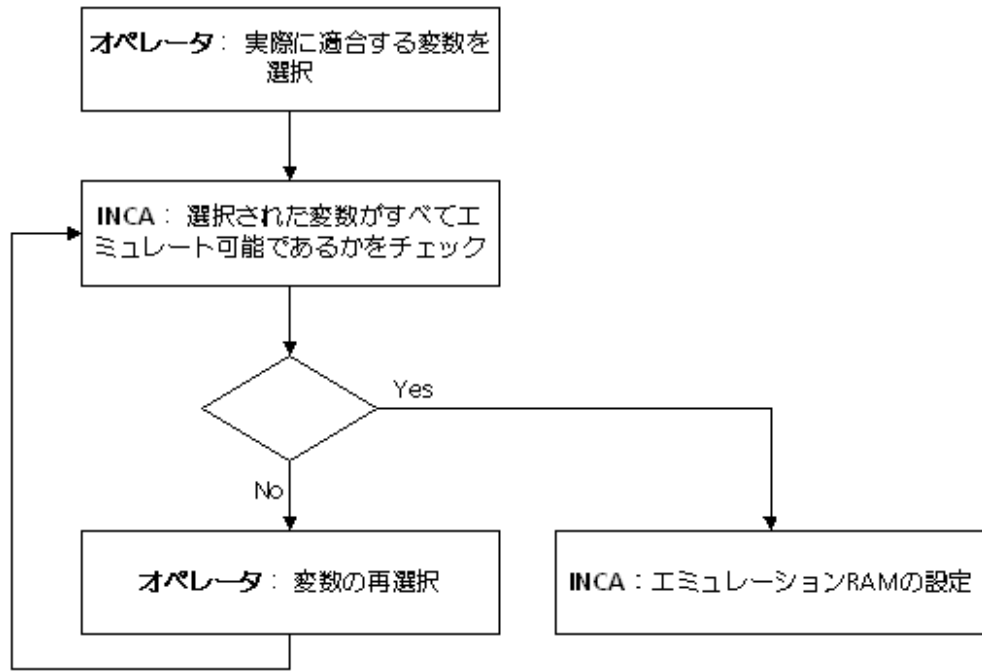
### 3.3.1 INCAによるエミュレーションRAMの自動設定

INCAのオペレータが実験開始またはハードウェア初期化の操作を行います。その際にINCAは、エミュレーションRAM構成の自動設定を試みます。ここでは以下のフローに従って処理が行われます。



### 3.3.2 オペレータによる実験準備作業

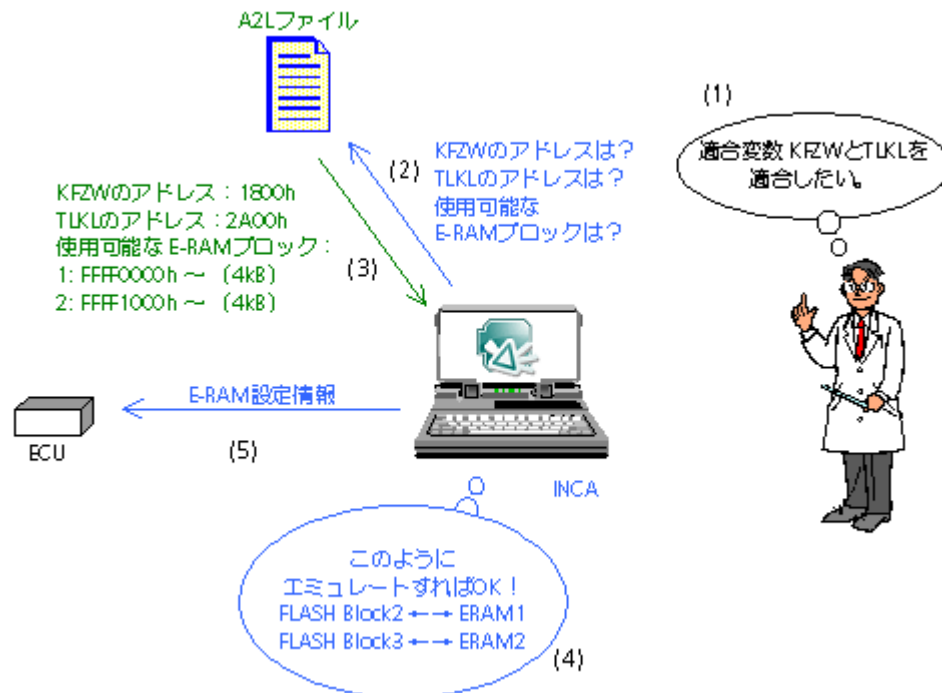
適合する変数を含むフラッシュメモリ領域のサイズがエミュレーションRAM領域より大きい場合は、オペレータが選択し変数をすべて一度にエミュレーションすることはできません。そのような場合、INCAは警告メッセージを表示し、ECUに適切な制御情報をセットできるように、オペレータに対して変数の数を減らすことを要求します。



#### ケース1： 選択された変数をすべてエミュレートできる場合

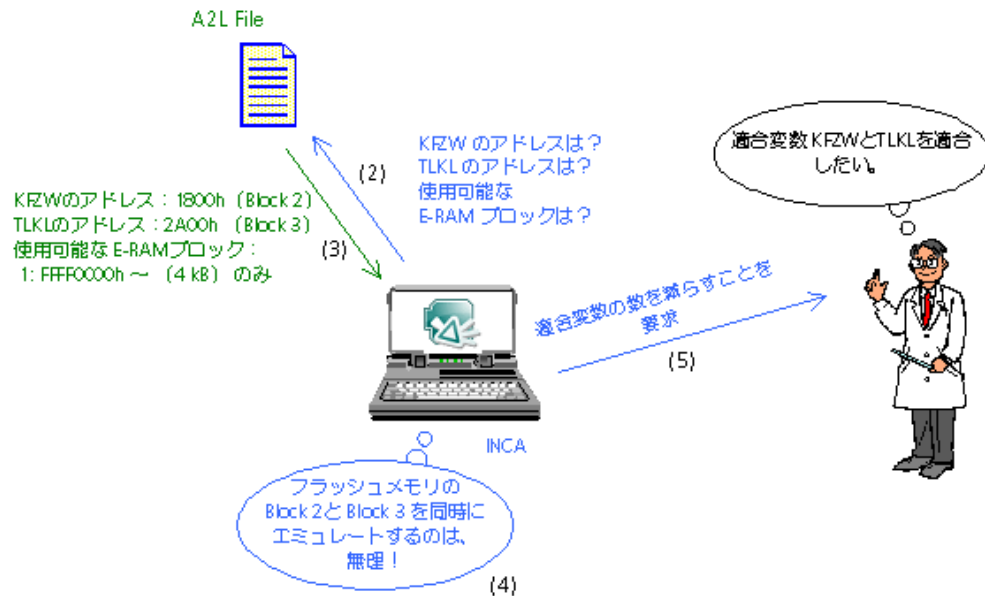
INCAのオペレータがいくつかの適合変数を含む実験を開きます。リミテッドエミュレーションRAM機能は、これらの変数のアドレスと、エミュレーションRAM情報をASAM-2MCファイルから読み取り、ECUのマイクロコントローラの制御レジスタに適切な値をセットします。

プロジェクトに含まれる適合変数をすべて一度にエミュレートできるかを調べる最初の自動設定のステップは、以下のイラストには含まれていません。



A2Lファイルに定義するブロックサイズとアドレスについての情報は、「A2Lファイルの記述例」(ページ39)で説明しています。

ケース 2：エミュレーションRAMのサイズ不足により適合変数の数を調整する必要がある場合

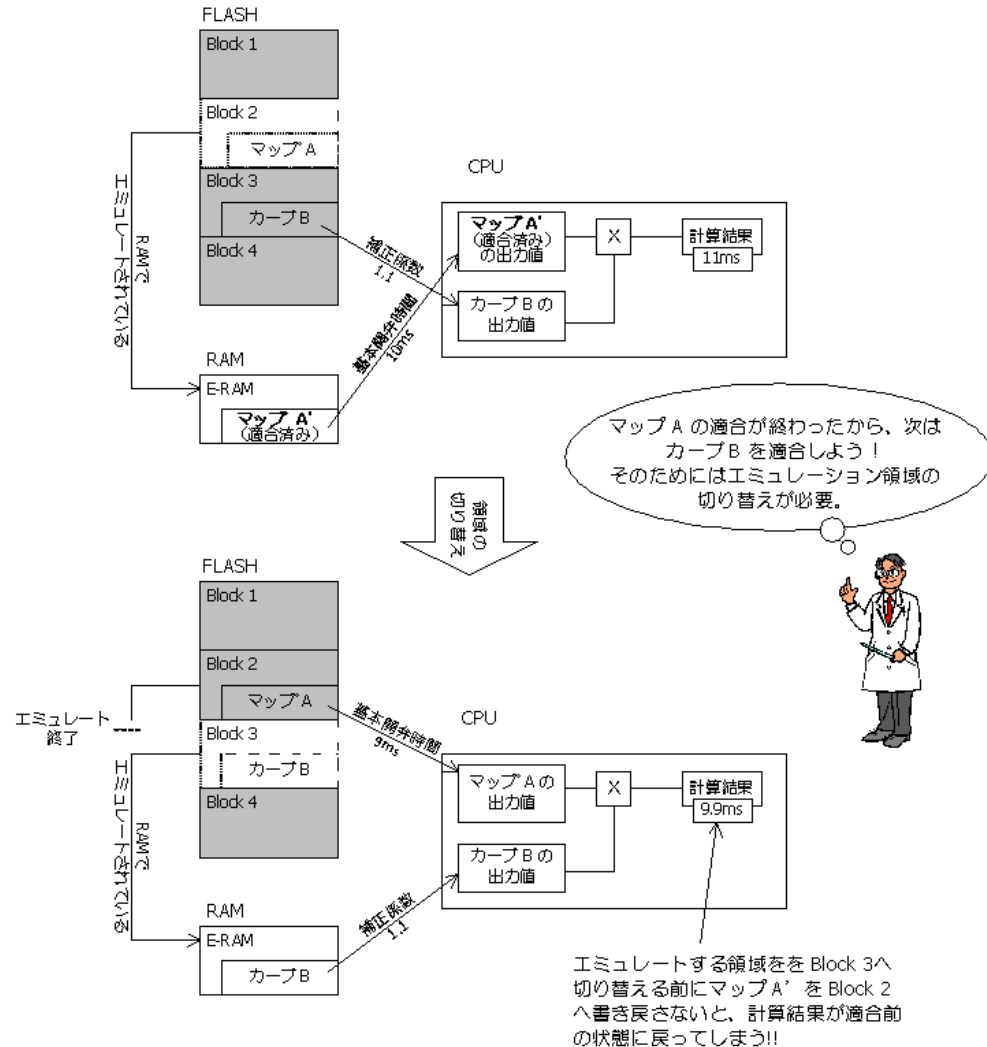


### 3.3.3 エミュレーション領域切り替え時の適合データの書き戻し

エミュレーションするフラッシュメモリの領域を変更して別のブロックの適合を行う場合、新たに選択されたブロックのデータがエミュレーションRAMにコピーされるため、エミュレーションRAM上で適合さ

