

The logo for ETAS, consisting of the letters 'ETAS' in a bold, blue, sans-serif font. The 'E' and 'S' have a unique, blocky design with horizontal lines.

DRIVING EMBEDDED EXCELLENCE

ETAS INTECRIO V5.0

入門ガイド

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は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 2022** ETAS GmbH, Stuttgart, Germany

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。

INTECRIO は ETAS GmbH の登録商標です。

MATLAB および Simulink は The MathWorks, Inc. の登録商標です。

INTECRIO V5.0 入門ガイド R03 JP – 11.2022

目次

1	安全と個人情報保護についての注意事項	7
1.1	安全に関する情報	7
1.2	本書の対象ユーザー	7
1.3	使用目的	7
1.4	安全に関する注意事項の記述書式	8
1.5	安全に関する注意事項	9
1.6	個人情報保護に関する注意事項	9
1.6.1	データの処理	9
1.6.2	データとデータカテゴリ	9
1.6.3	技術的／組織的な対策	10
2	INTECRIO について	11
2.1	システム情報	11
2.2	本書の構成	12
2.3	INTECRIO オンラインヘルプ	14
3	インストール	16
3.1	準備	16
3.1.1	パッケージの内容	16
3.1.2	システム要件	17
3.1.3	インストールと操作に必要なユーザー権利	17
3.2	INTECRIO のインストール	17
3.2.1	初回のインストール	17
3.2.2	特殊なインストールの手順: 仮想プロトタイピング (Virtual Prototyping) パッケージ	25
3.2.3	コマンドラインからのインストール	25
3.3	ETAS Experiment Environment (ETAS 実験環境) のインストール	28
3.3.1	Experiment Environment のシステム要件	28
3.3.2	初回のインストール	28
3.3.3	コマンドラインからのインストール	33
3.4	ライセンス管理に関する設定	35
3.5	ソフトウェアのライセンスを取得する	38
3.6	アンインストールする	38
4	INTECRIO クイックガイド	42
4.1	はじめに	42
4.2	新しいワークスペースの作成	43
4.3	モジュールの準備とインポート	44
4.3.1	モジュールの準備	44
4.3.2	モジュールのインポート	45
4.4	ファンクションの作成	46
4.5	ソフトウェアシステム／環境システムの作成	49
4.6	ハードウェアシステムの作成と設定	52

4.6.1	ハードウェアシステムのインポート.....	53
4.6.2	デジチェーンの設定.....	54
4.6.3	LIN コントローラの設定.....	55
4.6.4	バイパスの設定.....	57
4.6.5	CAN コンフィギュレーションファイルのインポート.....	60
4.6.6	CAN コンフィギュレーションのエクスポート.....	61
4.7	システムプロジェクトのセットアップ.....	61
4.7.1	システムプロジェクトの作成.....	62
4.7.2	ハードウェアとソフトウェアの接続.....	63
4.8	オペレーティングシステムの設定.....	64
4.8.1	オペレーティングシステムの自動設定.....	64
4.8.2	オペレーティングシステムの手動設定.....	65
4.9	実行ファイルの生成.....	68
4.10	実験.....	69
4.10.1	実験の準備.....	69
4.10.2	測定/適合ウィンドウの作成とセットアップ.....	71
4.10.3	測定変数と適合変数の割当て.....	73
4.10.4	レイヤの使用.....	74
4.10.5	実験の管理.....	75
4.10.6	データ収集の設定.....	76
4.10.7	実験の実行.....	77
4.10.8	測定と適合.....	79
4.10.9	ポストプロセッシング.....	80
4.11	仮想プロトタイピング (バーチャルプロトタイピング).....	82
4.12	ワークスペースの管理と交換.....	84
4.12.1	ワークスペースの管理.....	84
4.12.2	インポート/エクスポートによるワークスペースの交換.....	85
4.12.3	外部で変更されたワークスペースを開く.....	86
4.13	グラフィックエディタ上のブロックのレイアウト編集.....	87
4.14	ドキュメントの自動生成.....	89
5	INTECRIO チュートリアル.....	90
5.1	はじめに.....	90
5.1.1	準備.....	90
5.1.2	記述形式について.....	91
5.1.3	概念.....	91
5.1.4	チュートリアルの概要と目標.....	94
5.2	レッスン 1: INTECRIO 用モジュールの準備.....	96
5.2.1	本レッスンで利用する重要な概念.....	96
5.2.2	Simulink® モデル.....	96
5.3	レッスン 2: 準備.....	99
5.3.1	本レッスンで利用する重要な概念.....	99
5.3.2	INTECRIO とワークスペース.....	99
5.4	レッスン 3: モジュール、ファンクション、ソフトウェアシステム、環境システム... 102	
5.4.1	本レッスンで利用する重要な概念.....	102

5.4.2	モジュールのインポート	102
5.4.3	ファンクションの作成	104
5.4.4	ソフトウェアシステムと環境システムの作成	108
5.5	レッスン 4: ハードウェアシステム	111
5.5.1	本レッスンで利用する重要な概念	111
5.5.2	ハードウェアシステムの作成とセットアップ	111
5.6	レッスン 5: システムプロジェクト	112
5.6.1	本レッスンで利用する重要な概念	112
5.6.2	システムプロジェクトの作成	113
5.6.3	ソフトウェアシステムと環境システムの接続	116
5.7	レッスン 6: プロトタイプの生成	118
5.7.1	本レッスンで利用する重要な概念	118
5.7.2	オペレーティングシステムのコンフィギュレーション設定	119
5.7.3	ビルド処理	123
5.8	レッスン 7: 実験	126
5.8.1	本レッスンで利用する重要な概念	126
5.8.2	準備	126
5.8.3	実験のセットアップ	127
5.8.3.1	測定/適合インストゥルメントの作成	128
5.8.3.2	測定/適合変数の割当て	131
5.8.3.3	測定/適合インストゥルメントのセットアップ	131
5.8.3.4	測定/適合インストゥルメントの保存	135
5.8.4	データ記録に関する設定	135
5.8.5	測定/適合タスクの実行	137
5.8.5.1	シミュレーションの開始	137
5.8.5.2	測定	138
5.8.5.3	適合	140
5.8.5.4	記録	141
5.8.5.5	仮想プロトタイプピング実験の特殊な機能と条件	141
5.9	レッスン 8: ES930 – 設定と操作の例	143
5.9.1	本レッスンの最重要コンセプトの概要	143
5.9.2	ES930 デイジーチェーンモジュールの設定	144
5.9.3	ES910 ハードウェアシステムの作成とセットアップ	146
5.9.3.1	ES900 システム	146
5.9.3.2	ES910 シミュレーションターゲット	147
5.9.3.3	デイジーチェーンデバイス	147
5.9.4	ES930 による実験	150
5.9.4.1	システムプロジェクトの作成	150
5.9.4.2	ES910 の設定	153
5.9.4.3	実験の実行	155
6	ETAS ネットワークマネージャ	158
7	一般的なトラブルシューティング	159
7.1	ETAS ネットワーク用のネットワークアダプタを選択できない	159
7.2	PC に接続されたイーサネットハードウェアが検出できない	160

8	お問い合わせ先	164
9	用語集	165
9.1	略語	165
9.2	用語	169
	図	174
	索引	176

1 安全と個人情報保護についての注意事項

この章では、本製品の使用目的、対象ユーザー、安全に関する情報および個人情報保護について説明しています。

ETAS Safety Advice(Help > Safety Advice) とユーザードキュメントに記載されている安全上のご注意に従ってください。

1.1 安全に関する情報

本製品を使用する際には、作業者の負傷や物的損害などを避けるため、「ETAS Safety Advice - 安全上の注意事項」、および下記の注意事項をよくお読みいただき、その指示に従ってください。

1.2 本書の対象ユーザー

本書は、組み込み制御システムの機能領域およびソフトウェア開発を専門とする、訓練を受けた従業員向けの内容となっています。

INTECRIO を使用するユーザーは、Windows® 8、Windows® 8.1、Windows® 10 の操作方を理解する必要があります。メニューコマンドの実行やボタン操作の方法に加え、Windows のファイルシステム、特にファイルやディレクトリで構成される階層構造についての知識が必要です。また Windows ファイルマネージャや Windows エクスプローラ等の使い方、さらに「ドラッグアンドドロップ」操作に習熟していることも必要です。

Microsoft Windows のマウスおよびキーボード操作に慣れていない方や、一般的なファイルのナビゲーション方法についてご存知でない方は、まずそれらについて学習した後に INTECRIO をご使用いただくことをお勧めします。Windows オペレーティングシステムについての詳しい説明は、マイクロソフトのドキュメントを参照してください。

INTECRIO がサポートしている BMT (ASCET や MATLAB/Simulink) についての知識も必要条件となりますが、さらに、プログラミング言語、特に ANSI-C の知識があれば、INTECRIO を一層効率的にご使用いただけます。

1.3 使用目的

INTECRIO は、車両組み込みコントロールシステムのプロトタイピング向けの統合プラットフォームです。ASCET、MATLAB、Simulink、C コードなどのさまざまなソースから、ECU 上にアプリケーションソフトウェアを統合できます。

バーチャルプロトタイピングは、開発時間を短縮するということから、非常に重要視されています。INTECRIO のバーチャルプロトタイピング能力を使用すると、複雑なプロトタイピングハードウェアを必要とせず、システムモデルの分析が可能となります。

INTECRIO のラピッドプロトタイピング能力を使用すると、実際の状況下、また実車両において、制御、診断機能の確認や検証が可能です。プロトタイプは、ETAS ラピッドプロトタイピングハードウェアを介して、既存の ECU 車両ネットワークに統合することができます。ETK、XETK、FETK、XCP を介したパイパスアプリケーションにおいて、INTECRIO は、ラピッドプロトタイピングハードウェアがシミュレーションコントローラーとして使用され、ECU の新機能に用いるパラメーターの計算をします。

製品の不適切な使用や安全に関する注意事項に従わないことにより生じた一切の損害について、ETAS GmbH は責任を負いません。

1.4 安全に関する注意事項の記述書式

安全に関する注意事項は以下の書式で記述されます。これらの情報は必ずよくお読みいただき、指示に従ってください。



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。

1.5 安全に関する注意事項

ユーザーの怪我や所有物の損傷を回避するため、安全に関する注意事項（「ETAS Safety Advice - 安全上のご注意」）、および下記の注意事項をよくお読みいただき、その指示に従ってください。



警告

不適切に初期化された NVRAM 変数は、車両やテストベンチの予期しない挙動を招く恐れがあり、そのような挙動により人身事故や物的損害が発生する危険性があります。

E-Target（実験ターゲット）の NVRAM 機能を使用する INTECRIO システムにおいては、ユーザー定義された初期化プロセス内で、すべての NV 変数の値がカレントプロジェクトに対して有効な状態になっているかを、個々の NV 変数単位および他の NV 変数との関連性においてチェックする必要があります。もし有効な状態になっていない場合は、すべての NV 変数をデフォルト値（有意で安全な値）に設定してください。

データ保存に関する NVRAM の特性から、不適切な初期値が使用されることにより人体や装置が傷付けられる可能性のある環境内（車上やテストベンチなど）においてプロジェクトが使用される場合は、特にこの注意事項を厳守してください。

ETAS 製品を安全にご使用いただくための注意事項（「ETAS Safety Advice - 安全上の注意事項」）は、以下の方法でもご覧いただけます。

- ・ インストールパッケージで以下のファイルを開く
Documentation\General\ETAS Safety Advice.pdf
- ・ プログラムで Help > Safety Advice を選択する

1.6 個人情報保護に関する注意事項

ユーザーの個人情報保護の問題は ETAS にとっても重要な案件であるため、本項では、INTECRIO 内でどのようなデータが処理されるか、どのような種類のデータを INTECRIO が使用するか、また個人情報の保護のためにユーザー自身がどのような技術的対策を講じるべきか、といった内容について説明します。さらに、本製品が個人データを保存する場所や、それらのデータの削除方法についても説明します。

1.6.1 データの処理

本製品の使用時には、個人データが処理されます。本製品の購入者は、GDPR（General Data Protection Regulation：EU の一般データ保護規則）の Article 4 No. 7 に従ってデータ処理を行う法的責任があります。製造者である ETAS GmbH は、当該データの不適切な扱いに関して、いかなる場合も責任を負いません。

1.6.2 データとデータカテゴリ

本製品は、エラー解析や外部プログラムとのやり取りなどの目的で、ソースライブラリを参照するファイル名とパスを含むファイルを作成します。

ファイルの名前とパスがカレントユーザーの個人ディレクトリやそのサブディレクトリを参照している場合は、以下のように個人データがファイル名とパスに含まれる可能性があります。

C:\Users*<UserId>*\Documents\...

さらに、試験車両内で ETAS ラピッドプロトタイピングソリューションを用いて実センサや車載バス、ECU などに接続すると、ETAS ツールは運転者の個人データにアクセスすることが可能になります。

これらのデータは、INCA-EIP や ETAS Experiment Environment (ETAS 実験環境) によりデータロガーに保存される可能性もあります。

また、ETAS ライセンスマネージャでユーザーベースライセンスを扱うと、以下のような個人データが、ライセンス管理の目的で記録される可能性があります。

- 通信データ：IP アドレス
- ユーザーデータ：ユーザー ID、Windows のユーザー ID

1.6.3 技術的／組織的な対策

本製品は、個人データを記録する際に暗号化を行いません。記録されるデータの機密保持のため、ユーザー側の IT システムに適した技術的対策または組織的対策を講じてください。

ログファイル内の個人データは、オペレーティングシステムのツールを用いて削除することができます。

2 INTECRIO について

昨今の ECU 開発においては、実際のターゲットハードウェアが存在しない状態で組み込み制御ソフトウェアの制御アルゴリズムを開発しなければならない、というケースが多くなっています。この際、アルゴリズムは ASCET や MATLAB®/ Simulink® などの BMT (Behavior Modelling Tool: 挙動モデリングツール)、つまり、記述されたモデルからソフトウェアコードを生成できるツールを用いて作成されます。そして、ターゲットハードウェアに代わるものとして ETAS の ES800 システムなどのラピッドプロトタイピングハードウェアシステムが使用されます。

ラピッドプロトタイピング用プラットフォームである「INTECRIO」は、組み込み制御ソフトウェアの開発に携わるエンジニアの日々の作業を支援する、新しい ETAS 製品ファミリーです。

INTECRIO は、各種 BMT で生成されたコードを統合し、ラピッドプロトタイピング用にモデルプロトタイプやハードウェアの構成をセットアップして、実行コードを生成します。

本書は、INTECRIO 製品の概要説明とともに、迅速かつ確実な結果を得るための情報を提供するものです。

2.1 システム情報

INTECRIO 製品ファミリーは以下のコンポーネントで構成されています。

- **INTECRIO-IP**

INTECRIO の統合プラットフォームです。各種 BMT で記述されたモデルを統合してプロトタイプを生成するための機能がすべて含まれています。またチュートリアルで使用するサンプルファイルと、サードパーティ製 I/O ハードウェアの設定と操作を行うためのサンプルコンフィギュレーションが、INTECRIO のインストール時に指定された <sample files> ディレクトリ (21 ページ参照) にコピーされています。

製品パッケージには以下のコンポーネントが含まれています。

- プロジェクトコンフィギュレータ (Project Configurator) - ソフトウェアシステムとシステムプロジェクトを定義する際に使用されます。
- ハードウェアコンフィギュレータ (Hardware Configurator) - INTECRIO-RP または INTECRIO-VP とともに使用され、プロトタイピングハードウェアの設定や、ハードウェアとソフトウェアの接続を行います。
- OS コンフィギュレータ (OS Configurator) - オペレーティングシステムを設定する際に使用されます。
- プロジェクトインテグレータ (Project Integrator) - システムプロジェクトのすべてのコンポーネントを結合して実行ファイルを生成する際に使用されます。
- ドキュメンタ (Documentor) - システムプロジェクトのコンポーネントについてのドキュメント生成に使用されます。
- MATLAB/Simulink 接続機能 (MATLAB/Simulink Connectivity) - Simulink モデルを INTECRIO に統合してラピッドプロトタイピングを行う際に必要な機能です。

- **INTECRIO-RP**

INTECRIO のラピッドプロトタイピングパッケージです。以下の機能が含まれています。

- ES900 接続機能 (ES900 Connectivity) – 実験ターゲット ES910、およびハードウェアやソフトウェアの接続についての設定を行うための機能です。
- ES800 接続機能 (ES800 Connectivity) – 実験ターゲット ES830 (+ ES891 / ES892 / ES882 / ES886)、およびハードウェアとソフトウェアの接続についての設定を行うための機能です。

- **INTECRIO-VP**

INTECRIO の仮想プロトタイピングパッケージです。以下の機能が含まれています。

- VP-PC 接続機能 (VP-PC Connectivity) – PC を仮想プロトタイピングターゲットとして設定し、使用するための機能です。

- **ETAS Experiment Environment** (ETAS 実験環境)

ETAS 製品用の以下のアドオン製品は、INTECRIO に関連するものです。必要に応じて個別にインストールしてください。

- **ASCET Connectivity** – ASCET モデルを INTECRIO に統合してラピッドプロトタイピングを行うための機能です。

ASCET V6.3 以降では、ASCET Connectivity は ASCET-MD に統合されています。

ASCET V6.2 以前では、ASCET Connectivity の機能は INTECRIO のアドオン「INTECRIO-ASC」として提供され、ASCET-MD 製品に同梱、または ASCET-RP の一部となっています。

2.2 本書の構成

INTECRIO のマニュアルは、個々の操作を説明する『オンラインヘルプ』、インストールについての説明、機能の概説、チュートリアルなどを含む『入門ガイド』、そして INTECRIO でのラピッドプロトタイピングについての詳しい技術情報をまとめた『ユーザーガイド』で構成されています。

『INTECRIO 入門ガイド』の内容は以下のとおりです。

- **第 1 章 「安全と個人情報保護についての注意事項」**

本製品の使用目的、対象ユーザー、安全に関する情報および個人情報保護について説明しています。

- **第 2 章 「INTECRIO について」 (本章)**

INTECRIO の適用分野についての大まかな概要について説明しています。

- **第 3 章 「インストール」**

INTECRIO の製品内容やハードウェア・ソフトウェア要件、および INTECRIO のインストール、メンテナンス、アンインストールの方法を説明します。

- **第 4 章 「INTECRIO クイックガイド」**

INTECRIO の概念を簡単に紹介します。一般的な作業に沿った操作手順をフローチャートで示し、INTECRIO の機能や動作について概説します。

第 5 章 「INTECRIO チュートリアル」

サンプルファイルを用いて INTECRIO の実際の操作方法を説明するもので、主に INTECRIO を初めて使うユーザーを対象としています。チュートリアルはいくつかの短いレッスンに分かれていて、レッスン 7 までは、各レッスンを順に行うことにより一連の基本操作を学習することができ、レッスン 8 は独立した内容になっています。チュートリアルを始める際は、あらかじめ『INTECRIO ユーザーガイド』の「INTECRIO を理解する」の章の内容を十分に理解しておいてください。