れたデータはすべて失われてしまいます。適合したデータを残しておき、その後の適合作業にも継続して使用したい場合は、ブロックの切り替えを行う前に、現在エミュレートされているデータをフラッシュメモリに書き戻す必要があります。

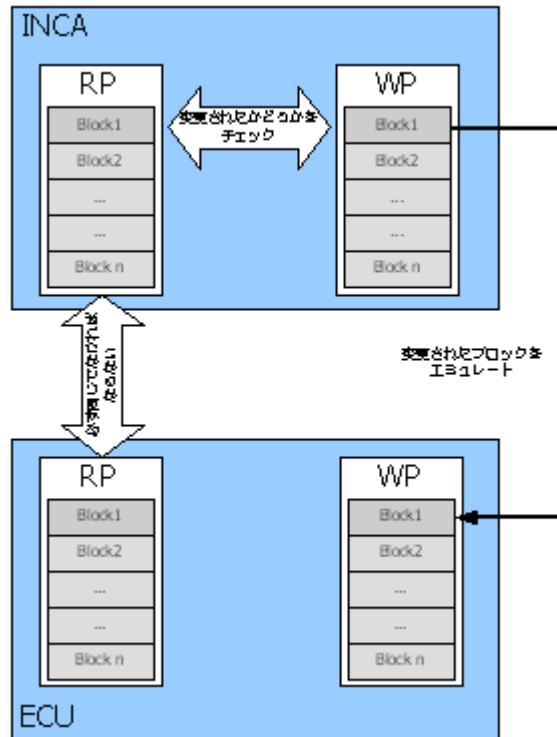
このような場合、INCAは以下の図に示すようにオペレータに対してデータのフラッシュ書き込みを行うことを要求します。



### 3.4 ダイナミックエミュレーション（固定サイズのブロックを使用）

固定サイズのブロックを使用するダイナミックエミュレーションのコンセプトにおいては、ユーザーが実験を開くと、INCAはPC上のリファレンスページとワーキングページの値を比較します。続いてA2Lファイルに定義されたエミュレーションRAM情報に従って固定サイズの一連のメモリブロックを算出して初期設定します。それらのサイズは、A2Lファイルで定義された対応するサイズと一致し、変更できません。エミュレーションは、INCAで値が変更された変数（つまりリファレンスページと異なる値がワーキングページにセットされた変数）についてのみ行われます。その後も、変数の値が変更されるたびに初期設定されたブロック内のいずれか（1つまたは複数）を使用して自動的にエミュレーションを行います。





### i 注記

この機能を使用するには、A2Lファイル内にCALIBRATION\_METHOD（適合メソッド）としてFixedSizeMovableEmuRAMが定義されている必要があります。  
 詳しくは、「[A2Lファイルの記述例](#)」（ページ39）を参照してください。  
 また、xETKデバイスのタイプとドライババージョンによってはこの機能がサポートされていない場合があります。ドライバのアップデートについては、ETASのサポート窓口までお問い合わせください。

この機能を利用すると、あらかじめ指定された一部の変数のみではなく、**任意**の変数をいつでも適合することができます。INCA上でユーザーがある変数の値を変更するたびに、INCAはその変数がすでにエミュレートされているかを調べ、エミュレートされていない場合は、その変数を含むブロックをエミュレートします。

変更された変数を最後の空きブロック上でエミュレートできなくなると、エミュレートされていない変数はすべて書き込み禁止になります。

その際には、ユーザーがメモリページマネージャを使用してワーキングページからリファレンスページへのフラッシュ書き込みを行うことにより、エミュレーションRAMがすべて解放され、適合作業を続行することができます。

### 注記

ダイナミックエミュレーションモードが選択され、かつA2Lファイルに定義されたDATA領域とOFFLINE\_DATA領域をすべて同時にエミュレートできるサイズのエミュレーションRAMが存在する場合は、INCAにおいて「プロジェクトのフルエミュレーション」が有効になります。プロジェクトのフルエミュレーションモードでは、INCAはエミュレーションRAMを自動的に構成して、DATA領域とOFFLINE\_DATA領域全体、つまりすべての適合変数をエミュレートします。どの適合変数を変更してもメモリブロックの再構成を行う必要がなく、適合作業中に変数が書き込み禁止になることもありません。

グループ軸適合時の補間処理は、プロジェクトのフルエミュレーションが有効になっている場合のみ行えます。それ以外の場合は、補間処理が設定されているグループ軸は書き込み禁止となります。

## 3.5 ダイナミックエミュレーション（再構成可能な可変サイズのブロックを使用）

このコンセプトは、基本的には固定サイズのメモリブロックを使用するダイナミックエミュレーションと似ています。しかしこのコンセプトの場合、使用されるメモリブロックのサイズは個別に指定でき、適合作業中に再構成することができるため、エミュレーションRAMの使用を最適化することができます。その結果、同じサイズのプロジェクトで、ユーザーはより多くの変数を適合することが可能になります。

### 注記

この機能を利用するには、A2Lファイル内にCALIBRATION\_METHOD（適合メソッド）としてReconfigurableSizeMovableEmuRAMが定義されている必要があります。

詳しくは、「[A2Lファイルの記述例](#)」（ページ39）を参照してください。

また、xETKデバイスのタイプとドライババージョンによってはこの機能がサポートされていない場合があります。ドライバのアップデートについては、ETASのサポート窓口までお問い合わせください。

実験を開くと、INCAはまず、プロジェクト全体の適合変数をすべてエミュレートすること（「フルエミュレーション」）を試みます。これが不可能な場合、INCAはユーザーオプションの**実験のフルエミュレーション** オプションが有効になっているかどうかに応じて動作します。**実験のフルエミュレーション** オプションが**オン（Yes）** に設定されていれば、INCAは実験に割り当てられている適合変数のみをすべてエミュレートすることを試みます。これにより、実験での適合中に時間のかかるエミュレーションを禁止することができます。実験のフルエミュレーションが不可能な場合、INCAはワーキングページの値がリファレンスページの値と異なる適合変数をエミュレートしようとします。

実験を開いたときの動作は、ユーザーオプションで実験のフルエミュレーションが有効になっているかどうかによって異なります。フルエミュレーションでは、INCAは適合変数をすべて同時にエミュレートすることを試みます。これは、実験において必要なすべての適合変数がすでに構成されている場合に有用です。

実験を開いた後に適合変数を追加するような場合においては、フルエミュレーションを無効にする必要があります。その場合、実験を開くと、INCAはPC上のリファレンスページとワーキングページの値を比較します。続いて、INCAはA2Lファイルに定義されたエミュレーションRAM情報に従っ

で異なるサイズの一連のメモリブロックを算出して初期設定します。エミュレートされるのは、値が変更された変数（つまりワーキングページの値がリファレンスページの値と異なる変数）に限られます。その後も、変数の値が変更されるたびに初期設定されたブロック内のいずれか（1つまたは複数）を使用して自動的にエミュレーションを行います。

オペレータが適合操作を進め、INCAが使用できるエミュレーションRAMのメモリブロックがなくなると、エミュレートされていない変数はすべて書き込み禁止になります。そのような場合、ユーザーはエミュレーションRAMのレイアウトを再構成するためのコマンドを実行できます。INCAは、変更された適合変数に合わせて、できる限り小さいサイズのブロックとエミュレーションメモリを使用する新しいエミュレーションRAMレイアウトを算出します。この目的は、適合操作に使用できる1つ以上のメモリブロックを確保することです。

メモリレイアウトを最大限効率的に使用してそれ以上の効率化が行えなくなり、変更された変数をエミュレートできなくなった場合は、ユーザーがメモリページマネージャを使用してワーキングページからリファレンスページへのフラッシュ書き込みを行う必要があります。これによりエミュレーションRAMがすべて解放され、適合作業を続行することが可能になります。

## 3.6 両コンセプトの比較

下の表では、手動で実験準備を行う従来のコンセプトと、ダイナミックエミュレーションのコンセプトの最も重要な特徴を比較したものです。

手動操作による実験準備	ダイナミックエミュレーション
<p>選択された適合変数がすべてエミュレートされるため、プロジェクトの規模によっては、適合変数の一部（「サブセット」）しか同時に適合することができません。</p> <p>適合される変数のサブセットは、エミュレーション領域のブロックごとに定義されます。</p>	<p>プロジェクトに含まれるすべての適合変数を任意に適合できます。</p> <p>実際に適合される変数のみ（つまり相違分だけ）がエミュレートされます。</p>
<p>実験の準備作業として、エミュレートする必要のある変数を前もって選択し、エミュレーションRAMの割り当てを固定的に設定しておく必要があります。INCAは、選択された変数をすべて一度にエミュレートできるかどうかを調べます。できない場合は、選択操作を繰り返しながら変数の数を調整する必要があります。</p>	<p>実験の準備は一切必要ありません。ユーザーはメモリブロックを構成する必要はありません。エミュレーションRAMの割り当ては、適合作業中に必要に応じて自動的に行われます。</p>
<p>実験開始時にオペレータが手動で変数を選択しなければならないため、テストベンチやHardware-in-the-LoopアプリケーションなどからINCAをリモート操作する場合には適していません。</p>	<p>オペレータによる介入が少ないため、テストベンチやHardware-in-the-LoopアプリケーションなどにおけるINCAのリモート操作に適しています。</p> <ul style="list-style-type: none"> <li>－ ブロックサイズが固定の可変エミュレーションRAMメソッドの場合： <p>すべてのブロックが使用されると、すべての差異を書き込んでエミュレーションRAMを解放するまで適合操作が行えなくなります。</p> </li> <li>－ ブロックサイズが可変で再構成可能な可変エミュレーションRAMメソッドの場合： <p>ユーザーがエミュレーションRAMの再構成を要求できます。再構成が不可能になった場合に限り、すべての差異を書き込んでエミュレーションRAMを解放するまで適合操作が行えなくなります。</p> </li> </ul>
<p>前もって計画されたとおりに実験が行われます。</p>	<p>エミュレーションRAMに空きがなくなると、オペレータへの通知が行われます。</p>

## 4 手動での実験準備

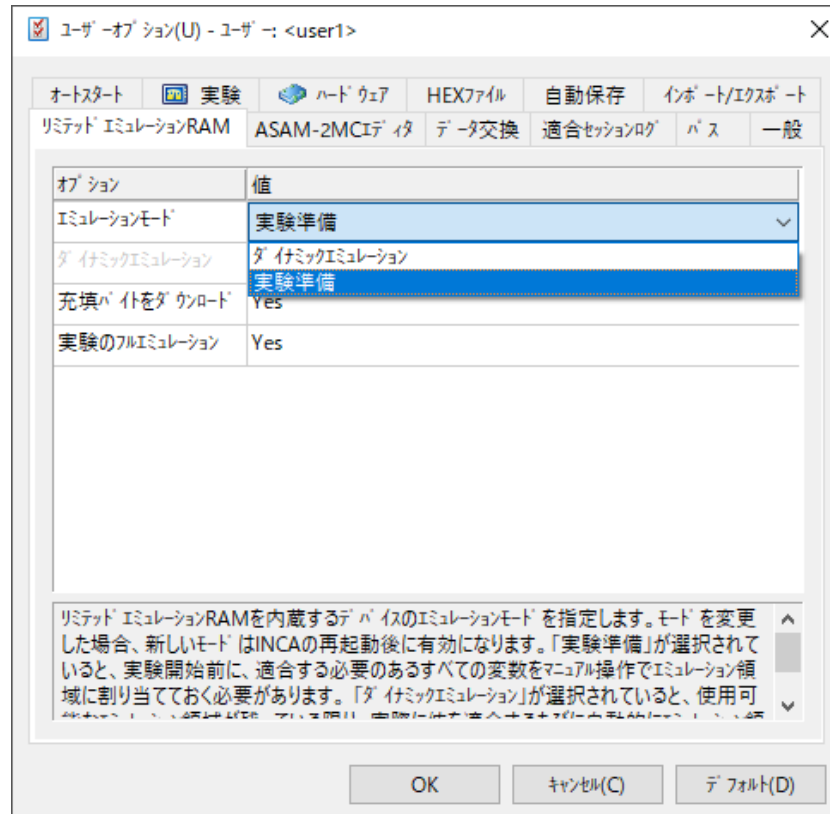
本章では、リミテッドエミュレーションRAMに関する従来のコンセプトを用いて実験を行う方法を説明します。

### 4.1 手動で実験準備を行うエミュレーションのためのユーザーオプション設定

シリアルxETKを用いた実験開始時に、リミテッドエミュレーションRAMの割り当てを行うための実験準備ダイアログボックスが開くようにするには、ユーザーオプションを以下のように設定します。

#### 実験準備を行うエミュレーション用のユーザーオプションを設定する

1. INCAデータベースマネージャで、**オプション > ユーザーオプション > 開く** を選択します。
2. "ユーザーオプション" ダイアログボックスの "リミテッドエミュレーションRAM" タブを選択します。
3. **エミュレーションモード** パラメータの値として **実験準備** を選択します。



4. INCAを終了して再起動し、変更内容を有効にします。

### 4.2 ワークスペースの準備

実験を開始する際には、通常どおりに以下のような準備作業を行ってください。

1. A2LファイルとHEXファイルを読み込んで、プロジェクトとデータセットを作成します。

### 注記

リミテッドエミュレーションRAMのメモリブロックエミュレーションメカニズムを有効にするには、A2Lファイル内にCALIBRATION\_METHOD（適合メソッド）として FixedSizeMoveableEmuRAMが定義されている必要があります。詳しくは、「[A2Lファイルの記述例](#)」（ページ39）を参照してください。

また、xETKデバイスのタイプとドライババージョンによってはこの機能がサポートされていない場合があります。ドライバのアップデートについては、ETASのサポート窓口までお問い合わせください。

2. ワークスペースを作成し、ハードウェアコンフィギュレーションをセットアップします。

### 注記

ハードウェアコンフィギュレーションエディタでのパラメータ設定により、作業を最適化することができます。詳しくは「[ご使用上のヒント](#)」（ページ44）を参照してください。

## 4.3 実験の実行

### 4.3.1 初期化

INCAハードウェアとの通信の初期化は、通常どおりに行ってください。詳細は「[INCA入門ガイド](#)」を参照してください。

INCAは、初期化処理の初めに各メモリページのチェックサムを計算し、ページ間の不一致がないかを調べます。エミュレーションRAMを使用する場合、リファレンスページ（RP）とワーキングページ（WP）の両方のチェックサムを計算するため、リファレンスページをアクティブにする必要があります。そのため、WPがアクティブになっている状態から初期化を行うと、INCAはオペレータに対してページ切り替えを行う旨を通知します。その通知に対して **はい** ボタンをクリックすると、INCAはRPをアクティブにして初期化処理を続行し、終了後、再びWPをアクティブに戻します。INCAがPCとECU間でチェックサムの不一致を検知すると、メモリページマネージャが開きます。その場合は、必要に応じてメモリページ操作を行ってください。

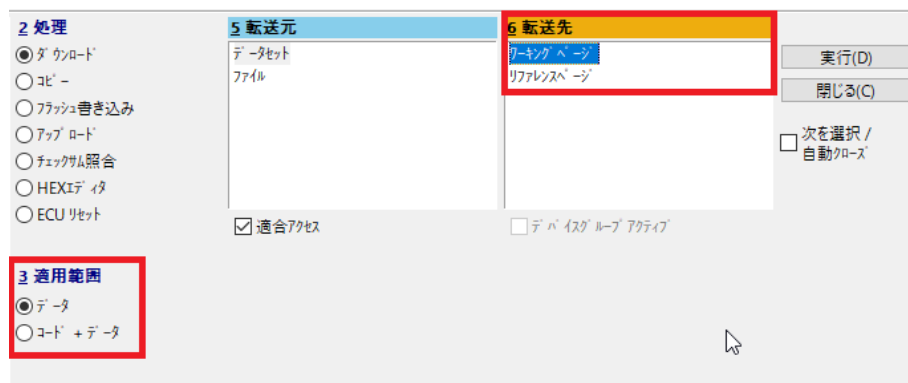
### 4.3.2 メモリページ操作の制限事項

エミュレーションRAMを使用している場合、メモリページマネージャでの操作において以下のような制約があります。

#### － ダウンロード

ダウンロードはワーキングページに対してのみ可能です。使用するエミュレーションRAMブロックは実験環境において選択できます。詳しくは「[エミュレートする領域の設定](#)」（次ページ）を参照してください。

コードページとリファレンスページをECUへ転送するには、フラッシュ書き込みを行ってください。



#### ー チェックサムの照合

チェックサムの照合を行う際は、リファレンスページへの切り替えが必要になる場合があります。以下のメッセージが表示された場合は、その内容に従ってページを切り替えてください。

#### ー コピー

一般的なxETKの場合、ECUのワーキングページからリファレンスページへ直接コピーを行うことはできません。リファレンスページからワーキングページへのコピーのみ可能です。

ワーキングページのデータをリファレンスページに反映するには、メニューコマンド **データセット > ワーキングデータのフリーズ** を選択する必要があります。その後、新しいリファレンスページの内容をECUのフラッシュメモリに書き込みます。詳しくはINCAのオンラインヘルプを参照してください。

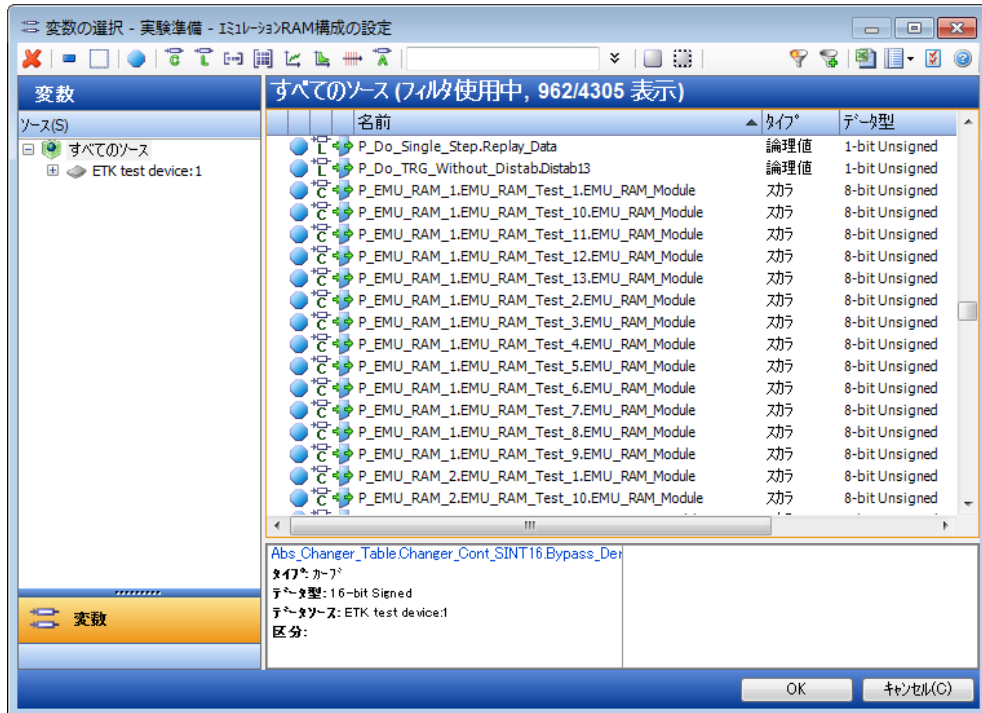
#### 注記

データフリーズ機能をサポートしているタイプのECUとxETKを使用する場合は、上記の制限は適用されず、「WPからRPへのコピー」は「ワーキングデータのフリーズ」と同様に機能します。つまりINCAはワーキングページからリファレンスページへのコピーを行い、そのバックグラウンドでECUのDATA領域のフラッシュ書き込みを行います。