注記

ETAS では、INTECRIO を短期間でより詳しく理解するためのユーザートレーニングをご提供させていただきます。詳しくは担当窓口までお問い合わせください。

第 6 章 「ETAS ネットワークマネージャ」

ETAS ネットワークマネージャについての説明です。

第 7 章 「一般的なトラブルシューティング」

特定のソフトウェアやハードウェアに依存しない一般的なトラブルが発生した際の対処方法を説明します。

第 8 章 「お問い合わせ先」

ETAS 本社および各国支社の連絡先です。

第 9 章 「用語集」

『INTECRIO 入門ガイド』（本書）と『INTECRIO ユーザーガイド』で使用されている略語と用語についての説明です。

『INTECRIO ユーザーガイド』の内容は以下のとおりです。

- 「安全と個人情報保護についての注意事項」
本製品の使用目的、対象ユーザー、安全に関する情報および個人情報保護について説明しています。
- 「INTECRIO について」
INTECRIO の大まかな概要と「ユーザーガイド」の目的を説明しています。
- 「INTECRIO を理解する」
INTECRIO システム、および INTECRIO がサポートする開発プロセスについて概説しています。初めて INTECRIO を使用する方は必ずこの章を最初にお読みください。
- 「INTECRIO と AUTOSAR」
INTECRIO の AUTOSAR 対応機能についての概説です。
- 「INTECRIO のコンポーネント」
INTECRIO の各コンポーネントとその機能について説明しています。具体的な操作方法はオンラインヘルプを参照してください。
- 「SCOOP と SCOOP-IX」
SCOOP というインターフェース記述の概念と、その記述言語である SCOOP-IX について説明しています。
- 「モデリングのヒント」
INTECRIO におけるモデリングの概念を説明し、INTECRIO でラピッドプロトタイピングを行うためには BMT でどのようなモデリングを行えばよいかを、具体的に紹介します。

- 「お問い合わせ先」
ETAS 本社および各国支社の連絡先です。
- 「用語集」
『INTECRIO 入門ガイド』（本書）と『INTECRIO ユーザーガイド』で使用されている略語と用語についての説明です。

2.3 INTECRIO オンラインヘルプ

INTECRIO の詳しい操作方法は、製品に組み込まれているオンラインヘルプに記載されています。

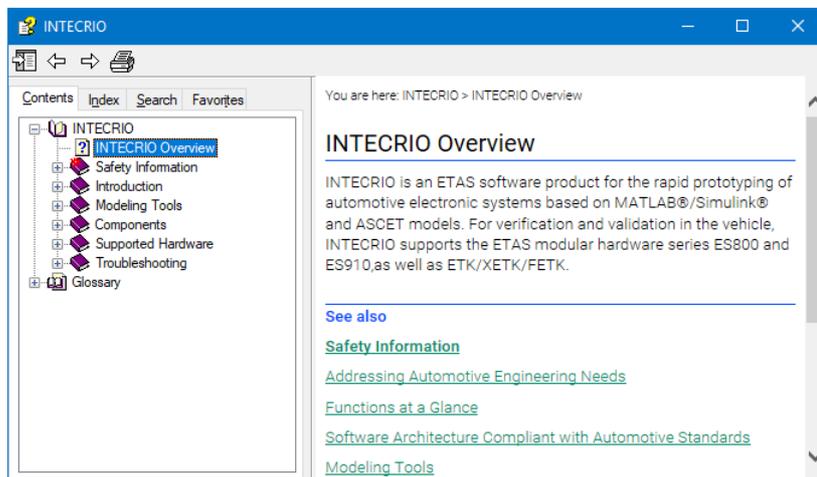
注記

製品インストール時には英語版オンラインヘルプがインストールされますが、代わりに日本語版のヘルプファイルをご使用いただくこともできます。日本語ヘルプファイルは、ETAS ホームページのダウンロードセンター (https://www.etas.com/ja/products/download_center.php) からご提供させていただいております。日本語版ヘルプファイルとそのインストール方法を記載したテキストファイルを含む ZIP ファイルをダウンロードしてご利用ください。

なおバージョンによっては、日本語版ヘルプのリリース時期が遅れる場合や、日本語版がリリースされない場合があります。詳しくはサポート窓口までお問い合わせください。

Help → **Help** メニューコマンド、または  ボタンで、すべてのトピックを含むオンラインヘルプが開きます。またファンクションキー **<F1>** を押すと、現在の状況に依存するトピックが開きます。

英語版：



日本語版：



ヘルプウィンドウの各タブから以下の操作が可能です。

- ・ “目次” タブ：ヘルプトピックを、その内容から検索できます。
- ・ “キーワード” タブ：ヘルプトピックの索引が表示され、キーワードを入力して希望の用語を表示することができます。
- ・ “検索” タブ：ヘルプトピックに含まれる語を検索できます。文字を入力して検索を実行すると、その語を含むトピックの一覧が表示され、そこから任意のページを開くことができます。
- ・ “お気に入り” タブ：トピックをブックマークしておくことができます。
- ・ “用語集” タブ：専門用語とその定義が表示されます。

注記

日本語版マニュアルは ETAS ホームページのダウンロードセンター (https://www.etas.com/ja/products/download_center.php) からご提供させていただいております。

なおバージョンによっては、日本語版マニュアルのリリース時期が遅れる場合や、日本語版がリリースされない場合があります。詳しくはサポート窓口までお問い合わせください。

3 インストール

この章では、オフラインまたはネットワーク経由での INTECRIO と ETAS Experiment Environment のインストール、メンテナンス、およびアンインストールを行う方法について説明します。またシステム管理者の方のための情報として、INTECRIO または ETAS Experiment Environment をファイルサーバーに格納してネットワーク経由でインストールできるようにセットアップする方法についても説明されています。さらに、製品の構成、インストールのためのハードウェア/ソフトウェア要件、インストール準備に関する重要な情報も含まれています。インストールとアンインストールの方法についても、順を追って詳しく説明されています。

各項の内容は以下のとおりです。

- 必要条件 (3.1 項)
- INTECRIO のインストール (3.2 項)
- ETAS Experiment Environment のインストール (3.3 項)
- ライセンス管理についての設定 (3.4 項)
- ソフトウェアのライセンス取得 (3.5 項)
- INTECRIO と ETAS Experiment Environment のアンインストール (3.6 項)

3.1 準備

インストールを行う前に、製品パッケージに含まれるすべてのアイテムが揃っていること、また使用する PC がシステム要件を満たしていることを確認してください。オペレーティングシステムやネットワーク環境によっては、特定のユーザー特権が必要な場合もあります。

3.1.1 パッケージの内容

INTECRIO の製品パッケージには以下のものが含まれています。

- INTECRIO インストールディスク
 - INTECRIO のプログラムファイル
 - 仮想プロトタイピングに必要な ETAS 仮想 OS 実行プラットフォーム (ETAS Virtual OS Execution Platform) / RTA ツールセット
 - INTECRIO の入門ガイド (Getting Started) とユーザーガイド (User Guide)、および ETAS ハードウェアのマニュアル – PDF ファイル¹
 - 安全上の注意事項 - PDF ファイル¹
 - ETAS Experiment Environment (ETAS 実験環境) のプログラムファイル
 - 『Licensing End User Guide』 マニュアル – PDF ファイル¹
 - HSP やデジチェーン・コンフィギュレータなどの最新のプログラムファイルを提供するダウンロードサイトへのリンク
 - MDA のプログラムファイル
 - INTECRIO で使用されるオープンソース・コンポーネントについての情報

さらに、INTECRIO または ETAS Experiment Environment を使用する際にはライセンスが必要です。必要なライセンスファイルの取得については、社内のシステム管理者の方にお問い合わせいただくか、または ETAS にライセンスポータル

サイト（URL は製品に同梱されているレターに記載されています）で発行することができます。なおこのサイトでライセンスを発行するには、製品出荷時に提供されるアクティベーション ID が必要です。

1. > PDF 閲覧ソフトが必要です。

3.1.2 システム要件

INTECRIO のシステム要件は、INTECRIO のリリースノートに記載されています。

ETAS Experiment Environment のシステム要件は、28 ページの 3.3.1 項に記載されています。

3.1.3 インストールと操作に必要なユーザー権利

INTECRIO または ETAS Experiment Environment を PC にインストールするユーザーには管理者特権が必要です。管理者特権が割り当てられていない場合は、システム管理者の方にお問い合わせください。

INTECRIO を操作するユーザーには特別な権利は必要ありません。

ETAS Experiment Environment を操作するのに必要な権利は、使用するターゲットサーバーのバージョンによって異なります。

3.2 INTECRIO のインストール

INTECRIO を使用するには、必ず INTECRIO 統合プラットフォームをインストールする必要があり、さらに用途に合わせたアドオン（ラピッドプロトタイピング用アドオン/仮想プロトタイピング用アドオン）を個々にインストールします。

ETAS Experiment Environment（ETAS 実験環境）は、INTECRIO 統合プラットフォームやそのアドオンとは独立しており、個別にインストールして使用することができます。ETAS Experiment Environment がなくても、INTECRIO でプロジェクトを設定してコードを生成することはできますが、実験を実行することはできず、**Open Experiment Environment** のツールバーボタンとメニューコマンドは使用できません。

INTECRIO と ETAS Experiment Environment のインストール方法は、外部メディアからインストールする場合もネットワークドライブからインストールする場合も、同じです。

本項では以下の内容について説明します。

- INTECRIO の初回インストールの方法（3.2.1 項）
- 仮想プロトタイピングパッケージのインストール方法（3.2.2 項）
- コマンドラインからインストールを行う際のオプション（3.2.3 項）

3.2.1 初回のインストール

操作手順：INTECRIO のインストールを起動する

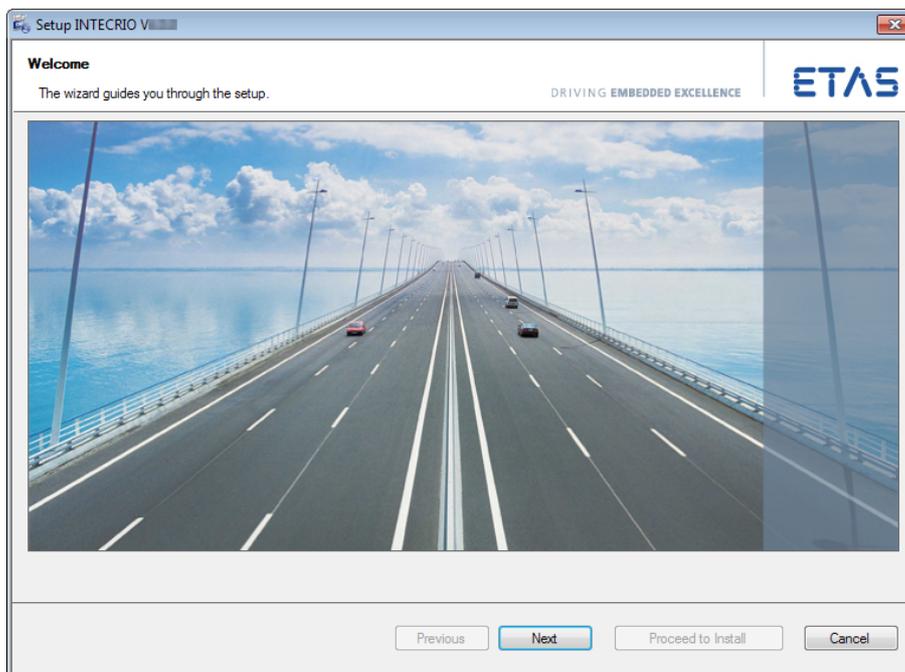
INTECRIO に使用できるオペレーティングシステムの一覧は、INTECRIO リリースノートに記載されています。非対応の OS に INTECRIO をインストールしようとすると、エラーメッセージが表示され、インストール処理が強制終了します。

1. インストールディスクを PC の所定のドライブに挿入します。

2. 以下のいずれかを行います。

- インストーラのウィンドウから **"Main"** リンクをクリックし、さらに **"INTECRIO V5.0.*"** をクリックします。
- Windows エクスプローラでメディアドライブを開き、Installation フォルダから setup.exe というファイルを開きます。

ETAS インストーラが起動します。



3. **Next** ボタンをクリックして次のページに進みます。

前のページに戻るには **Previous**、インストールを中止するには **Cancel** をクリックします。

インストーラは、PC が INTECRIO のインストール要件を満たしているかをチェックし、その結果を "System Check" ウィンドウに表示します。

要件が満たされている場合は、自動的に次のページに進みます。

操作手順: 前回インストールしたバージョンをアンインストールする

注記

互換性のないバージョンの INTECRIO がインストールされていない場合は、「ライセンスに関する合意文書と安全に関する注意事項を確認する」(19 ページ)に進みます。

INTECRIO V5.0.2 との互換性のないバージョンの INTECRIO が PC にインストールされている場合は、「Uninstall previous products」ページが開き、そのバージョンが表示されます。

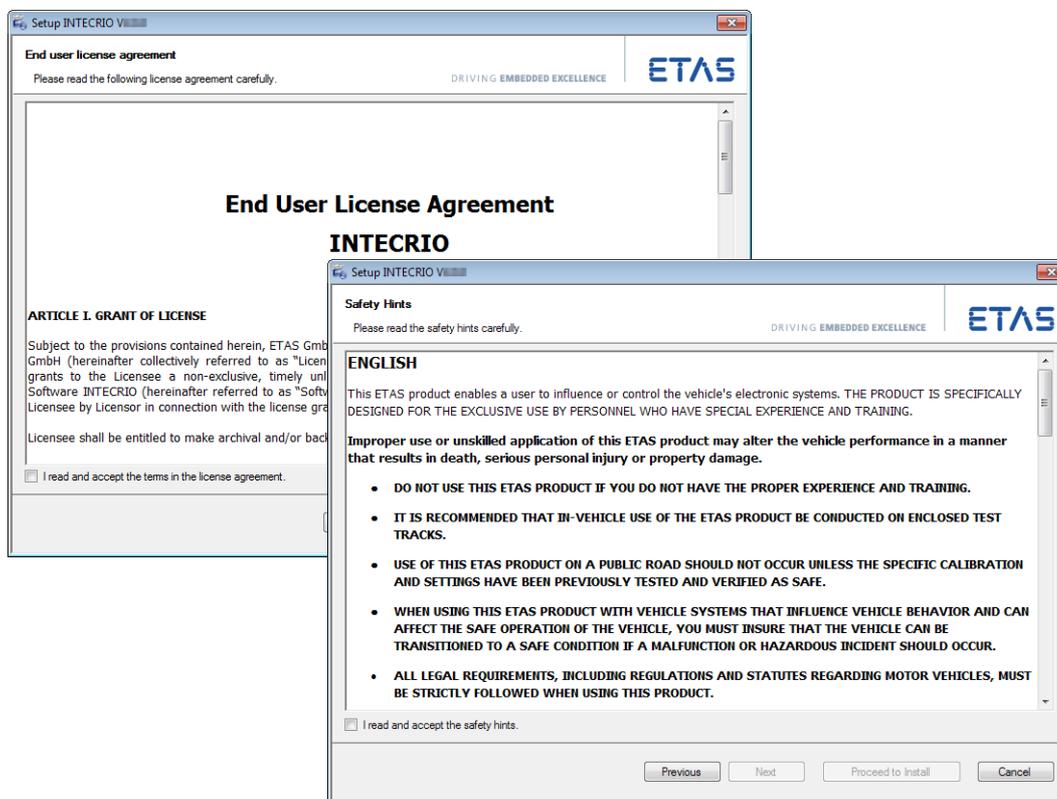
1. **Uninstall now** をクリックします。

現在インストールされている INTECRIO がアンインストールされ、INTECRIO V5.0.2 のインストールが行える状態になります。

2. アンインストールが完了したら、**Next** ボタンをクリックして次のページに進みます。

操作手順: ライセンスに関する合意文書と安全に関する注意事項を確認する

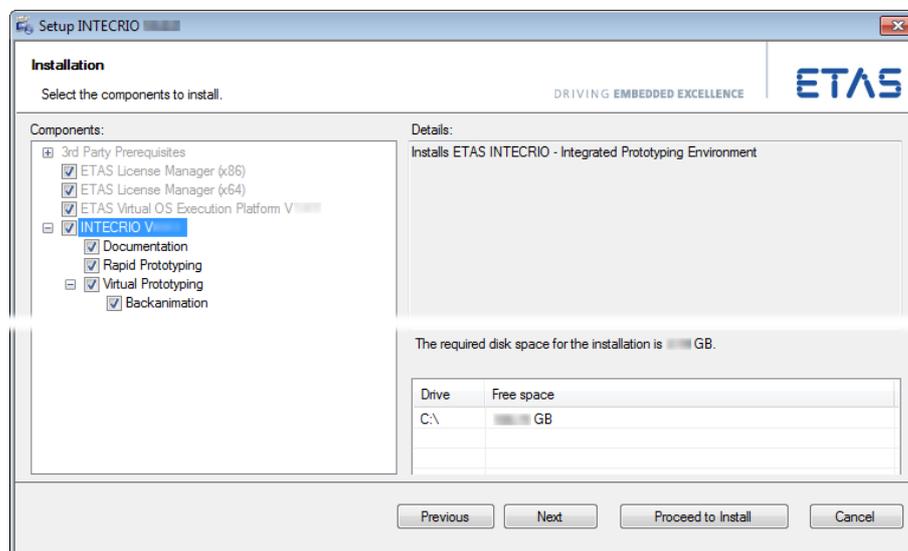
ライセンスに関する合意文書と安全に関する免責条項が、順に表示されます。



1. “License Agreement” ページに表示されるライセンスに関する合意文書の内容をよく読み、**I read and accept the items in the license agreement** チェックボックスをオンにして同意してください。
この文書に同意できない場合はインストールを続行できません。
2. **Next** をクリックします。
3. “Safety Hints” ページに表示される安全に関する免責条項をよく読み、**I read and accept the safety hints** オプションを選択してください。
免責条項に同意できない場合はインストールを続行できません。
4. 以下のいずれかを行います。
 - **Next** をクリックします。
次のページに進みます。
 - **Proceed to Install** をクリックして直接 “Ready to install” ページを開きます。この場合、インストールするコンポーネントの選択やインストール先のバス、ループバックネットワークアドレスの指定はスキップされ、デフォルト設定が適用されます。
23 ページの「INTECRIO をインストールする」に進みます。

操作手順：コンポーネントを選択する

次の "Installation" ページには、INTECRIO のコンポーネントや、インストールに必要なサードパーティ製コンポーネント、ETAS の各種ツール（ライセンスマネージャと ETAS 仮想 OS 実行プラットフォーム）が一覧表示されます。すでに PC にインストールされているコンポーネントについては、右側に "Already Installed" と表示されます。



ここでは、インストールするソフトウェアの機能範囲を指定しますが、選択や選択解除が行えるコンポーネントは、名前が黒字で表示されているものに限られます。

右側のテキストフィールドには、強調表示されているコンポーネントについての情報が表示されます。

1. インストールするコンポーネントを選択、または選択解除するには、それぞれ該当するオプションをアクティブ化、または非アクティブ化します。

注記

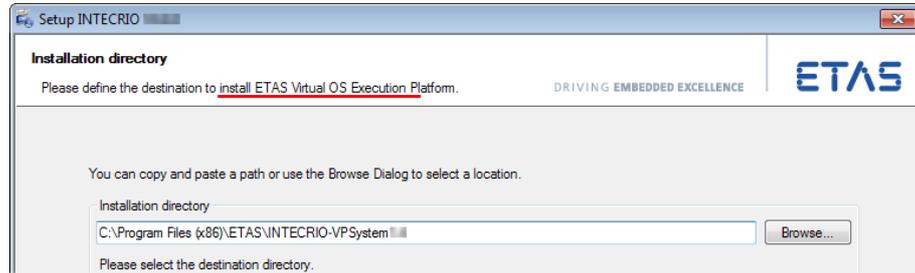
INTECRIO でハードウェアシステムを作成して設定するには、Rapid Prototyping と Virtual Prototyping のうち、少なくともいずれか一方のパッケージをインストールする必要があります。

INTECRIO で使用するモジュールを Simulink で作成するような場合は、Rapid Prototyping および Virtual Prototyping のいずれのパッケージも必要ありません。

2. 以下のいずれかを行います。
 - **Next** をクリックします。
次のページに進みます。
 - **Proceed to Install** をクリックして直接 "Ready to install" ページを開きます。この場合、インストール先のパス、ループバックネットワークアドレスの指定はスキップされ、デフォルト設定が適用されます。
23 ページの「INTECRIO をインストールする」に進みます。