### 4.3.3 エミュレートする領域の設定

実験に使用する変数を選択する前に、エミュレーションRAMの割り当てを設定し、フラッシュメモリ上のどのブロックをエミュレーションRAMでエミュレートするかを決定しておく必要があります。

INCAは、「[INCAによるエミュレーションRAMの自動設定](#)」(ページ13)に示される手順で、エミュレーションRAMの自動割り当てを試みます。ほとんどの場合、エミュレートする適合変数の数が多すぎるため、INCAはユーザーの操作なしではエミュレーションRAMを構成できません。この場合は、以下のような "実験準備 - エミュレーションRAM構成の設定" ダイアログボックスが自動的に開きます。



### 注記

下記のような条件においては、エミュレーションRAM設定の必要はないため、上記のダイアログボックスは開きません。

- － エミュレーションRAMのサイズ $\geq$ データ領域のサイズ

### 注記

上記の "実験準備 - エミュレーションRAM構成の設定" ダイアログボックスの外見は "変数の選択" ダイアログボックスと似ていますが、機能は異なります。

- － "実験準備" ダイアログボックス :  
ここでは、実験において実際に適合を行う変数を選択し、エミュレーションRAM構成をセットアップします。
- － "変数の選択" ダイアログボックス :  
ここでは、実験に割り当てる測定変数と適合変数を選択します。  
エミュレートされているブロックに含まれない適合変数を選択することも可能です。しかしこれらの変数は、ワーキングページがアクティブになっていても、実験環境ウィンドウ内においてグレイアウトされ、適合することはできません。

必要に応じて、エミュレーションRAMの設定を以下のように行ってください。

#### エミュレーションRAMの構成を設定する

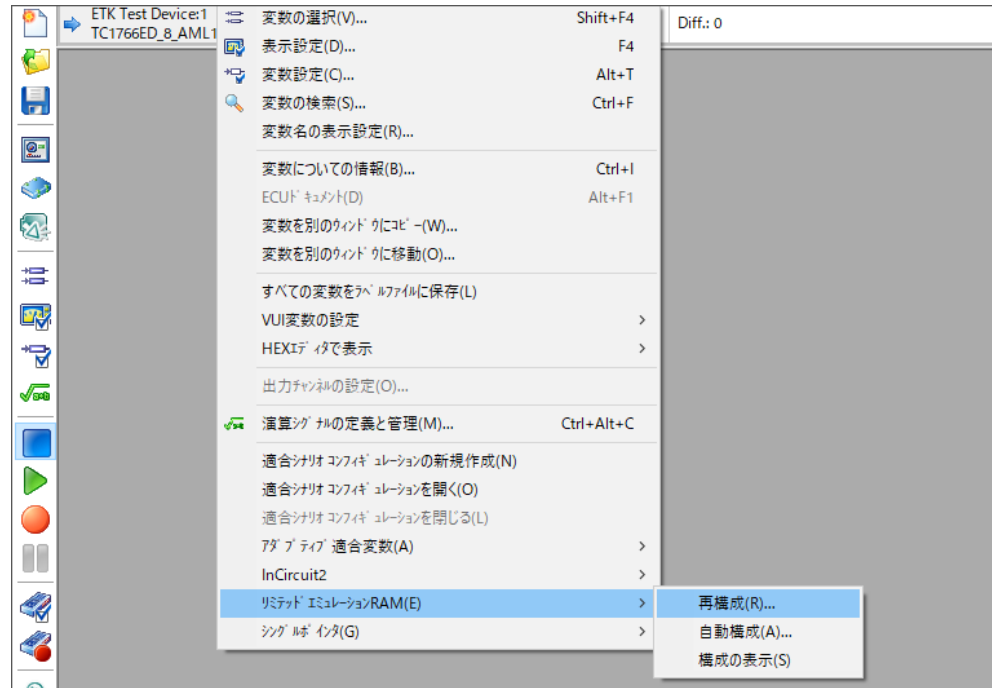
エミュレーションRAMの構成は、実験環境の "実験準備 - エミュレーションRAM構成の設定" ダイアログボックスで設定する必要があります。このダイアログボックスは、次の2通りの方法で開くことができます。



1. ハードウェア初期化時に、xETK上に保存された設定と実験に保存された設定との不一致が検知されると、実験環境で作業する前にダイアログボックスが自動的に開きます。  
xETKに保存された設定は、電源を切ってもそのまま残ります。

または

実験を開く際にダイアログボックスが自動的に開かない場合は、メニューコマンド **変数 > リミテッドエミュレーションRAM > 再構成** を選択して手動で開きます。

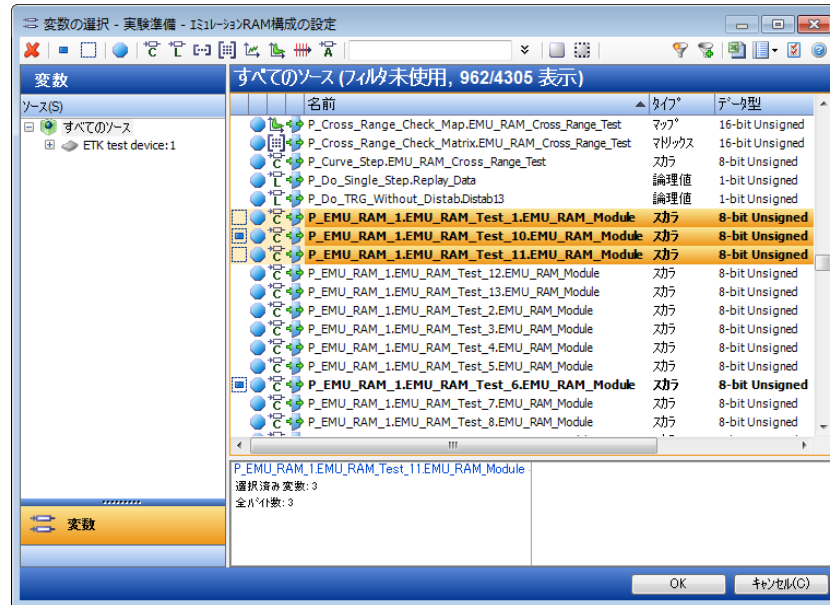


2. ワーキングページのデータとリファレンスページのデータの値が一致しない場合は、ワーキングページのデータをフリーズさせてフラッシュ書き込みを行うことを推奨するメッセージが表示されます。
3. ここでは必ず **はい** をクリックし、フラッシュ書き込みを行ってください。

### 注記

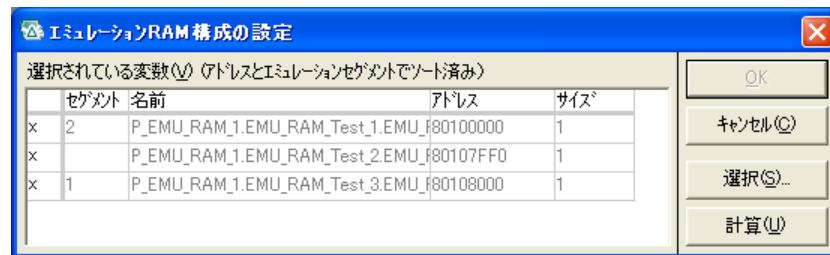
**いいえ** をクリックすると、ECUとINCAとの間でワーキングページのデータが一致しなくなる可能性があります。

ワーキングページのデータがフラッシュメモリに書き込まれます。終了すると "実験準備 - エミュレーションRAM構成の設定" ダイアログボックスが開きます。変数リスト内にすべての適合変数が表示されます。



4. 適合を行う変数を選択します。
5. **OK** をクリックします。

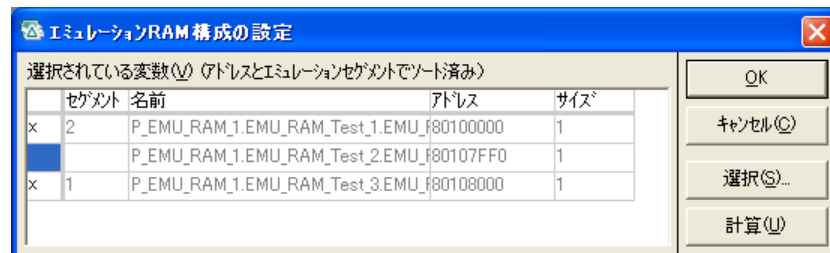
選択された変数をすべて同時にエミュレートできる場合、設定操作はここで完了します。エミュレーションRAMのサイズが不十分で一部の変数をエミュレートできない場合、「エミュレーションRAM構成の設定」ダイアログボックスが開きます。



上図の例では、使用できるエミュレーションRAMのブロックが2つしかないため、選択された3つの変数をすべて同時にエミュレートすることはできません。このため **OK** ボタンは無効になっています。

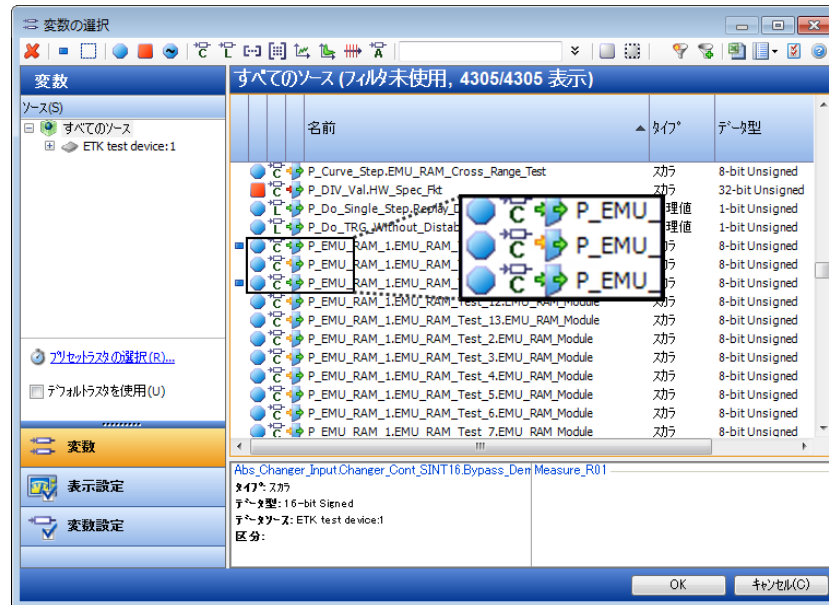
6. エミュレーションRAMの割り当てを調整するため、いずれかの変数の行に表示されている "x" マークを消し、その変数を無効にします。
7. **計算** ボタンをクリックします。