操作手順：パスを設定する

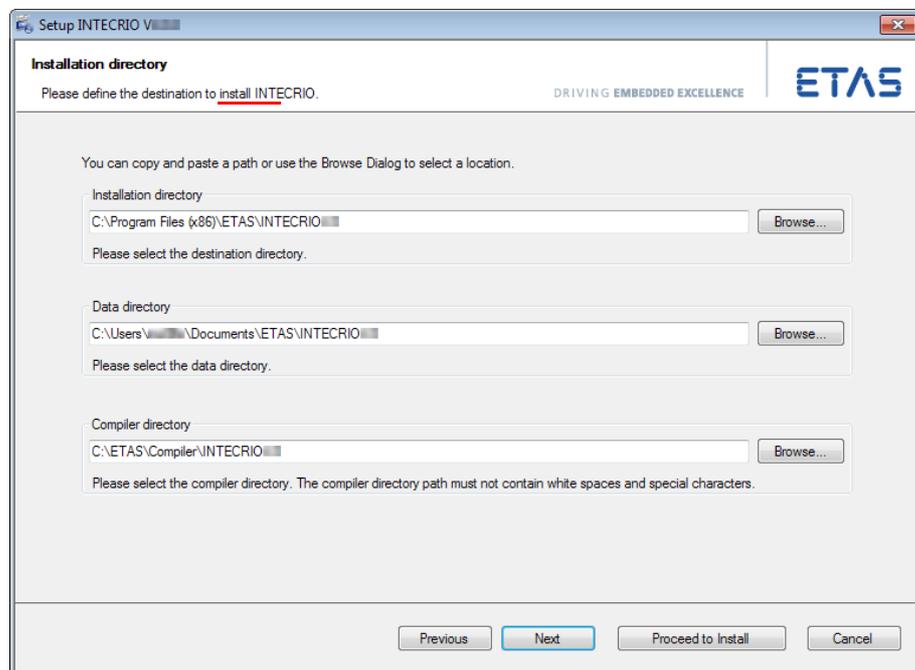
“Installation directly” ページの 1 ページ目では、ETAS 仮想 OS 実行プラットフォームをインストールするディレクトリを指定できます。

**i 注記**

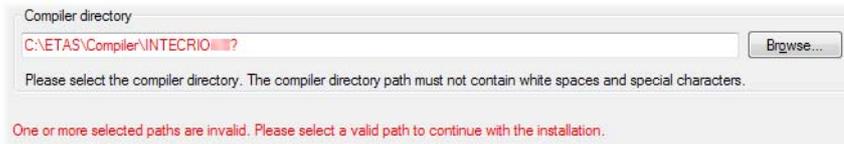
ETAS Virtual OS Extension Platform がすでにインストールされている場合、このページは開きません。

1. ディレクトリのデフォルト設定を変更するには、新しいパスを入力するか、または **Browse** ボタンで選択します。
無効なパスを指定した場合は、そのパスが赤字で表示され、警告メッセージ（22 ページ参照）が表示されます。
2. **Next** をクリックします。

“Installation directly” ページの 2 ページ目では、INTECRIO をインストールするディレクトリ (“<installation>” として参照されます) とサンプルファイルをインストールするディレクトリ (“<sample files>” として参照されます)、およびコンパイラをインストールするディレクトリを指定できます。



3. ディレクトリのデフォルト設定を変更するには、新しいパスを入力するか、または **Browse** ボタンで選択します。
- 無効なパスを指定した場合は、そのパスが赤字で表示され、警告メッセージが表示されます。



注記

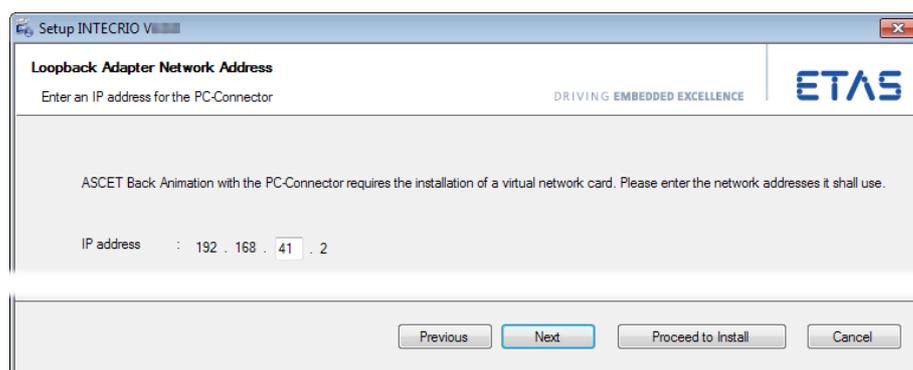
INTECRIO 内で使用されるコンパイラには空白文字を含むパスを使用できません。空白文字を含むコンパイラパスを指定すると、そのパスは無効なものとしてマークされます。

4. 以下のいずれかを行います。
- **Next** をクリックします。
次のページに進みます。
 - **Proceed to Install** をクリックして直接 “Ready to install” ページを開きます。この場合、ループバックネットワークアドレスの指定はスキップされ、デフォルト設定が適用されます。
23 ページの「INTECRIO をインストールする」に進みます。

操作手順: ループバックネットワークアドレスを入力する (仮想プロトタイピング用)

バックアニメーション部分の仮想プロトタイピングパッケージ (“Virtual Prototyping”) をインストールする場合は、仮想ネットワークボードが自動的にインストールされるので、このボードの IP アドレスを入力する必要があります。

1. “Loopback Adapter Network Address” ページで、IP アドレスを設定します。



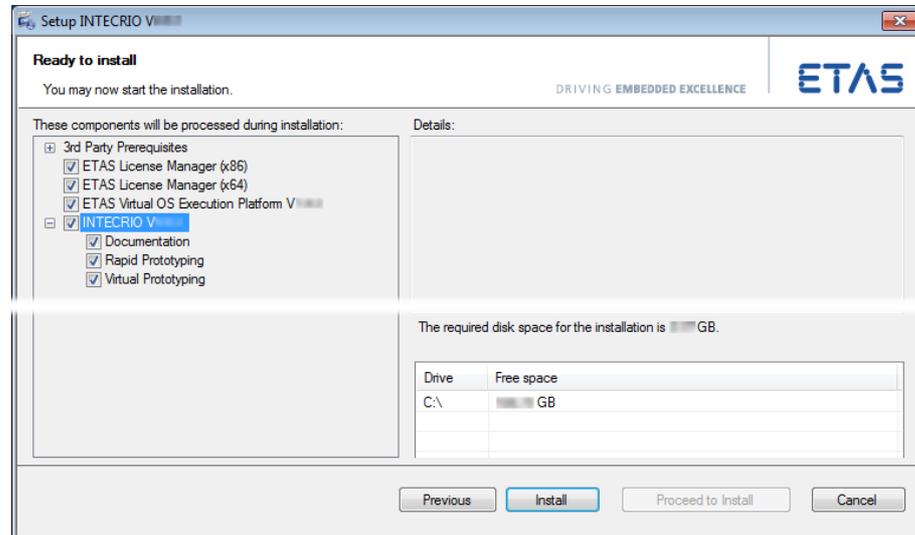
ここでは必ず、未使用のアドレスを指定してください。不明の場合はシステム管理者の方にお尋ねください。

入力されたアドレスが無効の場合、警告メッセージが表示されます。

2. **Next** または **Proceed to Install** をクリックします。

操作手順：INTECRIO をインストールする

次の "Ready to install" ページには、インストールされるコンポーネントが一覧表示されます。内容を変更するには、"Installation" ページ（20 ページの「コンポーネントを選択する」を参照）に戻る必要があります。



注記

次の操作によってインストール処理が実際に開始されます。開始後は **Cancel** ボタンが無効になり、インストールを中止することはできません。

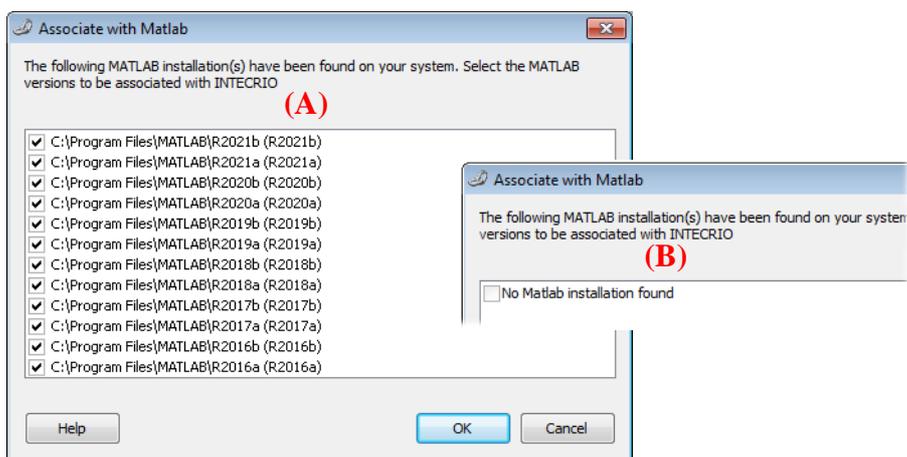
1. “Ready to Install” ページの **Install** をクリックします。

選択したコンポーネント（20 ページ参照）がインストールされます。すでにインストールされているコンポーネントはスキップされます。インストールの進行状況は進捗バーに示されます。

操作手順：INTECRIO と MATLAB/Simulink を関連付ける

インストールが終了すると、“Associate with Matlab” ダイアログボックスが開き、サポートされている MATLAB/Simulink のバージョン（R2016a ～ R2021b、および INTECRIO V5.0.2 のリリース時点でリリースされているそれらのサービスパック）のうち、PC 上にインストールされているものが表示される

ので（下図の (A)）、実際に使用するバージョンをここで選択します。有効なバージョンがインストールされていない場合も下図 (B) のように同じダイアログボックスが開きます。



注記

ネットワークドライブ上にインストールされた MATLAB/Simulink を使用する場合は、**Help** ボタンをクリックし、表示されるメッセージに従って操作してください。

1. “Associate with Matlab” ダイアログボックスで、INTECRIO に関連付けたい MATLAB/Simulink バージョンを選択します。
複数選択も可能で、1 つも選択しなくてもかまいません。
ここで **Cancel** をクリックすると、関連付けは行われずにダイアログボックスが閉じますが、MATLAB/Simulink 接続機能のインストールは行われます。
2. **OK** をクリックします。
選択された MATLAB/Simulink バージョンが INTECRIO に関連付けられます。
INTECRIO に関連付ける MATLAB/Simulink のバージョンは、後で変更することができます。詳しくは INTECRIO ユーザーガイドの「MATLAB®/Simulink® 接続機能」の項を参照してください。

操作手順: インストールを完了する

選択されたコンポーネントがすべてインストールされると、インストール処理を完了するためのキー入力が必要されます。

1. **Show readme when setup is closed** オプションをオンにします（推奨）。
2. **Finish** をクリックしてインストールを終了します。

Windows のスタートメニューのアプリ一覧に、以下のようなアイテムを含む **ETAS INTECRIO 5.0** フォルダが作成されます。

- **INTECRIO V5.0**
INTECRIO を起動します。

- **INTECRIO V5.0 - Tools** フォルダ
 - **Associate with ASCET**
ASCET でコード生成を行う際に、どのバージョンの INTECRIO を対象とするかを指定します。
 - **Associate with MATLAB**
インストールされている MATLAB/Simulink のバージョンを INTECRIO に関連付けます。詳しくは INTECRIO ユーザーガイドの「MATLAB®/Simulink® 接続機能」の項を参照してください。
この操作は、INTECRIO をインストールした後に MATLAB/Simulink のみを新たにインストールした際などに必要となります。
- **Manuals and Tutorials** フォルダ
ETAS のドキュメントディレクトリを開きます。ここには INTECRIO のドキュメントへのリンクが含まれています。

ETAS ライセンスマネージャを開くには、Windows スタートメニューのアプリケーションから **E > ETAS > ETAS License Manager** を選択します。

ETAS Virtual OS Execution Platform については、**E > ETAS > ETAS Virtual OS Execution Platform <x>.<y>** というフォルダに、以下のアイテムが作成されます。

- **Signal Configuration Editor V<x>.<y>**
これは INTECRIO では使用されません。

<installation> ディレクトリ下の LicenseTerms フォルダに、INTECRIO とともに配布されたオープンソース用ライセンス条件が保存されています。

PC のデスクトップには以下のアイコンが追加されます。



3.2.2 特殊なインストールの手順：仮想プロトタイピング（Virtual Prototyping）パッケージ

仮想プロトタイピング（バーチャルプロトタイピング）を行うには RTA-OSEK ツール（“RTA-OSEK Tools”）と PC 用 RTA-OSEK（“RTA-OSEK for PC”）が必要です。INTECRIO をインストールすると、仮想プロトタイピング用にこれらの製品もインストールされます。（20 ページの「コンポーネントを選択する」を参照してください）。

3.2.3 コマンドラインからのインストール

INTECRIO をコマンドラインからインストールすると、コマンドライン引数を使用してインストールをカスタマイズすることができます。

インストールプログラム Setup.exe を実行すると、そのたびに以下の場所にログファイル <date_time> Setup.log が作成されます。

- C:\ProgramData\ETAS\SETUP\Logs

以下に、Setup.exe で使用できるコマンドライン引数をご紹介します。

/silent

「サイレントモード」でインストールを行います。このモードにおいてはユーザーの介入を必要とするダイアログボックスが開きません。

 **注記**

サイレントインストールにおいて EULA と免責条項（19 ページ参照）に対して同意するには **/EULAAccepted** と **/SafetyHintsAccepted** を使用します。
PC の再起動が要求された場合の対応を指定するには、引数 **/NoRestart** または **/AllowRestart** を使用します。
インストール済みの旧バージョンの INTECRIO を自動的にアンインストールするには、引数 **/UninstallPreviousVersion** を使用します。

通常のインストールにおいてはダイアログボックスで入力される情報の代わりに、所定のデフォルト値が使用されます。エラーメッセージも表示されません。

/EULAAccepted

ライセンスに関する合意文書に同意します。

ライセンスに関する合意文書は、インストールディスクの EULA サブフォルダ内に収められています。

/SafetyHintsAccepted

安全に関する免責条項に同意します。

安全に関する免責条項の文書は、インストールディスクの SafetyHints サブフォルダ内に収められています。

/NoRestart

インストール終了後に要求される PC の再起動を省略します。実際に再起動が省略されると、ログメッセージが発行されます。

/AllowRestart

インストール終了後に要求される PC の再起動を許可します。再起動が行われる際には通知などは行われません。

/UninstallPreviousVersion

インストール済みの旧バージョンの INTECRIO を自動的にアンインストールします。

 **注記**

この引数を使用しない場合、INTECRIO の旧バージョンが PC にインストールされていると、`setup.exe` は強制終了します。

`/UninstallPreviousVersion` は `/Uninstall` と共には使用できません。

/Debug

*.msi パッケージのログファイルが、`C:\ProgramData\ETAS\SETUP\Logs` フォルダに作成されます。

/DefaultSettings

InstallationDefaultSettings.xml の代わりに独自の XML ファイルで以下のデフォルト設定を指定します。

変数	意味	参照
PRODINSTDIR	インストールディレクトリ	
PROUSERDOCUMENTSDIR	データディレクトリ	21 ページ
PRODCOMPILERDIR	コンパイラディレクトリ	
LOOPBACK_IP_ADDRESS	ループバックアダプタの IP アドレス	22 ページ

ファイルが Setup.exe と相対位置にある場合は、相対パスで指定できます。それ以外の場合は絶対パスで指定します。

構文： /DefaultSettings:"<path>¥<filename>.xml"

/Uninstall

インストールされている INTECRIO をアンインストールします。
/silent と共に使用することにより、ユーザーの介入なしにアンインストールを実行することができます。

/Repair

/silent と共に使用すると、修復処理を開始することができます。それ以外の場合はメンテナンスモードが選択されます。

 **注記**

インストールロケーションに setup.exe ファイルがある場合は、/Uninstall と /Repair は使用できません。これらの引数を使用する場合は、setup.exe を以下の場所に置いて実行する必要があります。

C:¥<programs>¥ETAS¥GENericSetup¥IPE INTECRIO-IP¥5.0.<x>.<y>

<programs> = Program Files (x86) (64 ビット OS)、または Program Files (32 ビット OS)

例：

Setup.exe /silent /EULAAccepted /SafetyHintsAccepted
サイレントインストールを開始します。

"C:¥Program Files (x86)¥ETAS¥GENericSetup¥IPE
INTECRIO-IP¥5.0.<x>.<y>¥setup.exe" /uninstall /Debug
サイレントモードではなく通常モードでアンインストールを開始し、詳細なログを書き込みます。

3.3 ETAS Experiment Environment (ETAS 実験環境) のインストール

3.3.1 Experiment Environment のシステム要件



注記

必ず、使用する製品バージョンのリリースノートに記載された最新情報を確認してください。

	必要最低条件	推奨条件
ハードウェア	<ul style="list-style-type: none"> PC: 1.5 GHz RAM: 2 GB DVD-ROM ドライブ イーサネットアダプタ: 10/100BaseT グラフィックアダプタ: 32 MB RAM、16 ビットカラー、DirectX 7 画面解像度: 1024x768 	<ul style="list-style-type: none"> PC: 2.0 GHz Multicore RAM: 8 GB DVD-ROM ドライブ 第2のイーサネットアダプタ: 10/100BaseT グラフィックアダプタ: 256 MB RAM、32 ビットカラー、DirectX 7 以上、HW アクセラレーション 画面解像度 1600x1200
OS	Windows® 8.1 (x64)、Windows® 10 (x64)	Windows® 10® (x64)
ディスク空き容量	1.8 G	2 GB 以上

3.3.2 初回のインストール

INTECRIO がインストールされていない場合、ETAS Experiment Environment を使用して既存のプロジェクトの実験を行うことができますが、プロジェクトを作成することはできません。

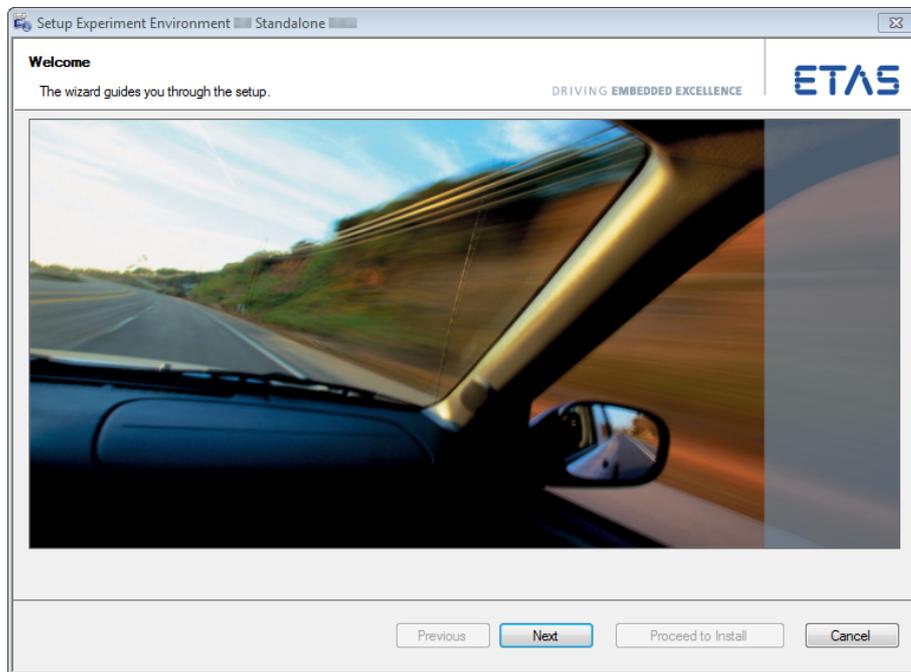
非対応のオペレーティングシステムに INTECRIO をインストールしようとする、エラーメッセージが表示され、インストール処理が強制終了します。

ETAS Experiment Environment は、システムドライブの Program Files (x86) \Common Files にインストールされ、インストール先を変更することはできません。

操作手順: ETAS Experiment Environment のインストーラを起動する

1. インストールディスクを PC の所定のドライブに挿入します。
2. 以下のいずれかを行います。
 - インストーラのウィンドウから **"Main"** リンクをクリックし、さらに [Experiment Environment](#) をクリックします。

- Windows Explorer で DVD の ExperimentEnvironment フォルダを開き、setup.exe ファイルを実行します。
ETAS インストーラが起動します。

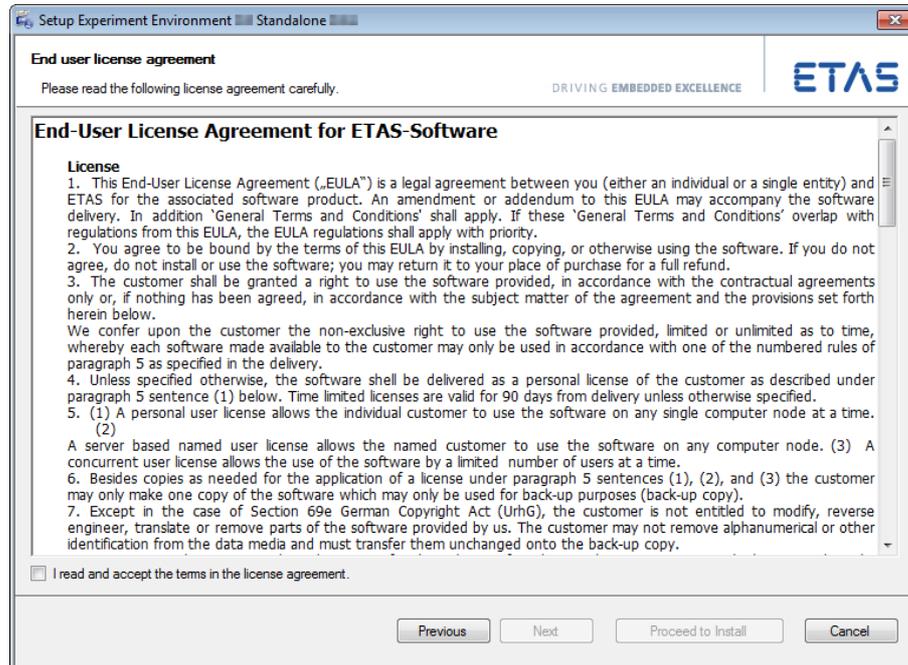


3. **Next** ボタンをクリックします。

インストーラは、PC が ETAS Experiment Environment のインストール要件を満たしているかをチェックし、その結果を “System Check” ウィンドウに表示します。
要件を満たしている場合は、自動的に次のページに進みます。

操作手順：ライセンスに関する合意

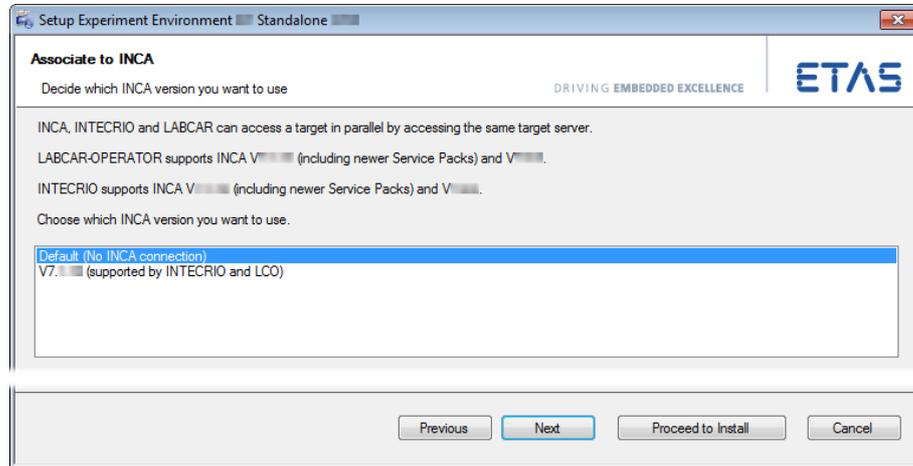
ライセンスに関する合意文書が表示されます。



1. ライセンスに関する合意文書の内容をよく読み、**I read and accept the terms in the license agreement** チェックボックスをオンにして同意してください。
この文書に同意できない場合はインストールを続行できません。
 2. 以下のいずれかを行います。
 - **Next** をクリックします。
次のページに進みます。
 - **Proceed to Install** をクリックして直接 “Ready to install” ページを開きます。この場合、インストールするコンポーネントの選択はスキップされ、デフォルト設定が適用されます。
- 31 ページの「ETAS Experiment Environment をインストールする」に進みます。

操作手順：INCA バージョンの選択：

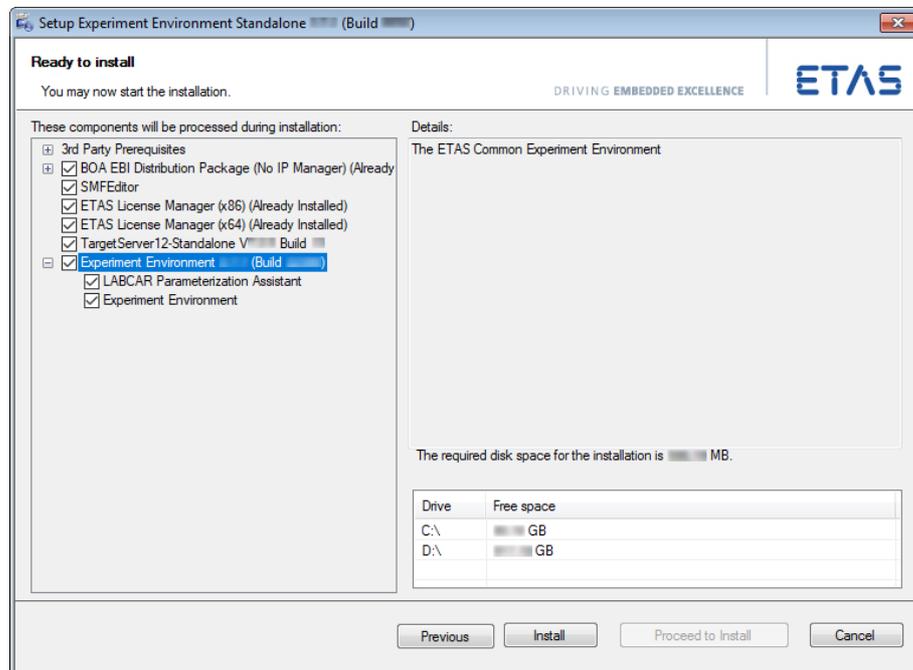
"Associate to INCA" ページには、PC にインストールされている INCA のバージョンのうち、ETAS Experiment Environment と同じターゲットサーバーにアクセスするものが一覧表示され、Default (No INCA connection) が選択された状態になっています。



1. ETAS Experiment Environment に関連付けたい INCA のバージョンを選択します。
2. **Next** または **Proceed to Install** をクリックします。

操作手順：ETAS Experiment Environment をインストールする

次の "Ready to install" ページには、インストールされるコンポーネントが一覧表示されます。



 **注記**

次の操作によってインストール処理が実際に開始されます。開始後は **Cancel** ボタンが無効になり、インストールを中止することはできません。

1. “Ready to Install” ページの **Install** をクリックします。
各コンポーネントがインストールされます。インストールの進行状況は進捗バーに示されます。

操作手順: インストールを完了する

選択されたコンポーネントがすべてインストールされると、インストール処理を完了するためのキー入力が必要されます。

1. **Finish** をクリックしてインストールを完了します。

Windows スタートメニューのアプリ一覧に、以下のアイテムを含むフォルダ **E > ETAS Experiment Environment <x>.<y>** (<x>.<y> はバージョン番号) が作成されます。

- **Experiment Environment V<x>.<y>**
ETAS Experiment Environment を起動します。
- **Experiment Environment <x>.<y> Tools** フォルダ
以下のショートカットを含むフォルダが開きます。
 - **Associate to INCA**
ETAS Experiment Environment と共に使用する INCA のバージョン (および共有モジュールのバージョン) を指定します。
 - **Associate to RTA-TRACE**
RTA-TRACE のインストール状況を調べます。問題がある場合はその内容が表示されます。
- **INCA <a> . Target Server - Tools**
以下のショートカットを含むウィンドウを開きます。
 - **ETAS Network settings**
ETAS ネットワークの設定ツールを起動します。
 - **Search for connected hardware**
ターゲットサーバーが起動し、PC に接続されているハードウェアが検索されます。
- **Manuals and Tutorials** フォルダ
ETAS ドキュメントフォルダを開きます。ここに ETAS Experiment Environment のユーザードキュメントが保存されています。
 - **Instrument Programming**
ユーザー定義のインストゥルメントをプログラミングして ETAS Experiment Environment に組み込む際に必要となる情報が記載されたオンラインヘルプが開きます。
 - **Online Help**
ETAS Experiment Environment のオンラインヘルプを開きます。
 - **Scripting API**
ETAS Experiment Environment のスクリプト用 API についてのオンラインヘルプを開きます。

PC のデスクトップには以下のアイコンが追加されます。



さらに、同じ ETAS プログラムグループ内に SMF エディタの専用フォルダ **SMFEditor <x>.<y>** が作成され、以下のアイテムが格納されます。この SMF エディタは LABCAR 関連のツールで、INTECRIO とともに使用するものではありません。

- **SuT Mapping File Editor**
SMF エディタを開きます。
- **SMFE Online Help**
SMF エディタのオンラインヘルプを開きます。

3.3.3 コマンドラインからのインストール

ETAS Experiment Environment をコマンドラインからインストールには、以下のようなコマンドライン引数を使用してインストールをカスタマイズすることができます。

Setup.exe が実行されるたびに以下のログファイルが作成されます。

```
%appdata%1 ¥ETAS¥SETUP¥<date_time> Setup.log
```

/silent

「サイレントモード」でインストールを行います。このモードにおいてはユーザーの介入を必要とするダイアログボックスが開きません。

注記

サイレントインストールにおいて EULA と免責条項（30 ページ参照）に対して同意するには **/EULAAccepted** と **/SafetyHintsAccepted** を使用します。

PC の再起動が要求された場合の対応を指定するには、引数 **/NoRestart** または **/AllowRestart** を使用します。

インストール済みの旧バージョンの ETAS Experiment Environment を自動的にアンインストールするには、引数 **/UninstallPreviousVersion** を使用します。

通常のインストールにおいてはダイアログボックスで入力される情報の代わりに、所定のデフォルト値が使用されます。エラーメッセージも表示されません。

/EULAAccepted

ライセンスに関する合意文書に同意します。

ライセンスに関する合意文書は、インストールディスクの ExperimentEnvironment¥EULA サブフォルダ内に収められています。

1. %appdata% =
C:¥Users¥<username>¥AppData¥Roaming

/SafetyHintsAccepted

安全に関する免責条項に同意します。

安全に関する免責条項の文書は、インストールディスクの ExperimentEnvironment¥SafetyHints サブフォルダ内に収められています。

/NoRestart

インストール終了後に要求される PC の再起動を省略します。実際に再起動が省略されると、ログメッセージが発行されます。

/AllowRestart

インストール終了後に要求される PC の再起動を許可します。再起動が行われる際には通知などは行われません。

/UninstallPreviousVersion

インストール済みの旧バージョンの INTECRIO を自動的にアンインストールします。

**注記**

この引数を使用しない場合、INTECRIO の旧バージョンが PC にインストールされていると、setup.exe は強制終了します。

/UninstallPreviousVersion は /Uninstall と共には使用できません。

/Debug

*.msi パッケージのログファイルを書き込みます。

ファイルは %appdata%¹ ¥ETAS¥SETUP フォルダに書き込まれます。

/DefaultSettings

InstallationDefaultSettings.xml の代わりに独自の XML ファイルでデフォルト設定を指定します。

ファイルが² setup.exe と相対位置にある場合は、相対パスで指定できます。それ以外の場合は絶対パスで指定します。

構文： /DefaultSettings:"<path>¥<filename>.xml"

/Uninstall

インストールされている ETAS Experiment Environment をアンインストールします。/silent と共に使用することにより、ユーザーの介入なしにアンインストールを実行することができます。

C:¥<programs>² ¥ETAS¥GENERICSetup¥EE EE Standalone¥3.7.<x>.<y> フォルダに収められている setup.exe ファイルを使用します。

例：

Setup.exe /silent /EULAAccepted /SafetyHintsAccepted
サイレントインストールを開始します。

-
1. %appdata% =
C:¥Users¥<username>¥AppData¥Roaming
 2. <programs> = Program Files (x86) (64 ビット OS)、または Program Files (32 ビット OS)

```
"C:¥P<programs>1¥ETAS¥GENERICSetup¥EE EE Standalone¥
  3.7.<x>.<y>¥setup.exe" /uninstall /Debug ↓
  サイレントモードではなく通常モードでアンインストールを開始し、詳細
  なログを書き込みます。

setup.exe /DefaultSettings:"D:¥myOwnSettings.xml"
  サイレントモードではなく通常モードでインストールを開始し、カスタマ
  イズされたデフォルト設定でインストールを実行します。
```

3.4 ライセンス管理に関する設定

INTECRIO をインストールすると、同時に ETAS ライセンスマネージャもインストールされます。*.ini ファイルの [Licensing] セクションを編集することにより、INTECRIO（および他の ETAS ソフトウェアプログラム）がライセンスにアクセスする方法を指定することができます。

インストールディスクには、このような *.ini ファイルが 2 つ含まれていて、1 つが INTECRIO 用、もう 1 つが ETAS Experiment Environment 用です。ファイル名はどちらも Licensing.ini です。

INTECRIO 用の Licensing.ini の場所は InstallationDefaultSettings.xml ファイルにおいて、INTECRIO パッケージ License の LIMA_INIFILE 変数で指定されています。

ETAS Experiment Environment 用の Licensing.ini は、ExperimentEnvironment¥Packages¥EE License_3.7.<x>.<y> フォルダに格納されています。

一般的な手順

Licensing.ini ファイルを編集してインストールを開始する手順は、インストール用ファイルが置かれている場所に応じて異なります。

(この手順を行うユーザーが) 編集できないディレクトリにインストール用ファイルが置かれている場合：

以下のいずれかの手順で行います。

- すべてのインストール用ファイルを、編集可能なディレクトリにコピーします。
その場所で Licensing.ini を編集し（36 ページ参照）、Autostart.exe または setup.exe（3.2.1 項と 3.3.2 項を参照）をダブルクリックするか、またはコマンドラインから（3.2.3 項と 3.3.3 項を参照）インストールを開始します。
- ユーザーが編集できるディレクトリに、Licensing.ini ファイルのみをコピーします。
そこで Licensing.ini ファイルを編集します（36 ページ参照）。さらに独自の XML ファイルを作成し、デフォルト設定（34 ページ参照）を定義し、編集した Licensing.ini ファイルの場所を LIMA_INIFILE 変数で指定します。

1. <programs> = Program Files (x86) (64 ビット OS)、または Program Files (32 ビット OS)

```
<Variable                                     と
  name="LIMA_INIFILE"                          と
  defaultValue="<path>¥Licensing.ini"         と
  validation="fileExists" />
```

コマンドラインからインストールを開始し（3.2.3 項と 3.3.3 項を参照）、その際に以下の引数を使用して、独自の、Licensing.ini ファイルの場所を指定した上記の XML ファイルを指定します。

```
/licconf="<absolute path>¥<filename>.ini"
```

（この手順を行うユーザーが）編集できるディレクトリにインストール用ファイルが置かれている場合：

Licensing.ini を編集し（36 ページ参照）、インストールを開始します。

***.ini ファイルの編集**

以下に Licensing.ini の編集方法を説明します。

操作手順：ライセンスへのアクセス方法を指定する

1. テキストエディタで .ini ファイルを開きます。
2. 要件にあわせて [Licensing] セクションを編集します。
このセクションで使用できるパラメータとその設定については以下を参照してください。
3. 変更内容を保存します。

[Licensing] セクションで使用できるパラメータ：

- **LicenseFileName**
追加するライセンスファイルの完全なパスと名前
- **LicensesToBorrow**
ライセンスサーバーからライセンスを借用できる場合、このパラメータを使用して借用メカニズムについて設定することができます。借用メカニズムを有効にするには、ここに製品または機能のライセンス名（例：INTECRIO）を指定します。2 つ以上のライセンスを指定するには、各ライセンス名をスペースで区切って入力してください。
INTECRIO と ETAS Experiment Environment では以下のライセンスが使用されます。

ライセンス名	機能
INTECRIO	INTECRIO プロトタイピング環境 ^a
INT_ASC	ASCET 接続機能 ^b
INT_ECC	MATLAB/Simulink Embedded Coder 接続機能 ^a
INT_RTA-RTE2.0	AUTOSAR-RTE - プロトタイピング用 ^c
INT_SLC	MATLAB/Simulink 接続機能 ^a
INT_TC_RTT	RTA-Trace 接続機能 ^{a, d}
INT_UCC_ES900	ES900 接続機能 ^e
INT_UCC_PC	VP-PC 接続機能 ^f
INT-VIP	ETAS 仮想 OS 実行プラットフォーム ^e
LD_RTA-OS_VRTA_SC34	仮想プロトタイピング用 RTA-OSEK バリエーション ^e

ライセンス名	機能
EE	ETAS Experiment Environment (ETAS 実験環境)
INT_RP_EE	INTECRIO と ETAS Experiment Environment によるラピッドプロトタイピング
INT_VP_EE	INTECRIO と ETAS Experiment Environment による仮想プロトタイピング
INT_UCC_LCO_FIL, LCS_LCO_CCI, LCS_LCO_LCE, LCS_LCO_LCX	ETAS Experiment Environment と INTECRIO との組み合わせにおいては使用されません。

- INTECRIO 統合プラットフォームに同梱
- ASCET に同梱
- RTA-RTE (INTECRIO プラットフォームに同梱)
- RTA-TRACE は 廃止となる製品です。このライセンスは、すでにインストールされている製品用のものです。
- INTECRIO-VP (仮想プロトタイピング) の一部
- INTECRIO-RP (ラピッドプロトタイピング) の一部

- **BorrowExpiryMode**

借用期限の指定方法を選択します。

- **Date**

借用期間は、`BorrowExpiryDate` で指定された日付に満了します。

- **Interval**

借用期間は、`BorrowExpiryInterval` で指定された日数を経過した時点で満了します。

- **BorrowExpiryDate**

BorrowExpiryMode が **Date** に設定されている場合、このパラメータで借用期間の終了日を指定します。フォーマットは `yyyy-mm-dd` です。

- **BorrowExpiryInterval**

BorrowExpiryMode が **Interval** に設定されている場合、このパラメータで借用日数を指定します。

- **ExecuteBorrowAutomaticExtensionInterval**

借用期間を自動延長するタイミングを指定します。借用期間の残日数がこのパラメータで設定された日数になると、

BorrowAutomaticExtensionInterval で指定された日数だけ自動的に期間が延長されます。

- **BorrowAutomaticExtensionInterval**

借用期間を自動延長する日数を指定します。

- **ImmediateBorrow**

ライセンスの自動借用を行うかどうかを指定します。

- **True**

インストール時に自動的にライセンスの借用が行われます。

 **注記**

ImmediateBorrow='True' は、インストールを行うユーザーに対してのみ有効です。借用されたライセンスは、同じ PC を使用する他のユーザーには使用できません。

- **False**

プログラムが初めてライセンスサーバーに接続した際にライセンスの借用が行われます。

• **CustomLicenseFolder**

追加されるライセンスファイルのデフォルトの格納場所（例：Windows 7 の場合は C:\ProgramData\ETAS\FlexNet）に管理者権限を持つユーザーしか書き込みができない場合、このパラメータで別のパスを指定することができます。

以下の例のように設定すると、INTECRIO についてライセンス借用が有効になります。INTECRIO の初回起動時にライセンスの借用が行われ、デフォルト状態においては 100 日後に借用期間が終了します。