構成が適切であれば、**OK** ボタンが有効になります。



ここでまだ **OK** ボタンが有効にならない場合は、さらに別の変数を無効にして、上述のように再計算を行います。

8. 割り当てが完了したら、**OK** をクリックしてダイアログボックスを閉じます。  
設定されたエミュレーションRAMの割り当てが、確定されます。
9. 通常どおりに "変数の選択" ダイアログボックスを開きます。



上図の例のとおり、エミュレートされていない（つまり適合を行えない）適合変数については、変数のシンボルの左側にオレンジの矢印が表示されます。適合可能な適合変数には緑色の矢印が表示されます。

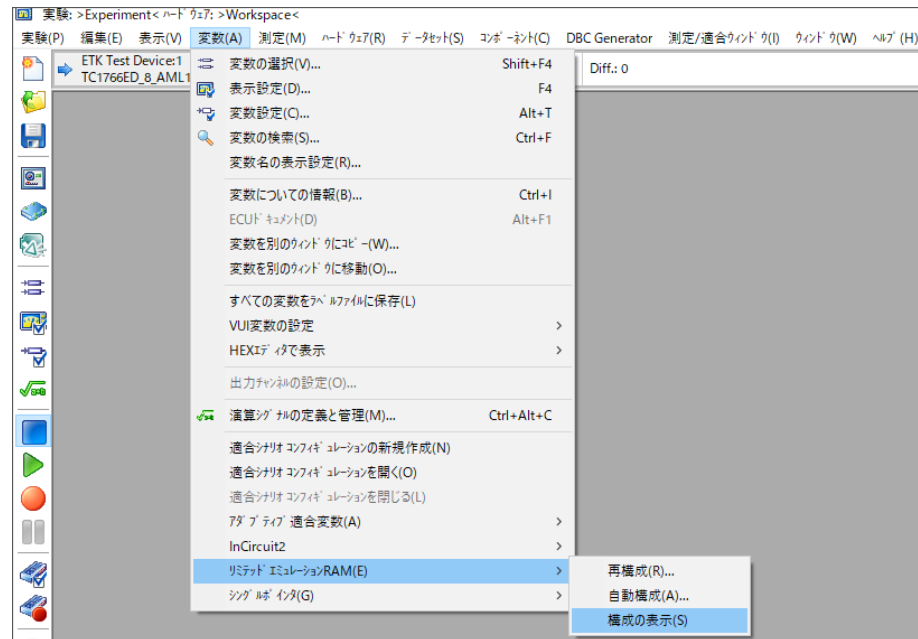
10. 通常どおり、実験で使用する変数を選択します。

#### 4.3.4 エミュレーションRAMの割り当て状況の確認

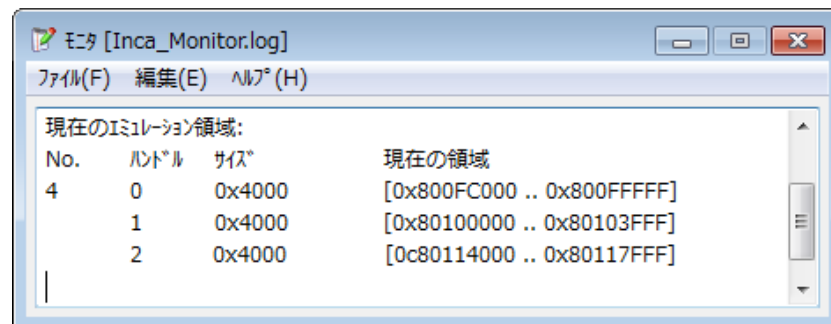
現在設定されているエミュレーションRAMの構成（ブロック番号、エミュレートされるフラッシュメモリブロックのアドレスなど）を確認することができます。以下のように操作してください。

## エミュレーションRAMの現在の構成を表示する

1. 実験環境を開きます。
2. **変数 > リミテッドエミュレーションRAM > 構成の表示** を選択します。



INCAモニタウィンドウに現在の構成が表示されます。



## 5 ダイナミックエミュレーション

本章では、INCAのダイナミックエミュレーションのコンセプトを用いて実験を行う方法を説明します。

### 5.1 ダイナミックエミュレーションのユーザーオプションの設定

シリアルxETKのダイナミックエミュレーション機能を使用するには、ユーザーオプションを以下のように設定します。

#### ダイナミックエミュレーションに必要なユーザーオプションを設定する

1. INCAデータベースマネージャで、**オプション > ユーザーオプション > 開く**を選択します。
2. "ユーザーオプション" ダイアログボックスの "リミテッドエミュレーションRAM" タブを選択します。
3. **エミュレーションモード** パラメータの値として **ダイナミックエミュレーション** を選択します。

#### 注記

ダイナミックエミュレーション機能をサポートしないタイプのxETKを使用している状態で "ダイナミックエミュレーション" モードに変更すると、INCAは自動的にモードを切り替えます。

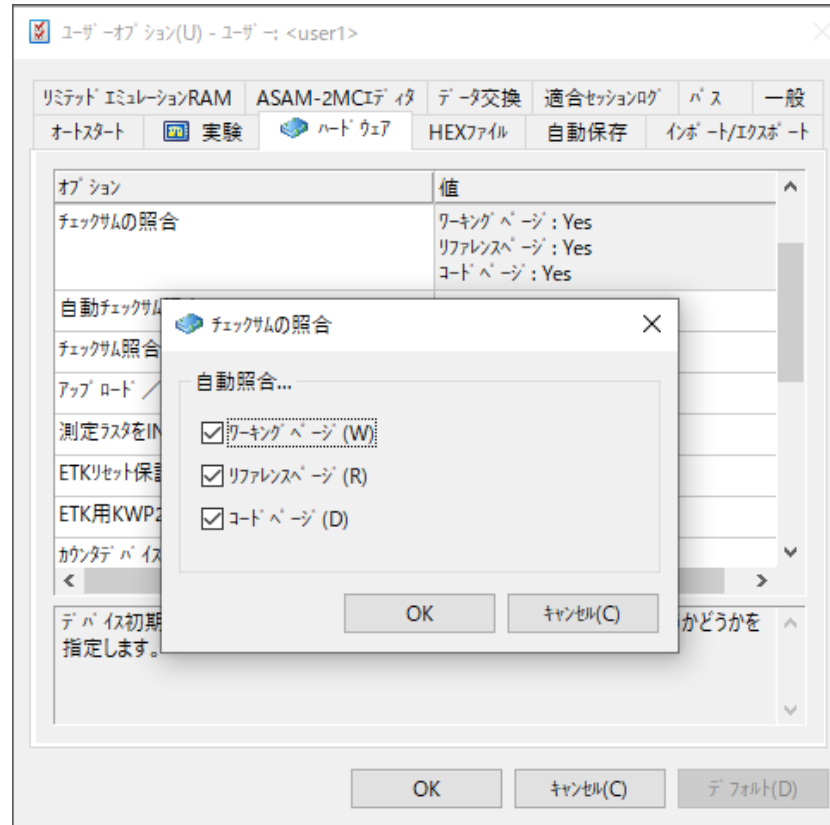
A2Lファイルに定義されているCALIBRATION\_METHOD (適合メソッド) の値が FixedSizeMoveableEmuRAMの場合は実験準備モード (デフォルトモード) に、ReconfigurableSizeMovableEmuRAMの場合はオフラインモードに切り替えます。

どちらの場合も、モードの切り替え時にはメッセージで通知します。

4. 実験のフルエミュレーションを行うかどうかを指定します。
  - 必要な適合変数がINCAの実験にすでに割り当てられている場合は、フルエミュレーションを有効にしておいてください。実験を開いた際に最適なエミュレーションRAM構成が得られます。
  - INCAの実験を開いた後に適合変数を追加する場合は、フルエミュレーションは無効にしておいてください。有効の状態では実験を開くと、その時点において実験に割り当てられているすべての適合変数がエミュレートされてしまうため、さらに適合変数を追加すると、エミュレーションRAMの空きセグメントが不足してしまう可能性があります。

ダイナミックエミュレーションを使用するには、上記の設定以外に、初期化時にリファレンスページとワーキングページのチェックサム計算が行われるように設定する必要があります。

5. "ユーザーオプション" ダイアログボックスの "ハードウェア" タブを選択します。
6. **チェックサムの照合** パラメータの値フィールドで、ワーキングページとリファレンスページのチェックサム計算がどちらもアクティブ (Yes) になっていることを確認します。



### 注記

このパラメータの値が No になっていると、チェックサムの照合が行われないため、リミテッドエミュレーションRAMの機能はワーキングページとリファレンスページの内容が異なっていると判断し、適合アクセスを無効にします。

7. **OK** をクリックして設定内容を確定します。その後、INCAを終了して再起動し、変更内容を有効にします。

さらに他のユーザーオプションを設定して作業を円滑に行えるようにすることができます。詳細については、「ご使用上のヒント」(ページ44)を参照してください。

## 5.2 ワークスペースの準備

実験を開始する際には、通常どおりに以下のような準備作業を行ってください。

1. A2LファイルとHEXファイルを読み込んで、プロジェクトとデータセットを作成します。

### 注記

リミテッドエミュレーションRAMのメモリブロックエミュレーションメカニズムを有効にするには、A2Lファイル内にCALIBRATION\_METHOD（適合メソッド）として FixedSizeMoveableEmuRAMまたは ReconfigurableSizeMovableEmuRAMが定義されている必要があります。詳しくは、「[A2Lファイルの記述例](#)」（ページ39）を参照してください。

2. ワークスペースを作成し、ハードウェアコンフィギュレーションをセットアップします。

### 注記

ハードウェアコンフィギュレーションエディタでのパラメータ設定により、作業を最適化することができます。詳しくは「[ご使用上のヒント](#)」（ページ44）を参照してください。