[Licensing]

LicenseFileName = 'd:\licenses\MyLicense.lic'

LicensesToBorrow = 'INTECRIO'

BorrowExpiryMode = 'Interval'

BorrowExpiryInterval = '100'

ImmediateBorrow = 'false'

3.5 ソフトウェアのライセンスを取得する

INTECRIO の使用には有効なライセンスが必要です。ライセンス取得に必要なライセンスファイルは、ツールの管理者、もしくは ETAS のインターネットサイトにあるセルフサービスポータル (<https://www.etas.com/support/licensing>) から取得できます。ライセンスファイルの要求には、注文時に ETAS から配布されたアクティベーション番号の入力が必要です。

Windows スタートメニューのアプリ一覧から **E > ETAS > ETAS License Manager** を選択し、ダイアログに表示される指示に従ってください。ETAS のライセンスモデルについての説明や、ライセンスの借用などの詳しい説明については、ライセンスマネージャで、**<F1>** キーを押して情報を参照してください。

3.6 アンインストールする

INTECRIO をアンインストールすると、インストールされているすべてのコンポーネントとアドオン（ターゲットサーバーと ETAS Experiment Environment を除く）が自動的にアンインストールされます。

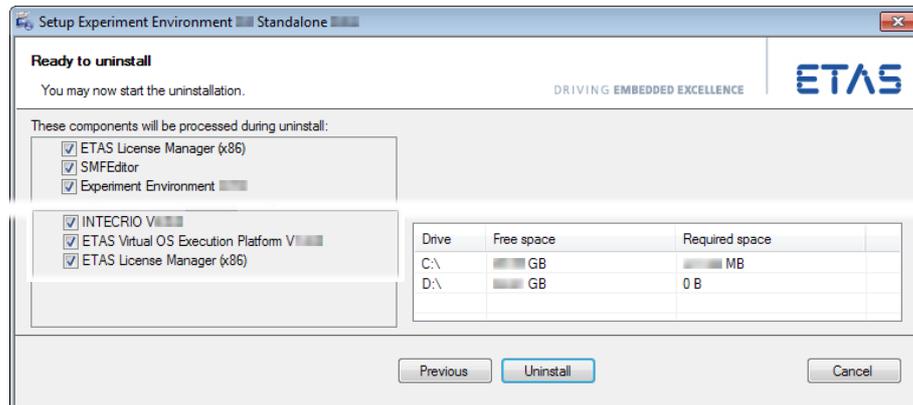
INTECRIO または ETAS Experiment Environment のアンインストールは、以下のいずれかの方法で開始します。

- 1 Windows コントロールパネルの **プログラムの追加と削除** または **プログラムと機能** を使用する

2. コマンドライン引数として /Uninstall を使用する (27 ページと 34 ページを参照)

操作手順: INTECRIO または ETAS Experiment Environment をアンインストールする

1. 上記のいずれかの方法でアンインストールを開始します。
ETAS アンインストーラが起動します。
2. **Next** ボタンをクリックして次のページに進みます。
前のページに戻るには **Previous**、インストールを中止するには **Cancel** をクリックします。
PC がアンインストール要件を満たしているかがチェックされ、その結果が "System Check" ページに表示されます。
PC がアンインストール要件を満たしているかがチェックされ、その結果が "System Check" ページに表示されます。
要件が満たされていると、"Ready to uninstall" ページが開き、アンインストールされるコンポーネントが一覧表示されます。
この内容は変更できません。

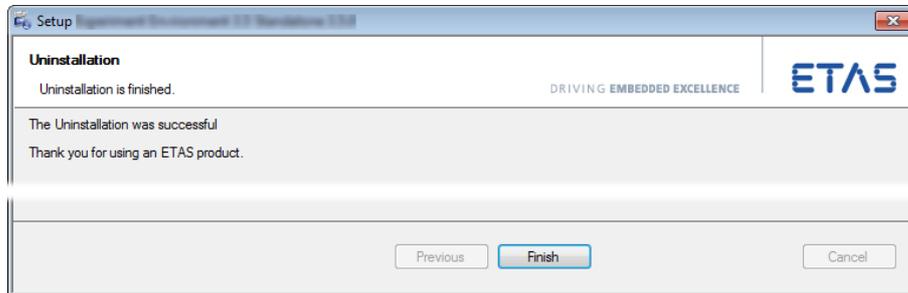


ETAS 仮想 OS 実行プラットフォームとライセンスマネージャのアンインストールは、他のプログラムによって使用されていない場合にのみ可能です。

注記

次の操作によってアンインストール処理が実際に開始されます。開始後は **Cancel** ボタンが無効になり、アンインストールを中止することはできません。

3. **Uninstall** をクリックしてアンインストールを実行します。
処理の進行状況は進捗バーに表示されます。
すべてのコンポーネントのアンインストールが終了すると、以下のページが開きます。



4. **Finish** をクリックしてインストールを終了します。

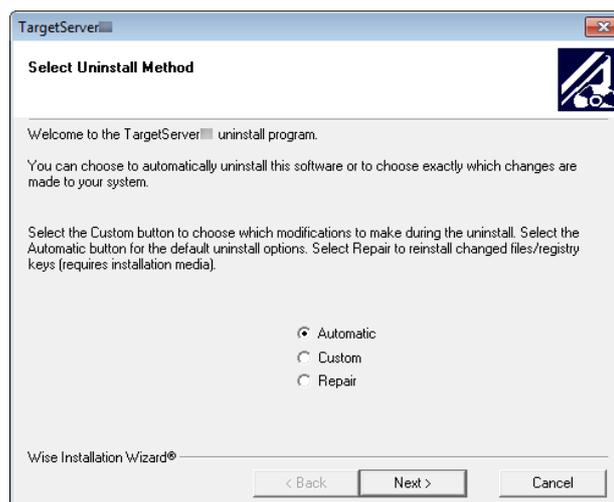
操作手順: ターゲットサーバーをアンインストールする

ご注意ください!

ターゲットサーバーのみをアンインストールすると、ターゲットへのダウンロードが実行できなくなります。この場合、INTECRIOとETAS Experiment Environmentを起動することはできませんが、エラーメッセージが発行されません。

1. Windows コントロールパネルの **プログラムと機能** を使用して、ETAS Target Access - <x>.<y> のアンインストールを開始します。
2. 安全に関する質問が表示されたら、メッセージを確認して **Yes** をクリックします。

Windows インストーラが開きます。



3. 表示されている注意事項をよく読み、Automatic または Custom を選択します。

4. **Next** をクリックします。
Automatic を選択した場合は、ターゲットサーバーがアンインストールされます。
Custom を選択した場合は、表示されるメッセージに従って操作を行います。
5. アンインストールを中止するには、Windows インストーラの **Cancel** ボタンをクリックします。
アンインストールが中止され、削除されたファイルが復元されます。

4 INTECRIO クイックガイド

4.1 はじめに

本章は、INTECRIO V5.0 の主要な機能をご理解いただくための「クイックガイド」です。INTECRIO で ECU ソフトウェアのプロトタイプ開発を行うためのさまざまな作業とその操作方法の概要を紹介します。開発手順に沿って、各作業のおおまかな手順をフローチャートで示し、その右側に操作方法を図示しています。図中の黒い矢印の付いた破線は、操作とその結果を示し、赤い矢印の付いた実線は、操作を行うボタンやオプションなどの場所を示しています。

以下に、本章で用いられている主要な略語について簡単に説明します。

- **BMT** – **Behavioral Modeling Tool** (挙動モデリングツール)
- **EE** – **ETAS Experiment Environment** (ETAS 実験環境)
- **ES** – **Environment System** (環境システム)
- **HBB** – **Hook-based bypass** (フックベースバイパス)
- **ISR** – **Interrupt Service Routine** (割り込みサービスルーチン)
- **HC** – **Hardware Configurator** (ハードウェアコンフィギュレータ)
- **HW** – **Hardware** (ハードウェア)
- **HWS** – **Hardware System** (ハードウェアシステム)
- **PC** – **Project Configurator** (プロジェクトコンフィギュレータ)
- **PI** – **Project Integrator** (プロジェクトインテグレータ)
- **OSC** – **OS Configurator** (OS コンフィギュレータ)
- **RE** – **Runnable Entity** (ランナブルエンティティ、「ランナブル」とも呼ばれます)
- **RP** – **Rapid Prototyping** (ラピッドプロトタイピング)
- **SBB** – **Service-based bypass** (サービスベースバイパス)
- **SP** – **System Project** (システムプロジェクト)
- **SWC** – 「レガシー AUTOSAR モジュール」、つまり INTECRIO ワークスペースにインポート済みの **AUTOSAR Software Component** (AUTOSAR ソフトウェアコンポーネント)
- **SWS** – **Software System** (ソフトウェアシステム)
- **VP** – **Virtual Prototyping** (仮想プロトタイピング)
- **WS** – **Workspace** (ワークスペース)

4.2 新しいワークスペースの作成

ワークスペース (WS) の主な役割は、INTECRIO での作業時に作成されたデータ (モジュール、ファンクション、ハードウェア/ソフトウェアシステム、システムプロジェクト、実験) をデータベース形式で体系的に格納し、各データを明解なインターフェースで管理することです。ワークスペースにおいては、Windows エクスプローラと同様に、フォルダやアイテムの新規作成、移動、コピー、インポート/エクスポートが行え、またワークスペース自体を新規作成することもできます。

以下のデータはワークスペース内で管理されます。

- ファンクション
- ハードウェア/ソフトウェアシステム
- システムプロジェクト

以下のデータはワークスペース内では管理されません。

- *.c、*.h、*.a21 ファイル
- *.a21.cod ファイル
- 測定ファイル (*.dat)

INTECRIO では、複数のワークスペースを使用することができます。ワークスペースのデータ量が極端に大きくなると、システムパフォーマンスが低下してしまう可能性があるため、作業ごとに個別のワークスペースを作成することをお勧めします。これによって1つの作業で使用されるデータ量を小さく抑え、扱いやすくすることができます。

一般的に、新しい作業を開始する際は、まず新しいワークスペースを作成し、そこに必要なデータを割り当てて具体的なプロジェクトを構成していきます。これにより、必要なデータのみで構成された簡潔な作業環境を実現できます。

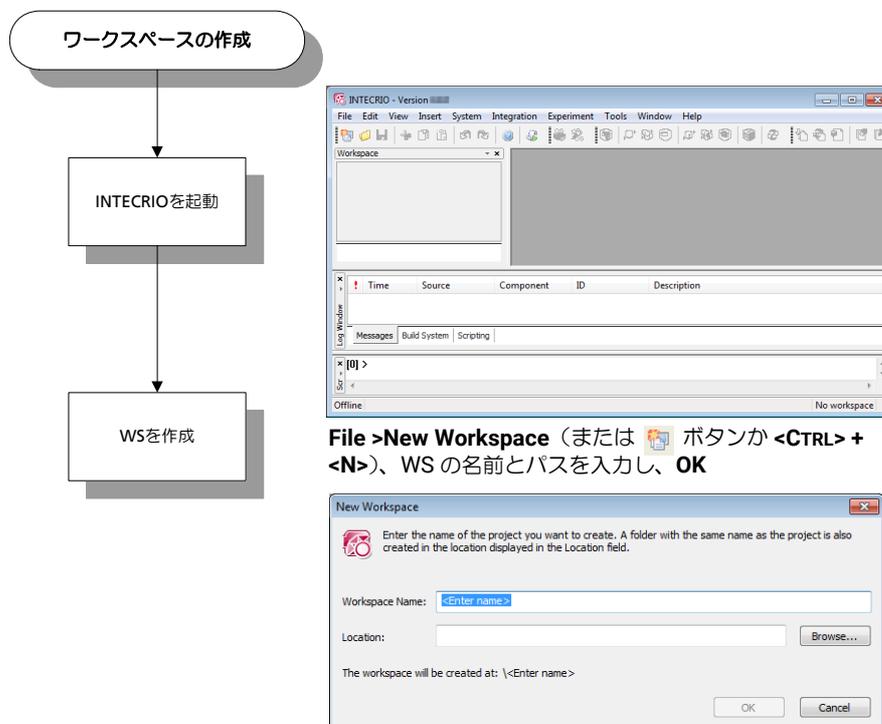


図 4-1 ワークスペースを作成する

4.3 モジュールの準備とインポート

INTECRIO で実験を行うには、実際に機能するソフトウェアモジュールが必要です。各種 BMT を用いて作成されたモジュールをインポートして INTECRIO 上で使用できるようにするには、各モジュールを各種記述ファイル（*.c、*.a21、*.six）の形式にエクスポートしたものがが必要です。

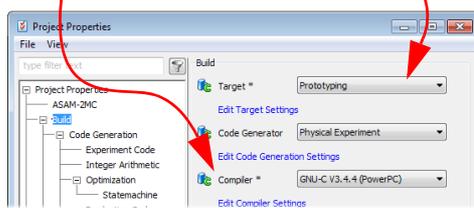
ワークスペース内において、モジュールは */Modules フォルダに格納されます。

4.3.1 モジュールの準備

Simulink® の場合：
 make ファイル：
 irt_default_tmf または ier_default_tmf
 ターゲット：
 irt.tlc または ier.tlc

Simulink® の場合：
**Tools > C/C++ Code > Build Model, <CTRL> + **, ボタンのいずれか

ASCET の場合：
 プロジェクトエディタ：
 ターゲット ES910/Prototyping、
 コンパイラ Gnu-C V3.4.4
 (PowerPC)/QCC V6.5 を選択



Build > Transfer、または [Transfer] ボタン

INTECRIO 実験を選択

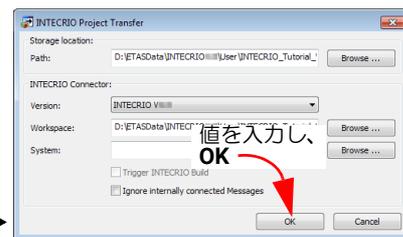


図 4-2 モジュールを作成する

4.3.2 モジュールのインポート

モジュールのインポート時には、インターフェース記述ファイル (*.six) のみが読み込まれます（『INTECRIO ユーザーガイド』の「SCOOP と SCOOP-IX」の項を参照してください）。

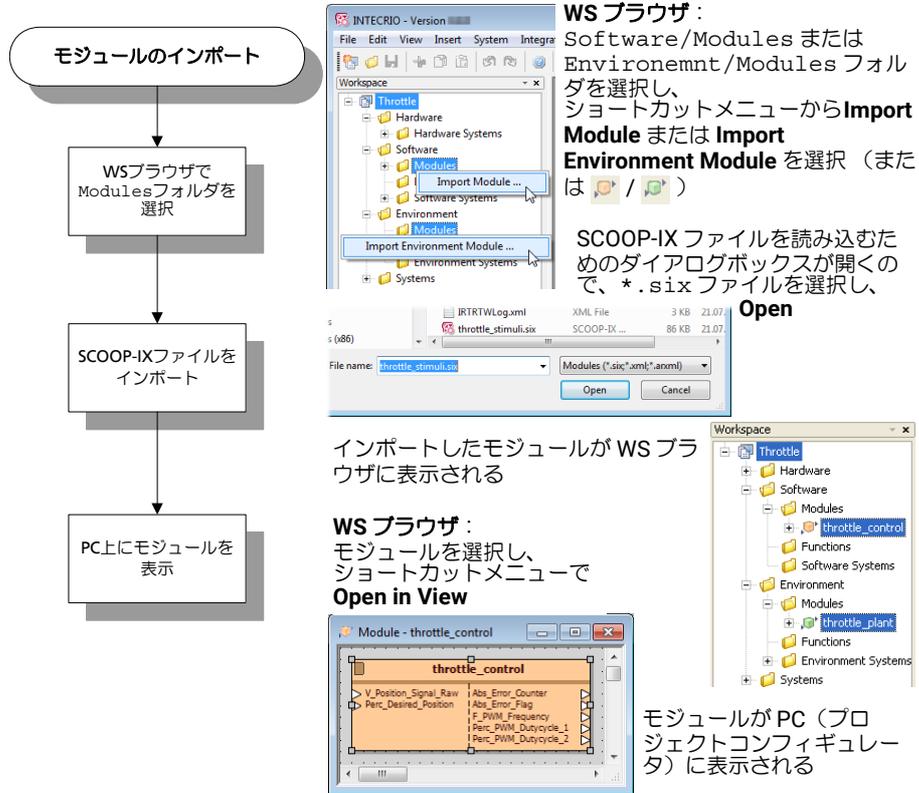


図 4-3 モジュールをインポートする

i 注記

INTECRIO V5.0.2 における AUTOSAR のサポートは限定的なものになります。「レガシー AUTOSAR モジュール」を含む既存のワークスペースはサポートされますが、AUTOSAR ソフトウェアコンポーネントを新たにインポートすることはできません。

4.4 ファンクションの作成

関連する複数のモジュールや「レガシー SWC」をまとめて1つの「ファンクション」に割り当てることにより、モデルがより明確になり、再利用しやすくなります。ファンクションはモジュールの構造化を行うためのアイテムで、それ自体の機能はありません。ファンクションに含めることができるのは、モジュールまたは SWC のみで、両者の混在も可能です。ワークスペース内において、ファンクションは Software/Functions フォルダに格納されます。

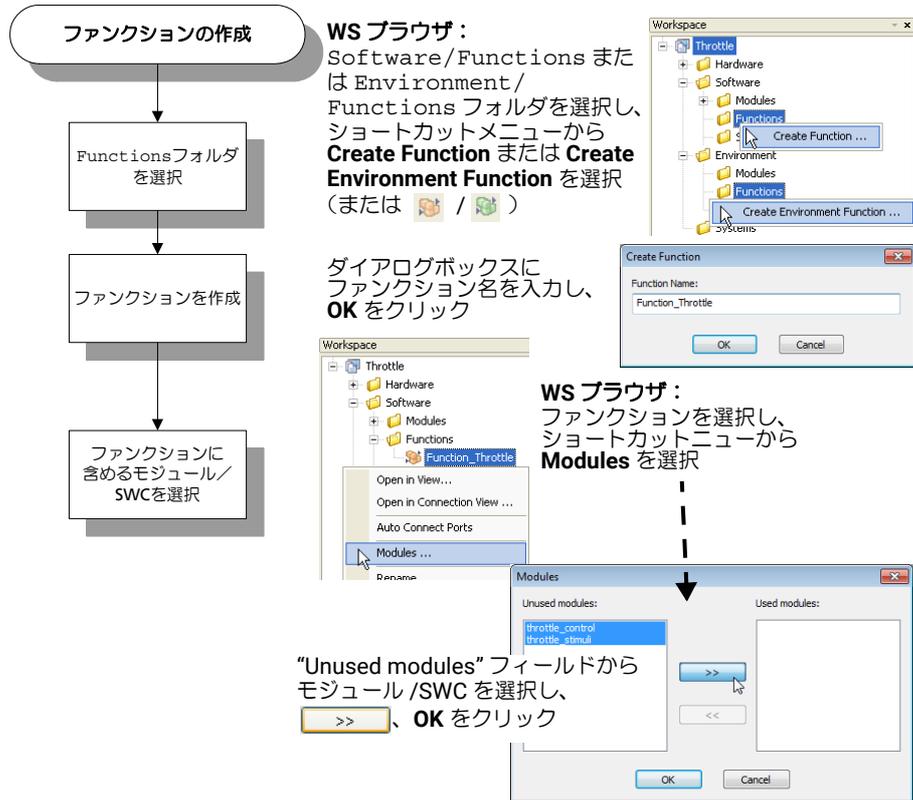


図 4-4 ファンクションを作成する

次に、ファンクション内のモジュールまたは SWC 間の接続や、ファンクション外部との接続のためのポートを、手動操作で定義します。

SWCを含まないファンクションの設定

PC上にファンクションを表示

ポートを挿入

ソースとシンクを接続

接続ウィザードを使用

WS ブラウザ：ファンクションをダブルクリック

PC (プロセスコンフィギュレータ)：ショートカットメニュー Create Port > Scalar In Port => in
Create Port > Scalar Out Port => out