## 5.3 実験の実行

### 5.3.1 初期化

INCAハードウェアとの通信の初期化は、通常どおりに行ってください。操作の方法はPDFマニュアル「INCA 入門ガイド」、および本書の「[初期化](#)」（ページ22）という項を参照してください。

### 注記

A2Lファイルに以下のダイナミックエミュレーションメソッドが定義されている場合、INCAは ReconfigurableSizeMovableEmuRAMを使用します。

ダイナミックエミュレーションメソッド：

```
FixedSizeMoveableEmuRAM
FixedSizeNonMoveableEmuRAM
FlashRangeSpecificEmuRAM
ReconfigurableSizeMoveableEmuRAM
```

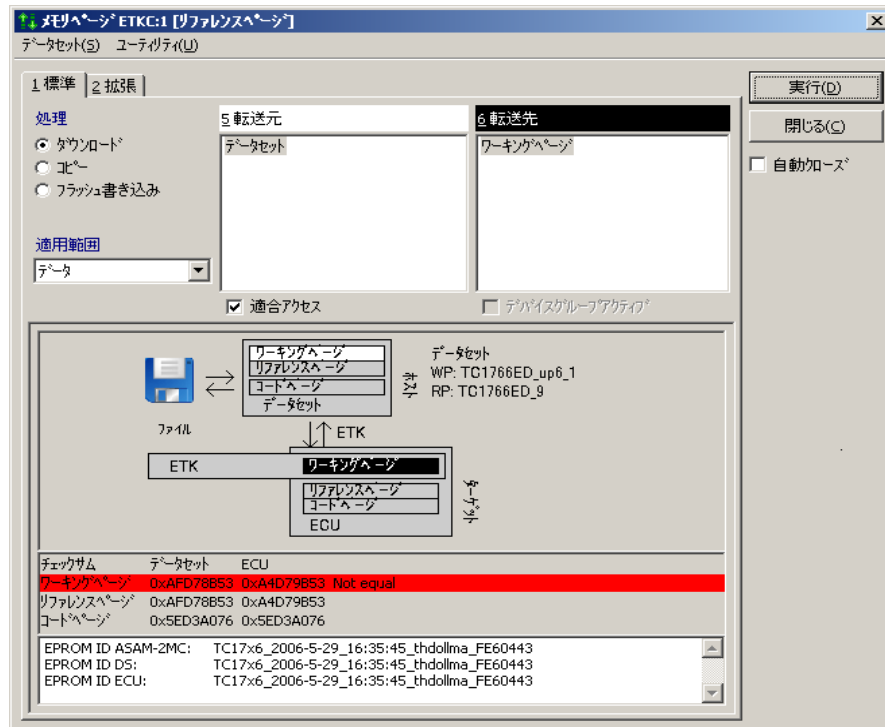
初期化時に、INCAは各メモリページのチェックサムを計算して照合します。INCAとECUのワーキングページとリファレンスページが一致していれば、すぐに適合を開始できます。（「[ダイナミックエミュレーションによる適合の実行](#)」（ページ33）を参照）

一方、INCAがPCとECU間でチェックサムの不一致を検知すると、メモリページマネージャが開きます。その場合は、必要に応じてメモリページ操作を行ってください。

### 5.3.2 メモリページ間の不一致の解決

以下の3通りの状況が考えられます。

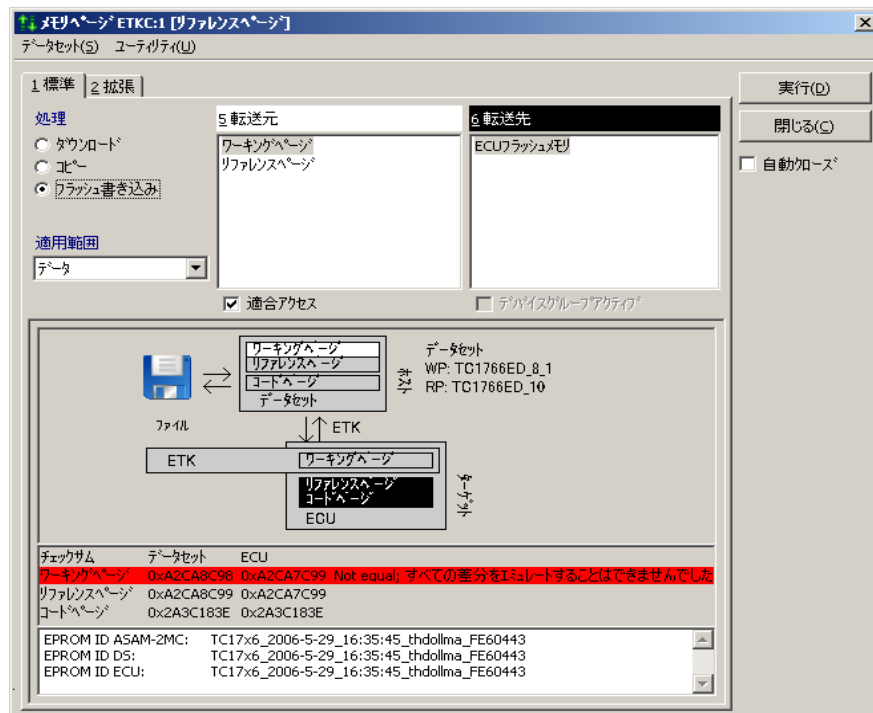
## INCAとECUとでワーキングページの内容が異なる場合



## データの不一致を解決する

1. INCAのワーキングページをECUにダウンロードして、適合が行えるようにします。

## INCAとECUとでワーキングページの内容が異なり、不一致の度合いが大きい場合

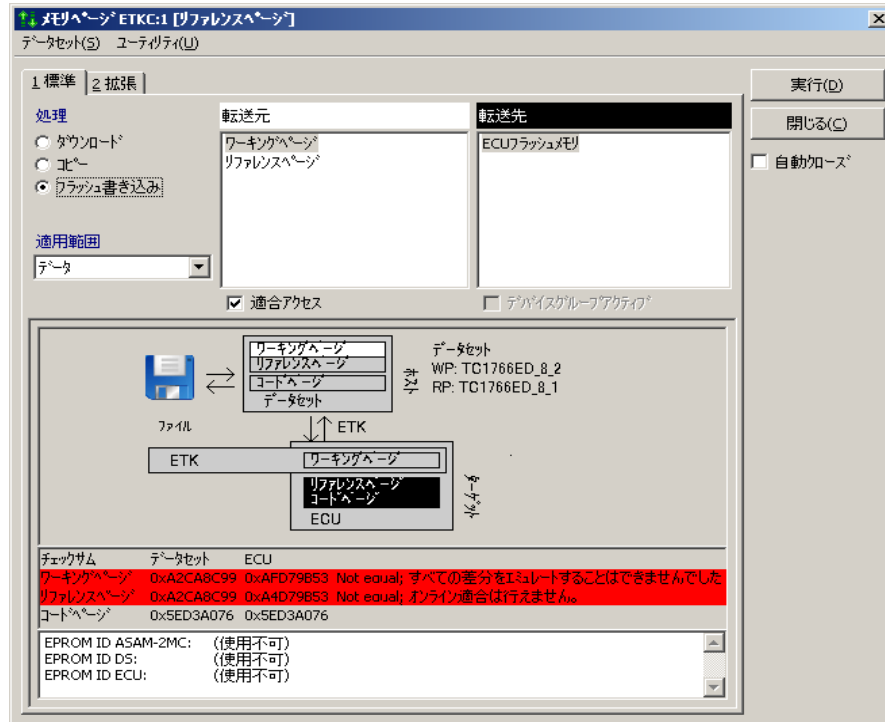




### データの不一致を解決する

1. INCAのワーキングページをECUのフラッシュメモリに書き込んで、適合が行えるようにします。

### INCAとECUとでリファレンスページの内容が異なる場合



### データの不一致を解決する

1. INCAのワーキングページをECUのフラッシュメモリに書き込んで、適合が行えるようにします。

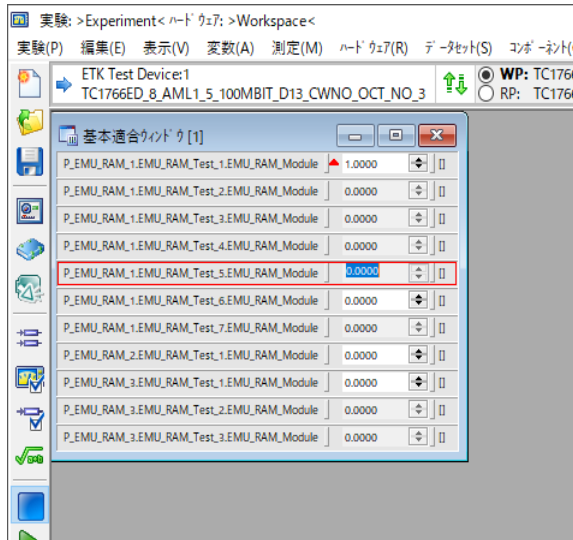
または

INCAのリファレンスページをECUのフラッシュメモリに書き込み、INCAのワーキングページをECUにダウンロードします。この場合、ワーキングページとリファレンスページの内容は一致しないままになります。

### 5.3.3 ダイナミックエミュレーションによる適合の実行

ダイナミックエミュレーションを利用して適合を行う場合は、実験環境において通常どおりに変数の選択や表示設定を行い、適合作業を開始します。前述のメモリページの操作以外の準備作業は必要ありません。

ただし、一度にエミュレートできる上限数以上の変数が実験に含まれている場合があります。エミュレートされていない変数はすべて書き込み禁止になり、これらの変数の値は、エディタにおいて以下のように灰色の背景で表示されます。



以下のいずれかの操作を行います。

- － エミュレーションRAMを再構成して、適合したい適変数をエミュレートします。
- － ダイナミックエミュレーションによりエミュレーションRAMレイアウトを最適化します。

### ① 注記

上記の2つの機能を使用するには、A2Lファイル内にCALIBRATION\_METHOD (適合メソッド) としてReconfigurableSizeMovableEmuRAMとFlashRangeSpecificEmuRAMが定義されている必要があります。詳しくは、「[A2Lファイルの記述例](#)」(ページ39)を参照してください。

- － 適合変数の値をリファレンスページの値に戻すか、またはINCAのワーキングページをECUにフラッシュ書き込みして、エミュレーションRAMを解放します。

### **エミュレーションRAMを再構成して目的の変数がエミュレートされるようにする。**

以下のようにしてエミュレーションRAMの再構成を行うことにより、現在書き込み禁止になっている変数を適合できるようにすることができます。この再構成処理においては、INCAは1つの適合ウィンドウ内のすべての変数、または選択された変数のエミュレーションを試みます。

#### **1つの適合ウィンドウに含まれる一部の変数を適合する**

1. 適合ウィンドウ内で適合したい変数を選択します。
2. 適合ウィンドウのショートカットメニューから、**リミテッドエミュレーションRAM > 選択済み変数のエミュレーションを試みる**を選択します。

#### **1つの適合ウィンドウに含まれるすべての変数を適合する**

1. 適合ウィンドウのショートカットメニューから、**リミテッドエミュレーションRAM > 全変数のエミュレーションを試みる**を選択します。

INCAは、指定された変数をすべてエミュレートできる場合にのみエミュレーションRAMを再構成します。一部しかエミュレートできない場合は再構成を行いません。再構成が不可能な場合は、下記の方法でエミュレーションRAMを解放してください。

### 注記

エミュレーションRAMの再構成が行われる際には、前もって、INCAがユーザーに対してリファレンスページへの切り替えを要求します。

#### エミュレーションRAMを再構成して最適化する

以下のように操作すると、特定の変数だけでなく、エミュレーションRAM全体の構成を最適化することができます。

1. 実験環境のメニューから **変数 > リミテッドエミュレーションRAM > 自動構成** を選択します。

INCAは、ワーキングページとリファレンスページの間で値が異なっている変数を調べ、それに応じたエミュレーションRAMの効率的なレイアウトを算出します。この処理によって解放されたエミュレーションRAMは、他の変数のエミュレーションに使用できるようになります。

再構成を行っても十分なメモリが確保できない場合や、再構成が不可能な場合は、下記の方法でエミュレーションRAMを解放することができます。

### 注記

エミュレーションRAMの再構成が行われる際には、前もって、INCAがユーザーに対してリファレンスページへの切り替えを要求します。

#### 値のリセットまたはフラッシュ書き込みによってエミュレーションRAMを開放する

エミュレーションRAMを解放して適合を続行できるようにするには、次のように操作します。

1. 適合ウィンドウ内の変数を選択し、ショートカットメニューまたは **編集** メニューから **リファレンスページの値にリセット** を選択して、リファレンスページの値に戻します。

リファレンスページの値がワーキングページに書き込まれ、両ページの差異がなくなるため、その変数用に使用されていたエミュレーションRAMは他の変数用に使用できるようになります。

または

メモリページマネージャを開き、INCAのワーキングページをECUにフラッシュ書き込みします。

INCAのワーキングページの値とECUの値が等しくなります。

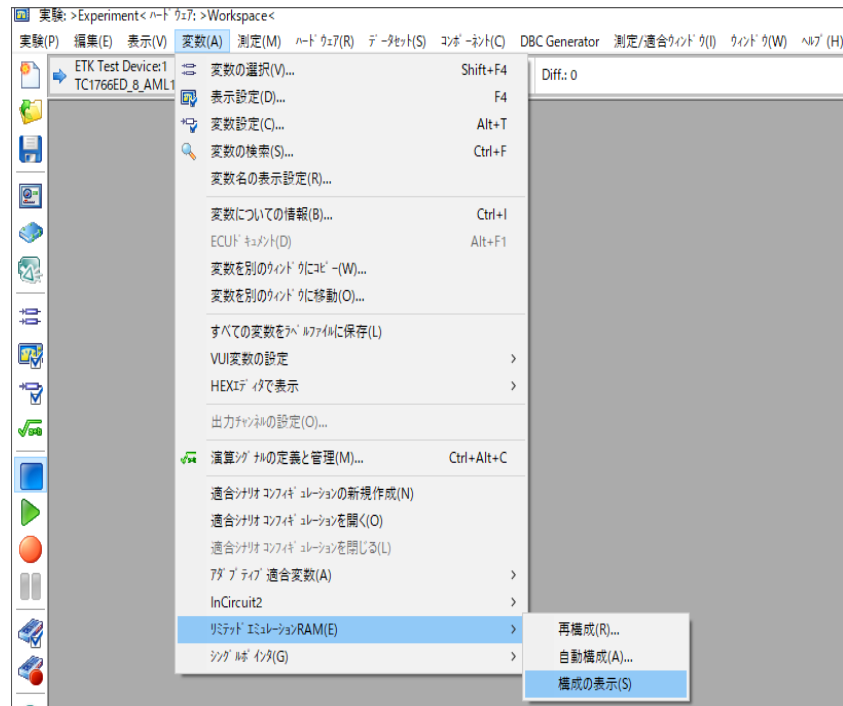
INCAの実験に含まれる適合変数は、ダイナミックエミュレーションによってすべて書き込み可能な状態になり、適合操作の続行が可能になります。

### 5.3.4 エミュレーションRAMの割り当て状況の確認

現在設定されているエミュレーションRAMの構成（ブロック番号、エミュレートされるフラッシュメモリブロックのアドレスなど）を確認することができます。以下のように操作してください。

## エミュレーションRAMの現在の構成を表示する

1. 実験環境を開きます。
2. **変数 > リミテッドエミュレーションRAM > 構成の表示** を選択します。



INCAモニタウィンドウに現在の構成が表示されます。



## 6 制限事項

本製品は、ユーザーの期待に応えるために、品質、使いやすさ、完全性に細心の注意を払って製作されています。

ただし、ご使用の状況によっては以下のような制限事項がありますので、ご注意ください。

### 6.1 一般的な制限事項

- － リミテッドエミュレーションRAMの機能は、以下のタイプの適合変数をサポートしていません。
  - Dependent
  - Virtual
  - 書き込み禁止に指定されている適合変数
  - DATAメモリセグメントまたはOFFLINE\_DATAメモリセグメント（オプション）の外に配置されている適合変数
  - RESERVEDメモリセグメント内に配置されている適合変数

- － 一部のETKデバイス（ETK-S20.0など）の場合、エミュレーションRAMのレイアウトが、ETKデバイスの不揮発性メモリ上ではなくETASインターフェースモジュール（ES592など）のRAM上に保存されます。

そのため、ETASインターフェースモジュールが電源オフになったり省電力モードに切り替わると、エミュレーションRAMレイアウトの内容が失われてしまいます。それにより、次の初期化時には、INCAとECUとの間でワーキングページのチェックサム値の差異が検出されるため、INCAからECUへのダウンロード処理が必要になります。

XETK デバイスの場合も同様で、XETK デバイスの電源が切断されたりXETK デバイスがスタンバイモードに切り替わると、次の初期化時にINCAとECUのワーキングページのチェックサムの差異が検出されます。

### 6.2 従来のコンセプト（実験開始時にエミュレーションRAMの設定を行う方法）の制限事項

- － 実験準備を行う際の変数選択はASAP3で自動化できないため、このコンセプトは、リモート操作によるユースケース（テストベンチ、Hardware-in-the-Loopアプリケーションなど）には適していません。

### 6.3 新しいコンセプト（ダイナミックエミュレーション）の制限事項

- － ASAP2のデータ範囲外の変数の書き込み禁止：A2Lファイル内でDATAセグメントとして定義されたメモリセグメントの範囲外のメモリブロックに配置された変数については、オンラインアクセスによる書き込みは禁止されます。データセット内のOFFLINE\_DATAセグメントに配置されたものについては、オフラインで適合することができますが、データセットのアドレス範囲外に配置された適合変数については一切の適合操作は行えません。
- － メモリページマネージャにおいて、ファイルからワーキングページへのダウンロードはサポートされていません。

- グループECUの適合：ECUのグループ適合を行うには、実験環境において、適合する変数をグループ内の各ECUごとに個別に表示できることが必要条件となります。個別に表示できない場合は、1つのECUしか適合できません。この状況が発生する条件としては、各ECUプロジェクト内で定義されたエミュレーション領域の数が異なる場合や、各ECUごとにワーキングページとリファレンスページの差異が異なる場合などが考えられます。
- 再構成可能な可変サイズのブロックサイズを用いたダイナミックエミュレーションを行うには、A2Lファイル内にCALIBRATION\_METHOD（適合メソッド）として ReconfigurableSizeMovableEmuRAMが定義されている必要があります。詳しくは、「[A2Lファイルの記述例](#)」（ページ39）を参照してください。また、xETKデバイスのタイプとドライババージョンによってはこの機能がサポートされていない場合があります。ドライバのアップデートについては、ETASのサポート窓口までお問い合わせください。

## 7 A2Lファイルの記述例

リミテッドエミュレーションRAMの機能は、メモリブロックのオーバーレイメカニズム（=エミュレーションメカニズム）を用いてシリアルxETKでの適合作業を実現するものです。このメカニズムを実際に適合ツールで使用するには、A2Lファイル内に所定の記述が必要です。

以下のいずれかをA2Lファイル内の CALIBRATION\_METHOD（適合メソッド）の下に定義します。

- FixedSizeMoveableEmuRAM
- FixedSizeNonMoveableEmuRAM
- FlashRangeSpecificEmuRAM.
- ReconfigurableSizeMoveableEmuRAM

個々のエミュレーションブロックは、ハンドル番号、開始アドレス、ブロックサイズにより定義されます。使用できる値は、ECU上のマイクロコントローラと、使用されているxETKに応じて異なります。

### 7.1 A2Lファイルの記述例 – FixedSizeMoveableEmuRAMを使用する場合

適合メソッド **FixedSizeMoveableEmuRAM** を使用する場合は、一般的には全ブロックのサイズは同じです。一部のマイクロコントローラでは異なるサイズのブロックを組み合わせて使用することもできますが、この場合もサイズをダイナミックに変更する（=「再構成する」）ことはできません。どちらの場合もブロックサイズはA2Lファイルに定義します。

#### 注記

A2Lファイルにおいて、各エミュレーションブロックはCALIBRATION\_HANDLEとして個別に定義されます。

ここで定義される各ブロックのサイズと先頭アドレスは相互に関連し合っています。エミュレーションブロックのサイズ（以下の例では "0x4000"）により、リファレンスページのセグメント境界が決まるため、ブロックサイズが0x4000の場合、各ブロックの先頭アドレスは0x4000の倍数である必要があります。