PC：ソース (シンク) をクリックし、シンク (ソース) までラインを引く

WS ブラウザ：
 ファンクションを選択し、ショートカットメニューから **Open in Connection View** を選択して接続ウィザードを開く

接続ウィザード：
 ポートタイプを選択 (1)¹、出力を持つコンポーネント (2)、入力を持つコンポーネント (3)、入力 / 出力 (4) を選択し、ボタンをクリック

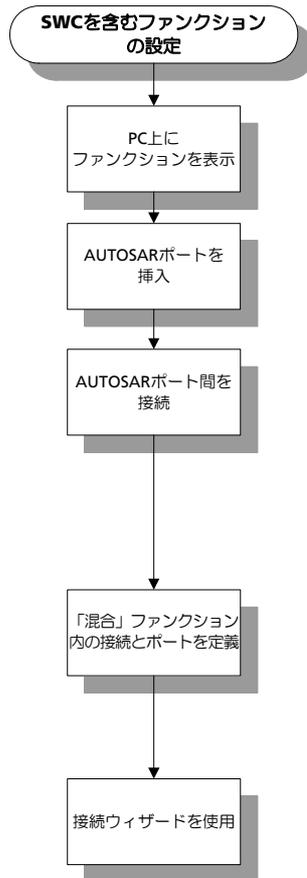
(同名のポートを自動接続するにはをクリック)

接続モードを“Dynamic”に設定

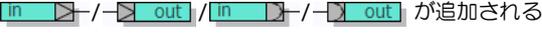
Dynamic	Out	In
<input checked="" type="checkbox"/>	Perc_Desired_Position	Perc_Desired_Position

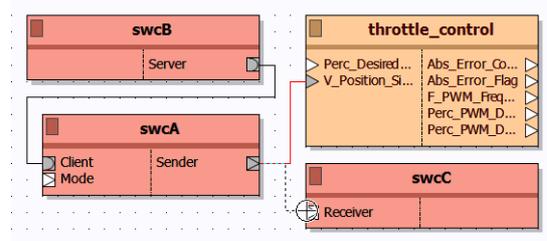
図 4-5 SWC を含まないファンクションを定義する

1. **Scalar in/out**：モジュール
Sender/Receiver / **Server/Client**：AUTOSAR SWC



WS ブラウザ：ファンクションをダブルクリック

PC (プロセスコンフィギュレータ)：
 センダ (レシーバ) またはクライアント (サーバー) をクリックして空白部分までラインを引き、**Create Port** を選択
 ⇒ AUTOSAR ポート

 センダ (レシーバ) をクリックしてレシーバ (センダ) までラインを引く
 クライアント (サーバー) をクリックしてサーバー (クライアント) までラインを引く



「混合」ファンクション：SWC のセンダ / レシーバをモジュールのシンク / ソースに接続、モジュールのシンク / ソースをクリックして空白領域にラインを引き、**Create Port** を選択
 ⇒ 非 AUTOSAR ポートが追加される



WS ブラウザ：
 ファンクションを選択し、ショートカットメニューから **Open in Connection View** を選択

接続ウィザード：
 ポートタイプを選択 (1)¹、出力を持つコンポーネント (2)、入力を持つコンポーネント (3)、入力 / 出力 (4) を選択し、ボタンをクリック

(同名のポートを自動接続するには  をクリック)

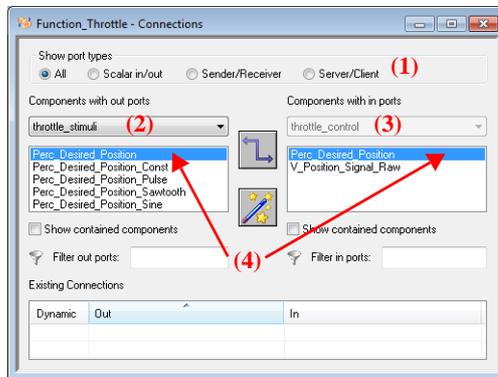


図 4-6 SWC を含むファンクションを定義する

1. **Scalar in/out**：モジュール
Sender/Receiver / **Server/Client**：AUTOSAR SWC

4.5 ソフトウェアシステム/環境システムの作成

モジュール、レガシー SWC、ファンクションは、参照形式で「ソフトウェアシステム」(SWS) または「環境システム」(ES) に組み込まれます。ただし現行バージョンでは、SWC は環境システムには組み込みません。1つのモジュールまたは SWC は、直接またはファンクションの一部として、1つのソフトウェアシステムに1回だけ組み込むことができます。ファンクションやモジュール、SWC のシグナルソースとシンクを互いに接続し、ハードウェア接続のためのインターフェースを作成します。

ワークスペース内においては、ソフトウェアシステムは Software/Software Systems フォルダに格納され、ETAS Experiment Environment は Environment/Software Systems フォルダに格納されます。

ソフトウェアシステム / 環境システムの作成

↓

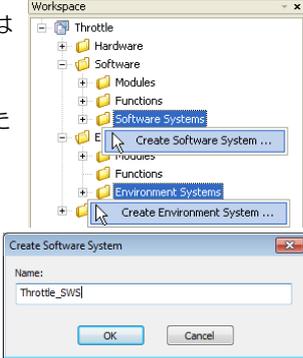
SWS/ESを作成

↓

SWS/ESに含めるモジュール/SWC/ファンクションを選択

WS ブラウザ :
Software Systems または Environment Systems フォルダを選択し、ショートカットメニューから **Create Software System** または **Create Environment System** を選択 (または / ボタン)

Dialogボックスに SWS / ES 名を入力し、**OK** をクリック



WS ブラウザ :
SWS または ES を選択し、ショートカットメニューから **Modules** または **Functions** を選択 (または /)

“Unused modules” または “Unused functions” フィールドからモジュール / SWC / ファンクションを選択し、、**OK**

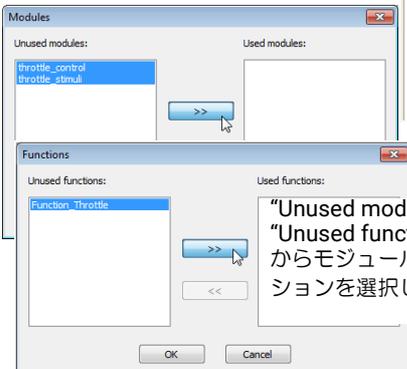


図 4-7 ソフトウェアシステム/環境システムを作成する

SWS 内のモジュール / SWC / ファンクション間の接続、およびハードウェア接続のためのポートは、プロジェクトコンフィギュレータ (PC) において、手動操作で定義します。

SWCを含まないソフトウェアシステム / 環境システムの設定

PC上にSWS/ESを表示

ポートを挿入

ソースとシンクを接続

接続ウィザードを使用

WS ブラウザ :
SWS/ES をダブルクリック

PC で操作する場合 :
ショートカットメニューの **Create Port > Scalar In Port / Scalar Out Port** ⇒ /

PC : ソース (シンク) をクリックし、シンク (ソース) までラインを引く

WS ブラウザ : SWS を選択し、ショートカットメニューから **Open in Connection View**

接続ウィザード :
ポートタイプを選択 (1)¹、
出力を持つコンポーネント (2)、
入力を持つコンポーネント (3)、
入力 / 出力 (4) を選択し、
 ボタンをクリック、
"Dynamic" 接続モード (ON/OFF) を設定 (5)
(同名ポートの自動接続には をクリック)

図 4-8 SWC を含まないソフトウェアシステム / 環境システムを定義する

1. Scalar in/out : モジュール
Sender/Receiver / Server/Client : AUTOSAR SWC

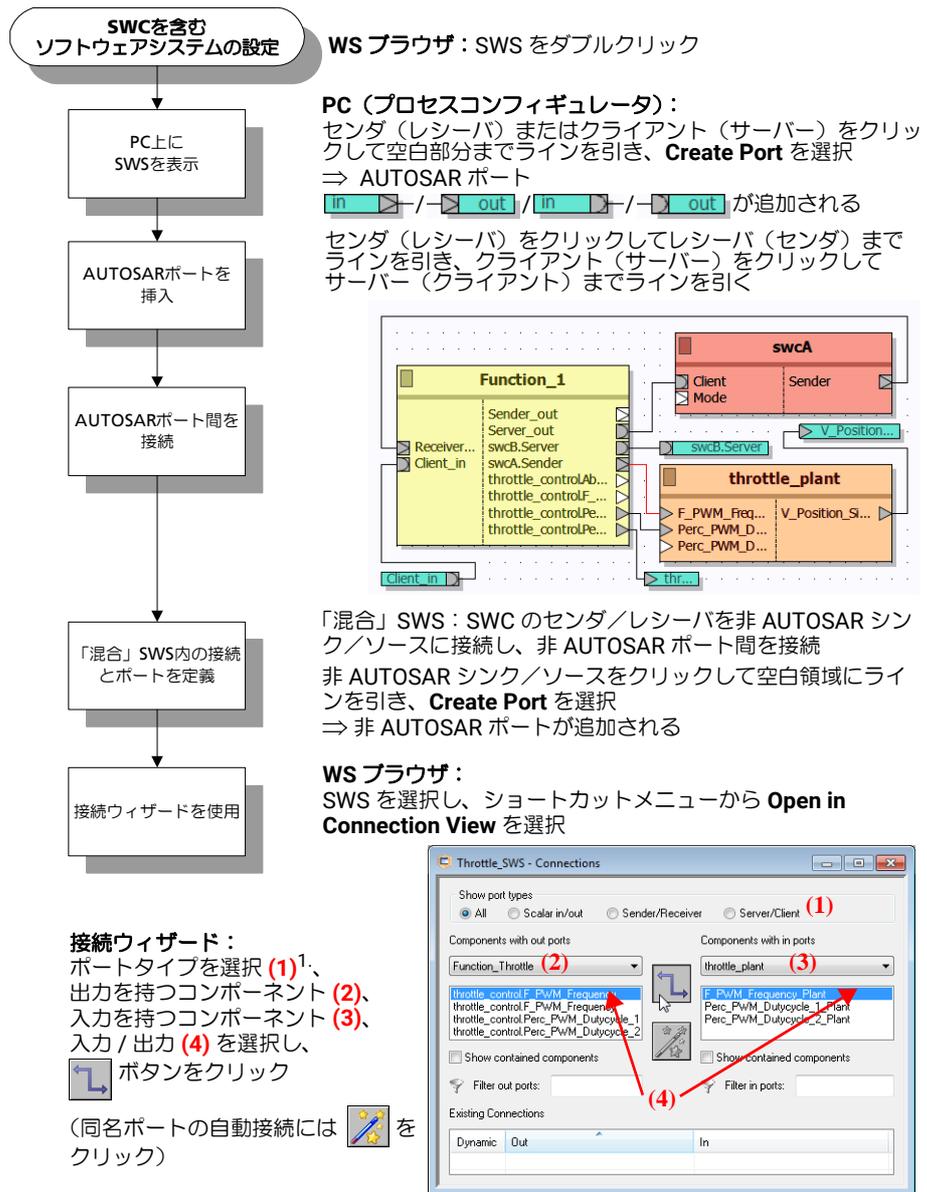


図 4-9 SWC を含むソフトウェアシステムを定義する

1. **Scalar in/out**：モジュール、
Sender/Receiver / **Server/Client**：AUTOSAR SWC

4.6 ハードウェアシステムの作成と設定

実験で使用する「ハードウェアシステム」を作成してコンフィギュレーションの設定を行うには、ハードウェアコンフィギュレータの機能を使用します。ハードウェアシステムにハードウェアを組み込む際は、“Insert Hardware System” または “Insert New Items” ダイアログボックスに各階層レベルで使用できるハードウェアアイテムが表示されるので、そこから必要なアイテムを選択します。

また、各ハードウェアコンポーネントの各種設定（モジュールパラメータ、測定ハードウェアのチャンネル設定など）も行えます。

ワークスペース内において、ハードウェアシステム（HWS）は Hardware フォルダに格納されます。

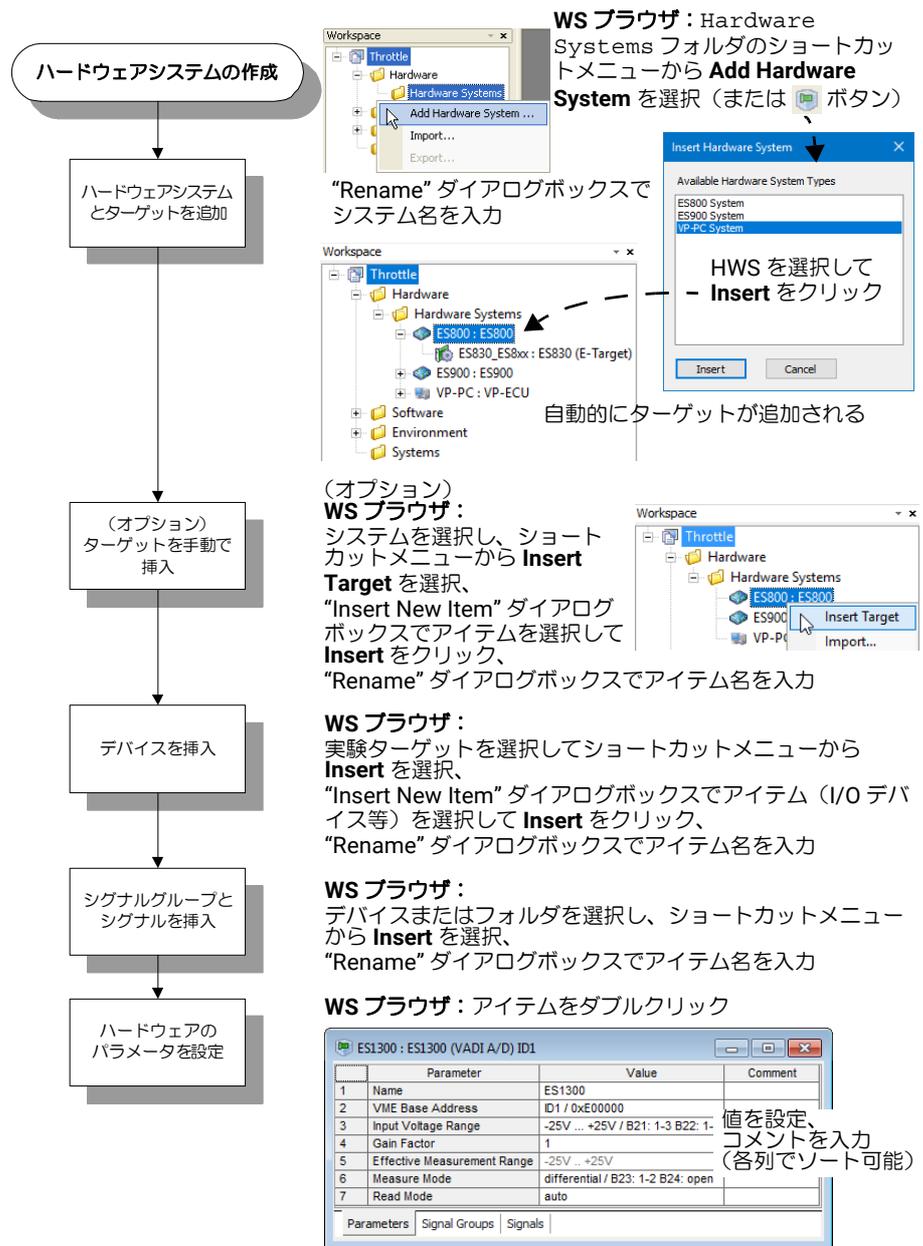


図 4-10 ハードウェアシステムを作成して設定する

4.6.1 ハードウェアシステムのインポート

HWS を手動操作で作成する代わりに、INTECRIO や INTECRIO-RLINK、ASCET-RP で作成されたハードウェアディスクリプション (*.hwx) を、新しい（または既存の）ハードウェアシステムにインポートすることができます。ディスクリプションには 2 種類のフォーマット（HWX1 と HWX2）があり、拡張子はどちらも *.hwx です。

i 注記

INTECRIO V4.7.3 以降、HWX1 フォーマットはインポートできなくなりました。

インポートするファイルを選択した後、ウィザードによってインポート処理が実行されます（図 4-11）。

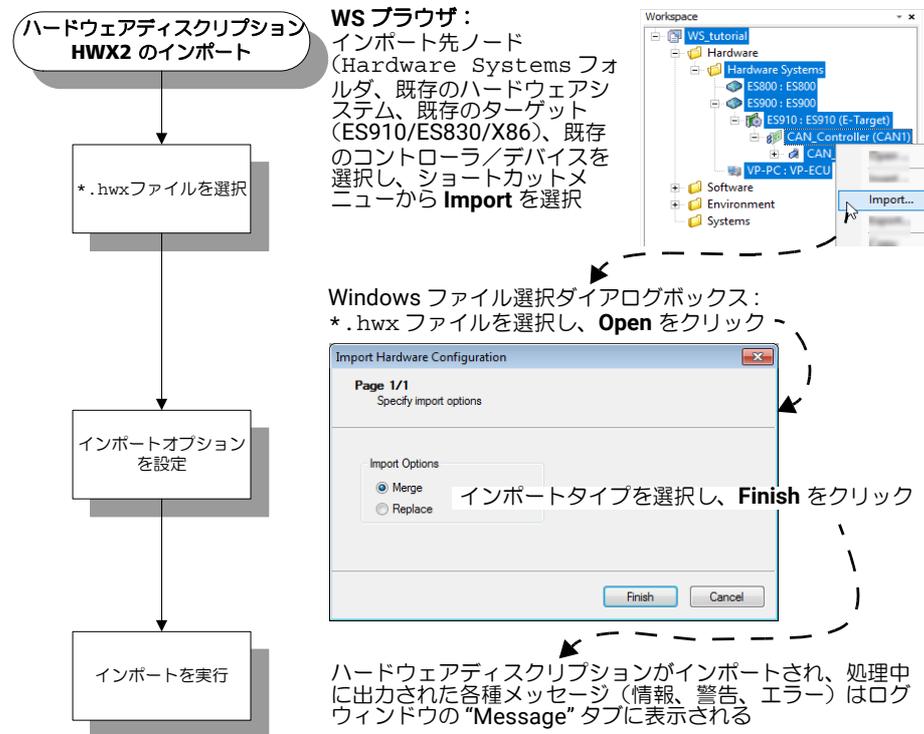


図 4-11 HWX2 フォーマットのハードウェアコンフィギュレーション (*.hwx) をインポートする

4.6.2 デイジーチェーンの設定

デイジーチェーンのコンフィギュレーション設定は、以下の手順で行います。

デイジーチェーンを挿入してコンフィギュレーションをインポート

→

デイジーチェーンを挿入

→

コンフィギュレーションファイルをインポート

→

デイジーチェーンを設定

WS ブラウザ：
ES910、または ES830 下の Ethernet コントローラを選択してショートカットメニューから **Insert** を選択、
“Insert New Item” ダイアログボックスで Daisychain を選択して **Insert** をクリック、
“Rename” ダイアログボックスでアイテム名を入力

WS ブラウザ： デイジーチェーンをダブルクリック

HC： ボタンをクリック

または WS ブラウザ：
デイジーチェーンを選択し、ショートカットメニューから **Import Daisychain Configuration** を選択

“Import file” ダイアログボックス
1 ページ目：
コンフィギュレーションタイプを選択して **Next** をクリック

2 ページ目：
コンフィギュレーションファイルを指定して **Next** または **Finish** をクリック

3 ページ目：
(既存のコンフィギュレーションをインポートする場合のみ)：
チェーンを選択して **Finish** をクリック

WS ブラウザ： インポート終了後、デイジーチェーンを選択してショートカットメニューから **Configure** を選択し、
デイジーチェーンコンフィギュレーションツールを開く
コンフィギュレーションツール： デイジーチェーンを設定
WS ブラウザ： デイジーチェーンを選択してショートカットメニューから **Update** を選択