#### 全ブロックのサイズが同じ場合

```
/begin CALIBRATION_METHOD
  "FixedSizeMoveableEmuRAM" /* Method name */
  1 /* Method version */
  /begin CALIBRATION_HANDLE
    0 /* Handle no. (emu register no.of CPU) */
    0xFFFF0000 /* Start address of emu RAM (in RAM area) */
    0x4000 /* Length of emu RAM */
  /end CALIBRATION_HANDLE
  /begin CALIBRATION_HANDLE
```

```

    1          /* Handle no.(emu register no.of CPU)      */
    0xFFFF4000 /* Start address of emu RAM (in RAM area)  */
    0x4000     /* Length of emu RAM                      */
/end CALIBRATION_HANDLE

...
/end CALIBRATION_METHOD

```

### 各ブロックのサイズが異なる場合

```

/begin CALIBRATION_METHOD
  "FixedSizeMoveableEmuRAM" /* Method name      */
  1                          /* Method version  */
  /begin CALIBRATION_HANDLE
    0          /* Handle no.(emu register no.of CPU)      */
    0xFFFF0000 /* Start address of emu RAM (in RAM area)  */
    0x8000     /* Length of emu RAM                      */
  /end CALIBRATION_HANDLE
  /begin CALIBRATION_HANDLE
    1          /* Handle no.(emu register no.of CPU)      */
    0xFFFF8000 /* Start address of emu RAM (in RAM area)  */
    0x4000     /* Length of emu RAM                      */
  /end CALIBRATION_HANDLE
...
/end CALIBRATION_METHOD

```

#### 注記

サイズが異なるブロックを使用できるかどうかは、xETKデバイスのタイプとドライババージョンに依存します。

さらに、マイクロコントローラのタイプに応じて使用できるブロックのサイズが制限される場合があります（例：2種類のブロックサイズが使用可能で、2つめのサイズはもう1つのブロックの2倍であること）。

## 7.2 A2Lファイルの記述例 – FixedSizeNonMoveableEmuRAMを使用する場合

### 設定例

```

/begin CALIBRATION_METHOD
  "FixedSizeMoveableEmuRAM" /* Method name      */
  1                          /* Method version  */
  /begin CALIBRATION_HANDLE
    0          /* Handle          */
    0xBF400000 /* Emu RAM start address */
  /end CALIBRATION_HANDLE

```



```

0x20000          /* Emu RAM size          */
0x00300000      /* Flash start address      */
/end CALIBRATION_HANDLE
...
/end CALIBRATION_METHOD

```

### 注記

固定サイズで移動不可能なブロックを使用できるかどうかは、xETKデバイスのタイプとドライババージョンに依存します。さらに、マイクロコントローラのタイプに応じて、使用できるブロックサイズやブロック数、各ブロックのフラッシュ可能領域などが制限される場合があります。

## 7.3 A2Lファイルの記述例 – FlashRangeSpecificEmuRAMを使用する場合

```

/begin CALIBRATION_METHOD
  "FlashRangeSpecificEmuRAM"      /* Method name          */
  1                                /* Method version       */
  /begin CALIBRATION_HANDLE
    0                              /* Overlay type: 0->
      ClusterOverlay              */
    32                            /* Handle start index   */
    32                            /* Handle count         */
    4                             /* Handle group size    */
    0x8000                        /* Handle page size     */
    0x8000                        /* Handle alignment     */
    1                             /* Number of Emu RAM ranges*/
    0xFB000000                   /* Emu RAM start address */
    0x100000                      /* Emu RAM size         */
    /* If more than one Emu RAM
    range is specified, start
    address and size is to be
    added here          */
    2                             /* Number of Flash Ranges */
    0x00000000                   /* Flash Bank A start
    address              */
    0x00400000                   /* Flash Bank A size     */
    0x00400000                   /* Flash Bank B start
    address              */
    0x00400000                   /* Flash Bank B size     */
    /* If more flash ranges are
    specified, start address
    and size is to be added

```

```

        here                                */
/end CALIBRATION_HANDLE
/begin CALIBRATION_HANDLE
    1                                        /* Overlay type: 1->
GlobalOverlay                             */
    0                                        /* Handle start index    */
    32                                       /* Handle count         */
    0x2000                                  /* Overlay min size     */
    0x20000                                 /* Overlay max size     */
    1                                        /* Number of Emu RAM ranges */
    0xFB400000                              /* Emu RAM start address */
    0x180000                                 /* Emu RAM size         */
    1                                        /* Number of Flash Ranges */
    0x00000000                              /* Flash start address  */
    0x00800000                              /* Flash size           */
/end CALIBRATION_HANDLE
/end CALIBRATION_METHOD

```

### 注記

フラッシュ領域が指定されたブロックを使用できるかどうかは、xETKデバイスのタイプとドライババージョンに依存します。

さらに、マイクロコントローラのタイプに応じて、使用できるブロックサイズやブロック数、各ブロックのフラッシュ可能領域などが制限される場合があります。

## 7.4 A2Lファイルの記述例 – ReconfigurableSizeMoveableEmuRAMを使用する場合

```

/begin CALIBRATION_METHOD
    "ReconfigurableSizeMoveableEmuRAM"    /* Method name        */
    1                                       /* Method version     */
        /begin CALIBRATION_HANDLE
            0                               /* Handle Range Lower */
            15                              /* Handle start index  */
            0x20                             /* Handle count       */
            0x20000                          /* Handle group size   */
            0xBF000000                       /* Handle page size    */
            0xC0000                           /* Handle alignment    */
            /* If more than one Emu RAM
range is specified for
each Emu RAM, start
address and size is to be
added here                                */

```

```
        /end CALIBRATION_HANDLE  
/end CALIBRATION_METHOD
```

 **注記**

再構成可能な可変サイズのブロックを使用できるかどうかは、xETKデバイスのタイプとドライババージョンに依存します。  
さらに、マイクロコントローラのタイプに応じて使用できるブロックのサイズに制限がある場合もあります。

ETAS固有の拡張機能を含むAMLディスクリプションの内容については、ETASのサポート窓口までお問い合わせください。

## 8 ご使用上のヒント

本章では、理解しにくい作業や、ちょっとした工夫で最適化できる作業を行う際に役立つヒントをご紹介します。

### ユーザーオプションとハードウェアパラメータの最適化

以下のように、ユーザーオプションやハードウェアコンフィギュレーションエディタで特定のパラメータを設定することで、作業を最適化することができます。

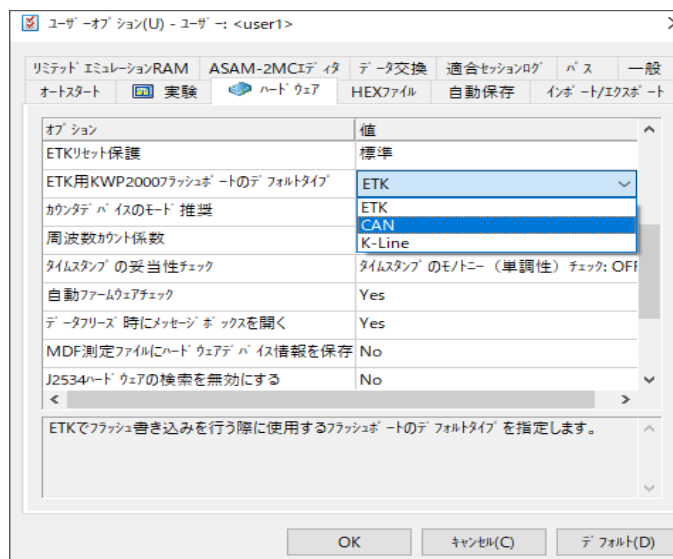
#### ユーザーオプションでデフォルト設定を最適化する

1. INCAデータベースマネージャで、**オプション > ユーザーオプション > 開く**を選択します。
2. **ユーザーオプション** ダイアログボックスの **リミテッドエミュレーションRAM** タブを選択します。
3. 「充填バイト」が充填されたアドレス領域がECUメモリにダウンロードされないようにするには、**充填バイトをダウンロードオプションを No** に変更します。

A2Lファイル内に定義されていないアドレス領域の一部は、INCAにおいては所定の「充填バイト」が代入されている必要があります。しかし、ワーキングページをECUメモリにダウンロードする際は、これらの充填バイトがダウンロードされることによって問題が発生する場合があります。それを解決するため、この**充填バイトをダウンロード**というユーザーオプションが導入されました。このオプションによって、ワーキングページのダウンロード時に、A2Lファイルにおいて「メモリタイプがFLASHであるMEMORY\_SEGMENT」として定義されたアドレス領域だけをダウンロードするか、または上記以外のメモリタイプのアドレス領域で充填バイトが代入されている部分もダウンロードするかを指定することができます。

4. **ユーザーオプション** ダイアログボックスの **リミテッドエミュレーションRAM** タブを選択します。
5. **ダイナミックエミュレーション**のオプションの値を**RPが不一致の状態**で**適合を許可**に変更します。  
RPが不一致の状態での適合は、使用されているECUプロジェクトのすべての適合変数がエミュレート可能な場合に実施可能です。
6. **ユーザーオプション** ダイアログボックスの **ハードウェア** タブを選択します。
7. ハードウェア初期化時にメモリページ切り替えの確認操作を行わないようにするには、**ページ切り替え時の確認**というオプションの値を **No** にします。
8. 新しく追加したECUデバイスに適用されるフラッシュポートのデフォルトタイプ（ETK、CAN、K-Lineのいずれか）を設定するには、**ETK用KWP2000フラッシュポートのデ**

フォルトタイプ オプションの値を変更します。



9. **OK** をクリックして設定内容を確定します。

### ハードウェアコンフィギュレーションエディタで各ETKの設定を最適化する

1. ハードウェアコンフィギュレーションエディタの **ハードウェアデバイス** リストから、適合作業に使用するシリアルETKを選択します。
2. デフォルトで使用するフラッシュポートのタイプを設定するには、ウィンドウ右側の **パラメータ** リストで **ProF用フラッシュポート** パラメータの値を変更します。
3. メモリページの切り替え時に確認操作を行うようにするかどうかを指定するには、**パラメータ** リストで **ページ切り替え時の確認** というパラメータを設定します。

## 8 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>