図 4-12 デイジーチェーンの作成と設定

4.6.3 LIN コントローラの設定

LIN コントローラのコンフィギュレーションは以下のようにインポートします。

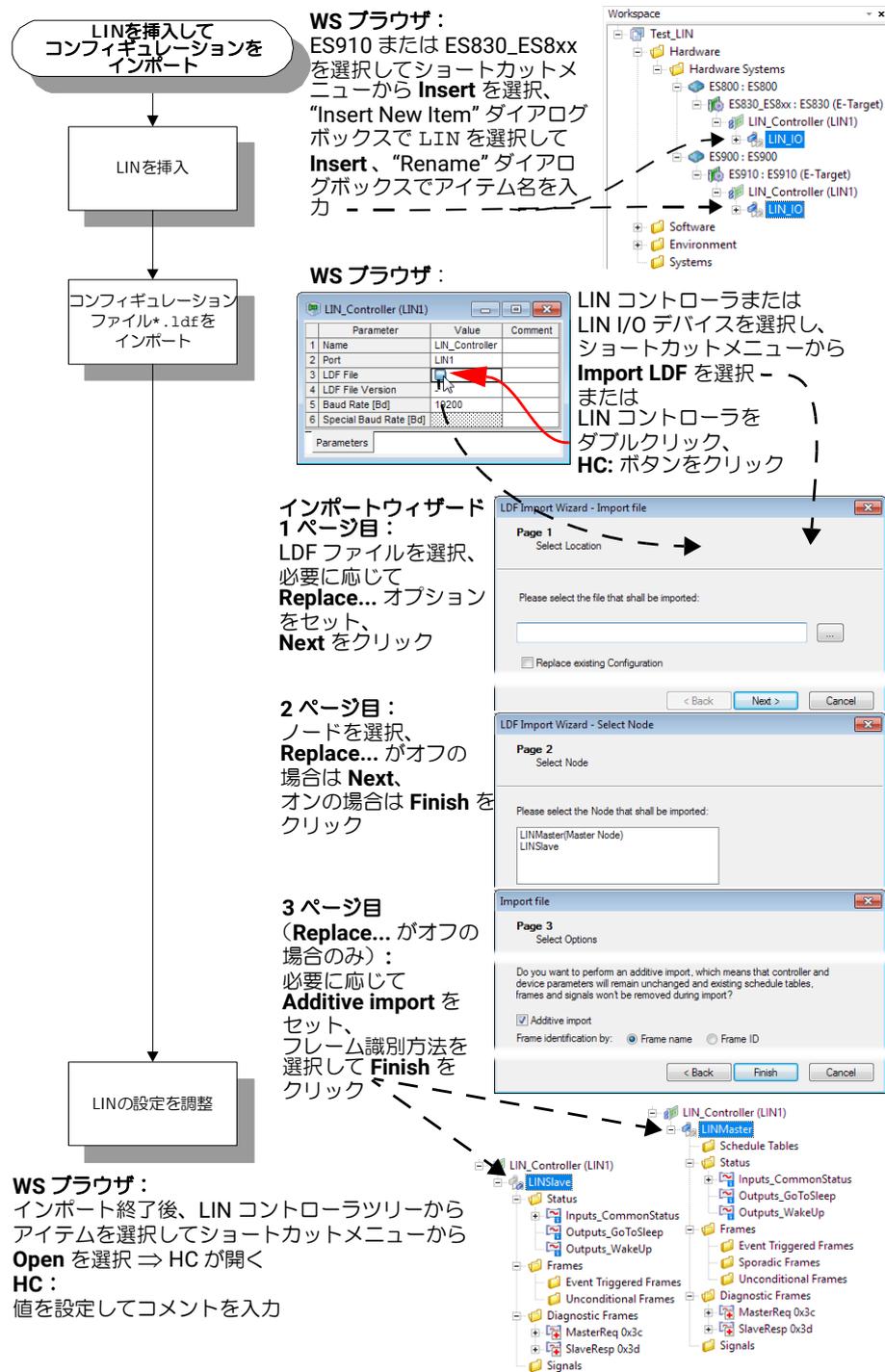


図 4-13 LIN コントローラの作成と設定

LIN ノード（デバイス）のタイプにはマスタとスレーブがあり、いずれかを設定できます。ノードタイプを変更すると、タイプに依存する設定内容は自動的に調整されます。

LINのデバイスタイプの変更

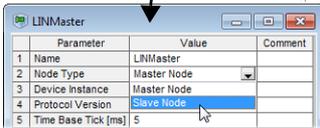
↓

マスタ → スレーブ

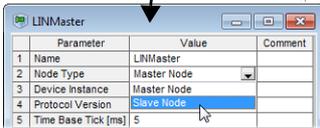
↓

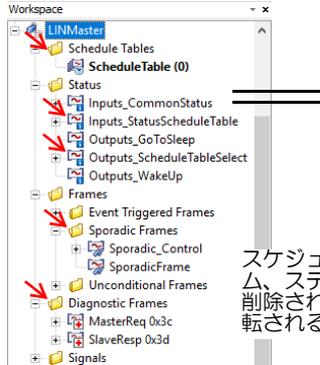
スレーブ → マスタ

WS ブラウザ：
マスタデバイスを選択してショートカットメニューから **Open** を選択、またはマスタデバイスをダブルクリック



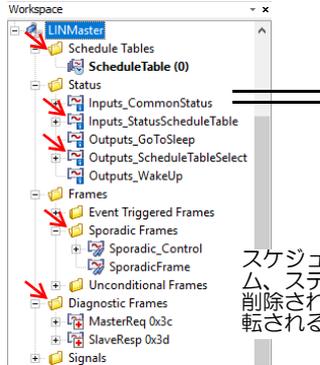
HC：
“Node Type” を Slave Node に変更、必要に応じてデバイス名を変更

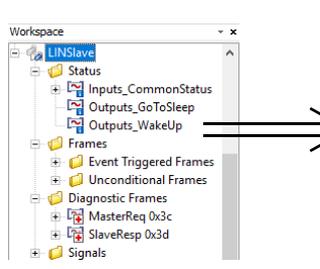




WS ブラウザ：
スレーブデバイスを選択してショートカットメニューから **Open** を選択、またはスレーブデバイスをダブルクリック

HC：
“Node Type” を Master Node に変更、必要に応じてデバイス名を変更





空のフォルダ Schedule Tables および Sporadic Frame、ステータスシグナルグループが作成され、診断フレームの方向が反転される

スケジュールテーブルと散発フレームを追加して設定する

図 4-14 LIN ノードタイプの変更

4.6.4 バイパスの設定

i 注記

ETK (8 Mbit/s) のサービスベースバイパスはサポートされていません。
INTECRIO V5.0.1 以降、ES910 での FETK バイパスは推奨されません。

バイパスデバイスを追加した後 (図 4-10)、以下の手順でバイパスを設定します。

バイパスのセットアップ

*.a21ファイルをインポート

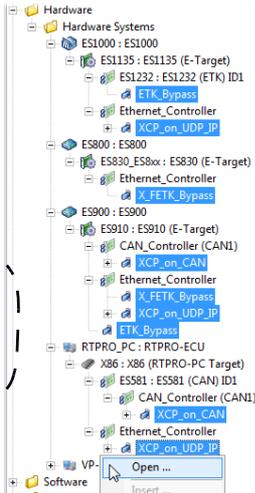
オプション:
デバイスを選択 (ETK/XETK/FETK バイパス)

オプション:
SBBバージョンを選択

バイパス信号を選択

信号グループと信号を設定

WS ブラウザ : ETK/XETK/FETK/XCP バイパスを選択してショートカットメニューから **Open** を選択、またはバイパスデバイスをダブルクリック

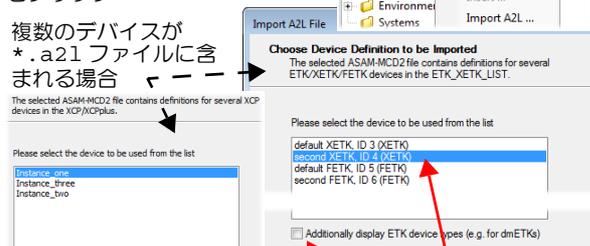


HC : 必要に応じてデクリプションキーを入力し、ボタンをクリック

または **WS ブラウザ** : ETK/XETK/FETK/XCP バイパスを選択し、ショートカットメニューから **Import A2L** を選択

ファイル選択ダイアログ : *.a21 ファイルを選択して **Open** をクリック

複数のデバイスが *.a21 ファイルに含まれる場合

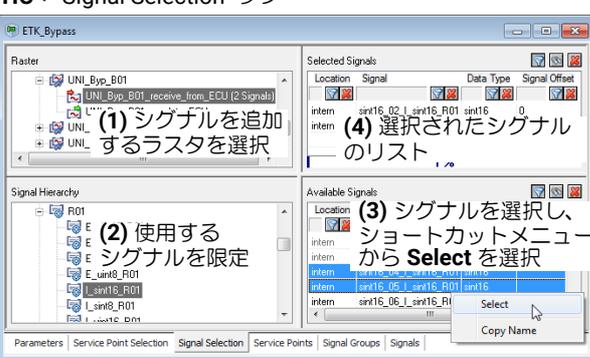


デバイスを選択し、必要に応じてオプションをオンにして、**Next** または **Finish**

複数の SBB バージョンが *.a21 ファイルに含まれる場合

SBB バージョン を選択し、**Finish**

HC : "Signal Selection" タブ



HC : 信号グループ/信号エディタを開き、オプションを設定

図 4-15 バイパスデバイスのセットアップ

「サービスベースバイパス」用のサービスポイントのディスクリプションは * .a21 ファイル内に含まれているので、これをインポートし、実際に使用するサービスポイントとシグナルを選択します。

サービスポイントの
セットアップ

サービスポイントの
機能を選択

シグナルの選択と調整

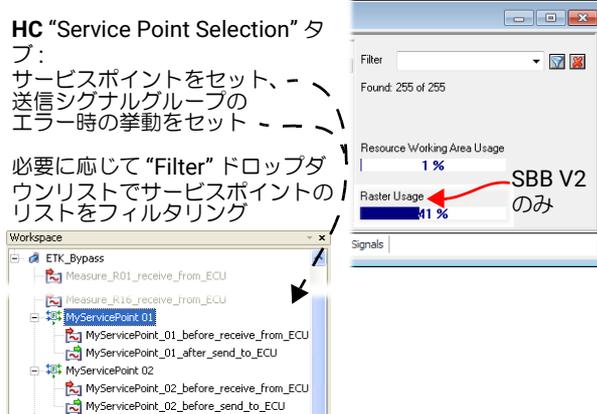
WS ブラウザ： サービスベースバイパス対応の * .a21 ファイルを使用する ETK/XETK/FETK バイパスデバイスを選択してショートカットメニューから **Open** を選択、またはバイパスデバイスをダブルクリック



↑ ↓ : ファンクション実行前に read/write (receive/send)
 📁 : オリジナルファンクションの実行 (追加ラスタ/メモリなし)
 ↑ ↓ : ファンクション実行後に read/write (receive/send)

HC “Service Point Selection” タブ：
 サービスポイントをセット、送信シグナルグループのエラー時の挙動をセット

必要に応じて “Filter” ドロップダウンリストでサービスポイントのリストをフィルタリング



HC “Signal Selection” タブ：
 サービスポイントの下のシグナルグループを選択してシグナルを追加 (図 4-15 参照)
 シグナルグループまたはシグナルをエディタで開き、オプションを設定

図 4-16 サービスポイントのセットアップ (サービスベースバイパス)

注記
 ETK (8 Mbit/s) のサービスベースバイパスはサポートされていません。

DISTAB17 を使用すると、サービスポイントのコンフィギュレーションをフックベースバイパス用に拡張することができます。フックベースバイパスのセットアップは、シグナルを選択する前に行う必要があります。

フックベースサービスポイントのセットアップ

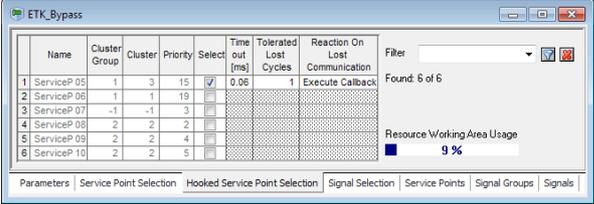
↓

フックベースサービスポイントの機能を選択

↓

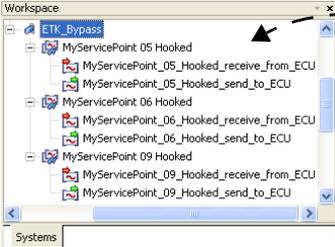
シグナルの選択と調整

WS ブラウザ : DISTAB17 対応の *.a21 ファイルを使用する ETK/XETK/FETK バイパスデバイスを選択してショートカットメニューから **Open** を選択、またはバイパスデバイスをダブルクリック



Name	Cluster Group	Cluster	Priority	Select	Time out [ms]	Tolerated Lost Cycles	Reaction On Lost Communication
1 ServiceP 05	1	3	15	<input checked="" type="checkbox"/>	0.06	1	Execute Callback
2 ServiceP 06	1	1	19	<input type="checkbox"/>			
3 ServiceP 07	-1	-1	3	<input type="checkbox"/>			
4 ServiceP 08	2	2	2	<input type="checkbox"/>			
5 ServiceP 09	2	2	4	<input type="checkbox"/>			
6 ServiceP 10	2	2	5	<input type="checkbox"/>			

HC "Hooked Service Point Selection" タブ :
使用するフックベースサービスポイントをアクティブ化、必要に応じて "Filter" ドロップダウンリストでフックベースサービスポイントのリストをフィルタリング



HC "Signal Selection" タブ :
フックベースサービスポイントの下のシグナルグループを選択してシグナルを追加 (図 4-15 参照)
シグナルグループまたはシグナルをエディタで開き、オプションを設定

図 4-17 フックベースサービスポイントのセットアップ (フックベースバイパス + DISTAB17)

4.6.5 CAN コンフィギュレーションファイルのインポート

CAN デバイスを追加した場合 (図 4-10)、シグナルとフレームを手動操作で作成するか、CAN コンフィギュレーションファイル (CAN データベースファイル (*.dbc) または AUTOSAR ファイル (*.arxml)) をインポートすることができます。CAN のほか、ES910+ES922 と ES830 では CAN FD がサポートされています。

CANコンフィギュレーションファイルのインポート

インポートウィザードを開く

CANコンフィギュレーションファイル、パスノード、インポートオプションを選択

CANフレームを選択

インポートの終了

WS ブラウザ : CAN_IO デバイスを選択して **Import CAN Configuration** を選択、または CAN_IO デバイスをダブルクリックして HC を開き、ボタンをクリック

“Import CAN Configuration File” ウィザード 1 ページ目 : ファイル名 (*.dbc か *.arxml) を入力または選択 (...) して

2 ページ目 : パスノードを選択、**Next** をクリック

3 ページ目 : フレーム識別モードを選択して、オプションを設定し、**Next** をクリック

4 ページ目 :

フィルタ使用

インポートするフレームの選択

コンフィギュレーション内に存在するフレームの表示

選択完了後、**Finish** をクリック

Frame Name	CAN Configuration Frame ID	Frame ID in Model
B19	0x2EDAF	0x2EDAF
B20	0x31524	
B23	0x38B83	0x38B83
B24	0x382F8	0x382F8
B27	0x42957	
B77	0x76ADEC	

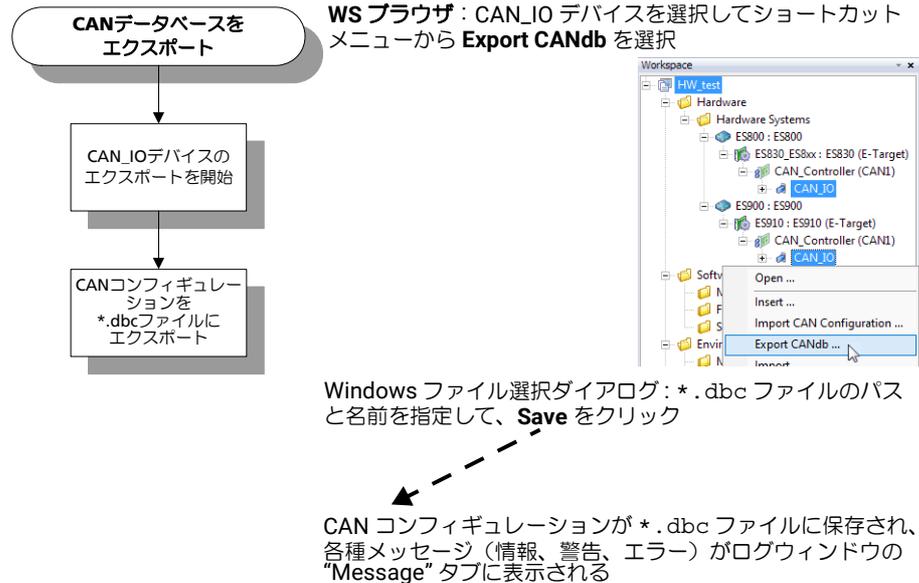
図 4-18 CAN コンフィギュレーションファイルのインポート

4.6.6 CAN コンフィギュレーションのエクスポート

注記

現時点では *arxml 形式での CAN コンフィギュレーションのエクスポートはサポートしていません。

設定の完了した CAN デバイスのコンフィギュレーションは、CAN データベースファイル (*.dbc) としてエクスポートすることができます。CAN のほか、ES910+ES922 と ES830 では CAN FD がサポートされています。



注記

CAN コンフィギュレーション内の以下のようなアイテムはエクスポートされません。

- 空の受信フレームと多重化グループ（multiplex group）
- CANdb フォーマットで使用できない HC パラメータ

図 4-19 CAN データベースのエクスポート

4.7 システムプロジェクトのセットアップ

1 つのハードウェアシステムと 1 つのソフトウェアシステム、さらにオプションで 1 つの環境システムを結合して、「システムプロジェクト」(SP) という 1 つの総合的なプロジェクトを定義します。

システムプロジェクトにおいて、ハードウェアの入出力信号をソフトウェアシグナルに割り当て、さらにオペレーティングシステムコンフィギュレーションで定義されているプロセスシーケンスにマッピングします。

4.7.1 システムプロジェクトの作成

ワークスペース内において、システムプロジェクトは Systems フォルダに格納されます。このフォルダには複数のシステムプロジェクトを格納でき、そのうち1つを「アクティブシステム」として実際に使用します。

本項では、ラピッドプロトタイピング用のシステムプロジェクトについて説明します。バーチャルプロトタイピングについては 4.11 項「仮想プロトタイピング (バーチャルプロトタイピング)」を参照してください。

システムプロジェクトをセットアップするには、まず Systems フォルダにシステムプロジェクト (SP) を作成し、そこにハードウェアシステム (HWS) とソフトウェアシステム (SWS)、および環境システム (ES) を割り当てます。

WS ブラウザ : Insert > Create System...
 ボタン、または Systems フォルダのショートカットメニューから **Create System** を選択

ダイアログボックスで SP の名前を入力

WS ブラウザ :
 SP を選択し、ショートカットメニューから **Add Hardware System** を選択

HW を選択して OK をクリック

WS ブラウザ :
 SP を選択し、ショートカットメニューから **Add Software System** を選択

SWS を選択して、OK をクリック

WS ブラウザ : SP を選択し、ショートカットメニューから **Add Environment System** を選択、ES を選択して OK

図 4-20 システムプロジェクトを作成する

4.7.2 ハードウェアとソフトウェアの接続

次に、ハードウェアシステム内のどのデバイスをシステムプロジェクト内で使用するかを選択します。ソフトウェアシステムからの接続は、ここで選択されたデバイスに対してのみ行えます。ハードウェアとソフトウェアのポートを接続し、実行可能なプロトタイプを生成します。

ハードウェアとソフトウェアの接続

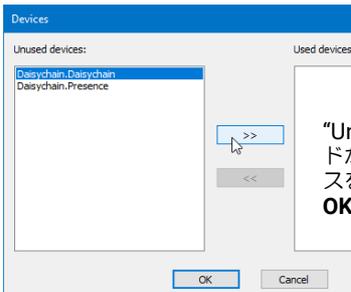
デバイス（ボード）を選択

PC内のシステムプロジェクトを表示

ソースとシンクを接続

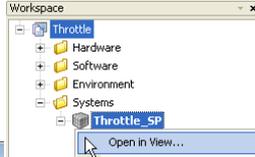
接続をインポート

WS ブラウザ：
SP を開き、 ボタンをクリック、または SP のショートカットメニューから **Devices** を選択



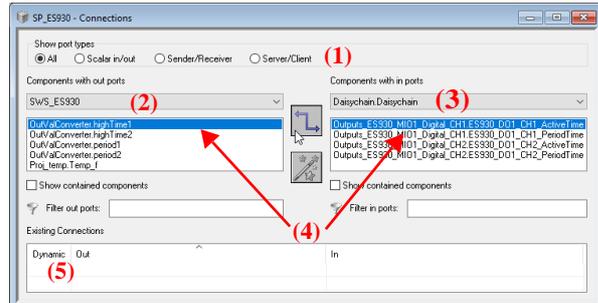
“Unused devices”フィールドからハードウェアデバイスを選択して、、**OK** をクリック

WS ブラウザ：
SP を選択してショートカットメニューから **Open in View** を選択（または SP をダブルクリック）



PC:
ソース（シンク）をクリックし、シンク（ソース）までラインを引く

または、**WS** を選択してショートカットメニューから **Open Connection Window** を選択



WS ブラウザ：
SP を選択して、ショートカットメニューから **Import HWC Connection** を選択、
“Import HWC Connection”ダイアログボックスで HWX ファイルを選択し、**Open** をクリック
HW 接続が名前ベースでインポートされます。同名の接続がすでに存在する場合、上書きは行われません。

接続ウィザード：
ポートタイプを選択 (1)¹、
出力を持つコンポーネント (2)、
入力を持つコンポーネント (3)、
入力 / 出力 (4) を選択し、 ボタンをクリック、
“Dynamic” 接続モード (ON/OFF) を設定 (5)²
(同名ポートの自動接続には  をクリック)

図 4-21 ハードウェアとソフトウェアを接続する

1. **Scalar in/out**：モジュール、**Sender/Receiver**：AUTOSAR SWC
Server/Client：ハードウェア接続では使用不可
2. SWC 接続では “Dynamic” 接続モードは使用不可

4.8 オペレーティングシステムの設定

オペレーティングシステムのコンフィギュレーション設定は、リアルタイムプロトタイプシステムを作成するうえで非常に重要な作業のひとつです。

オペレーティングシステム (OS) は、プロセッサの占有を競合するタスクやプロセス、およびランナブルエンティティ (RE: **Runnable Entity**、「ランナブル」とも呼ばれます) の処理シーケンスを決定し、必要に応じてタスクの実行を切り替えます。これらを行うために必要な条件設定を、OS コンフィギュレーションウィンドウで行います。

4.8.1 オペレーティングシステムの自動設定

INTECRIO には OS の自動設定機能があります。

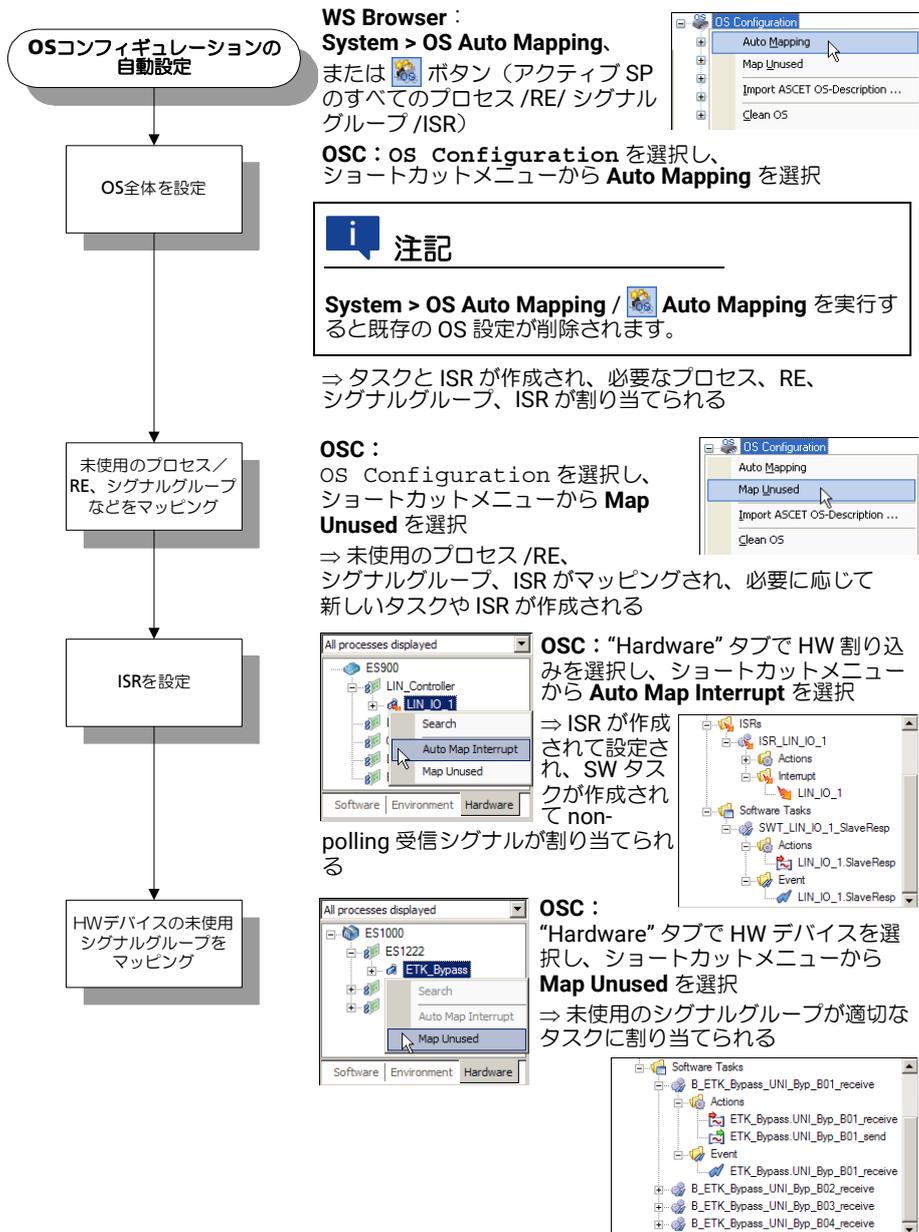


図 4-22 オペレーティングシステムを自動設定する

4.8.2 オペレーティングシステムの手動設定

INTECRIO では、OS のコンフィギュレーション設定を手動操作で行うこともできます。すべてを手動操作で設定したり、自動設定の結果を調整したりすることもできます (図 4-23)。

最初に必要なタイマタスクとソフトウェアタスク、および ISR を作成します。ただし Init タスクと Exit タスクは自動的に作成されます。

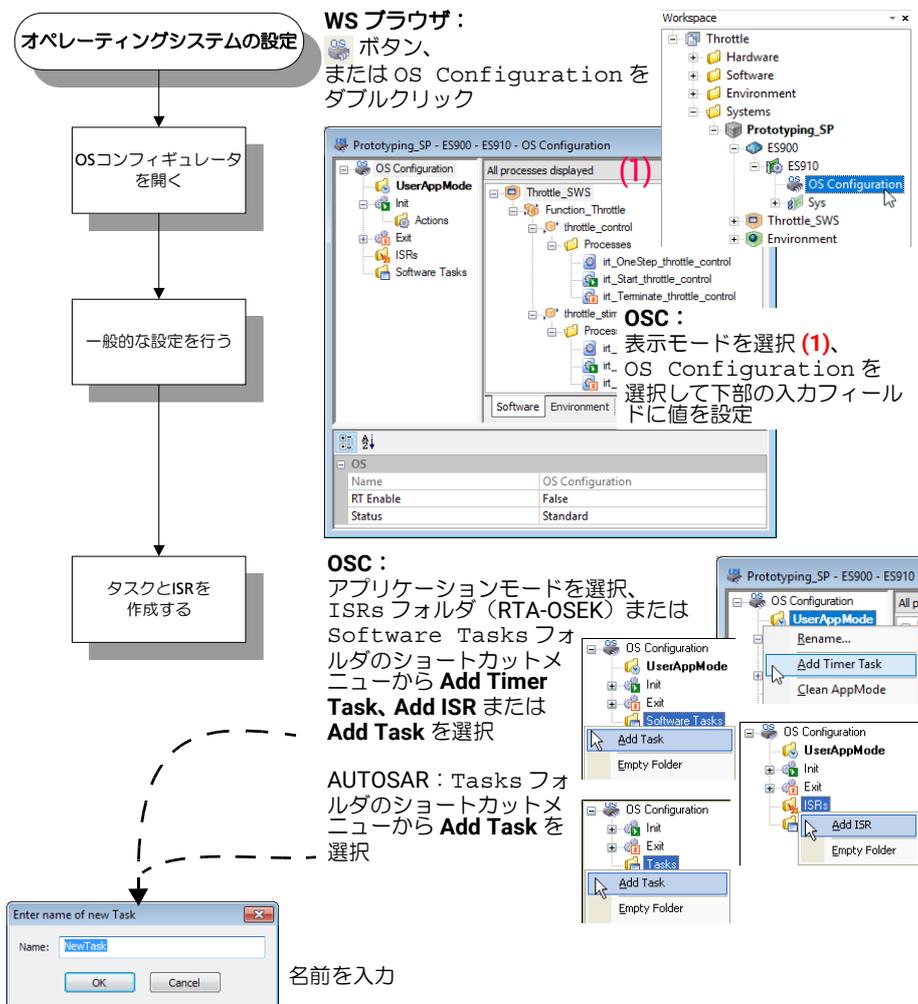


図 4-23 オペレーティングシステムを手動設定する

続いて、作成したタスクのプロパティを設定します。Init タスクと Exit タスクのプロパティは INTECRIO により自動的に定義されます。その後、プロセス（または RE）とシグナルグループをタスクに割り当てます。

タスクの設定

タスクを選択

タスク属性を調整

プロセス/REを
マニュアル操作で
割り当てる

シグナルグループと
イベントを割り当てる

空のタスクを削除

OSC : タスクを選択



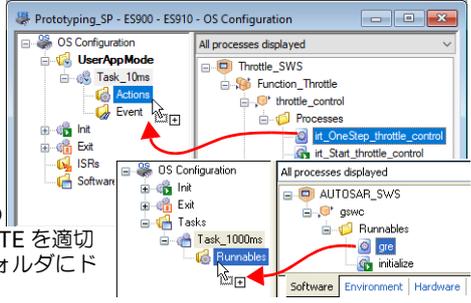
OSC 入力フィールド :
各プロパティの値を
選択、または直接入力

Timer Task	
Name	Task_10ms
Task ID	not yet assigned
Priority	0
Period	0.01
Delay	0
Execution Budget	0
Max. Number of Activation	1
Monitoring	False
Exclude from Tracing	True

Monitoring: True
Exclude from Tracing: True

Name	Task ID
Task_10ms	0

OSC :
"Software"/
"Environment" タブ上の
プロセス/RTE を適切な
タスクフォルダにドラッグ



OSC :
"Hardware"
タブ上のアイテムを、
タスクまたは
Actions /
Runnable
s フォルダ (  ), または Event フォルダ () に
ドラッグ

OSC :
OS Configuration (空の
タスクと ISR を削除)、アプリ
ケーションモード (空のタイマ
タスクを削除)、Software
Tasks フォルダ (空の SW タスクを削除) を
選択し、ショートカットメニューから **Delete
Empty Tasks** を選択



図 4-24 タスクを手動設定する

割り込みサービスルーチン (ISR) は、タスクの場合と同じような方法で設定します。各 ISR には HW 割り込みと、プロセスおよびシグナルグループが割り当てられます。HW 割り込みによっては、トリガイベントの下にサブイベントを定義し、そこにプロセスやシグナルグループを割り当てることもできます。

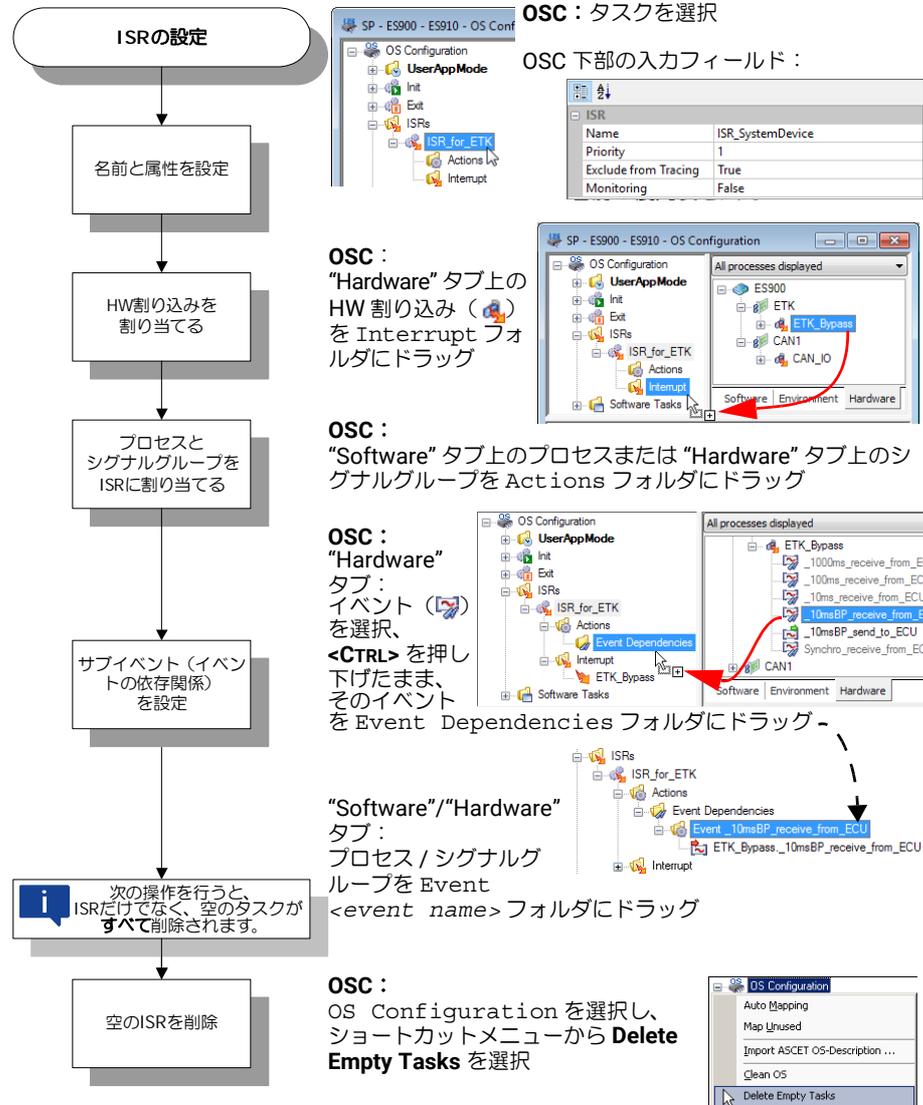


図 4-25 ISR を手動設定する

4.9 実行ファイルの生成

ビルド処理は、いくつかのオプションを設定するだけでほぼ自動的に実行されます。

WS ブラウザ :
Tools > Options、
 WS ショートカットメニュー
 の **Properties**、
 SP ショートカットメニュー
 の **Properties**

左側フィールド：
 Project
 Integration
 とサブアイテムを
 選択

右側フィールド：
 ビルドオプションを設定、**OK**

- Integration > Build** / ボタン： 実行ファイルを生成
- Integration > Rebuild** / ボタン： 実行ファイルを再生成
- Integration > Generate** / ボタン： コードだけを生成
- Integration > Clean** / ボタン： 生成されたファイルを削除
- Integration > Abort** / ボタン： 生成をキャンセル

図 4-26 実行ファイル（プロトタイプ）を生成する

4.10 実験

測定／適合、およびそれらの準備作業を行うための機能のほとんどは、ETAS Experiment Environment に含まれています。ETAS Experiment Environment (ETAS 実験環境) では、測定変数や適合変数を選択してオシロスコープ、棒グラフディスプレイ、適合エディタなどのさまざまなウィンドウに配置します。

注記

ES800 ハードウェアシステムを用いた実験を行うには、V7.2.17 以降の INCA と INCA-EIP、または V3.7.7 以降の ETAS 実験環境が必要です。

4.10.1 実験の準備

1 つのシステムプロジェクトについて、空の「実験」が 1 つ作成されます。ETAS Experiment Environment のウィンドウを開いて (図 4-27)、PC に接続されているハードウェア (ES900 または ES800) を選択し、実験に必要な各サブウィンドウとツールバーが表示されることを確認してください (図 4-28)。

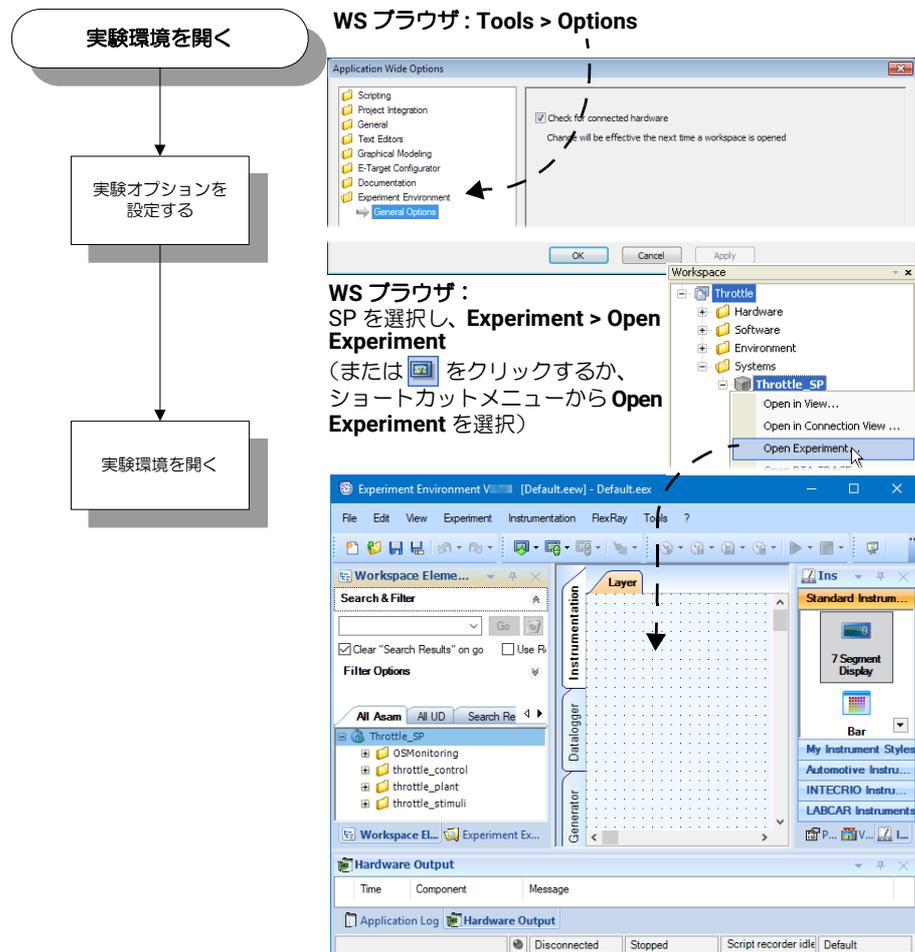


図 4-27 ETAS Experiment Environment (ETAS 実験環境) を開く

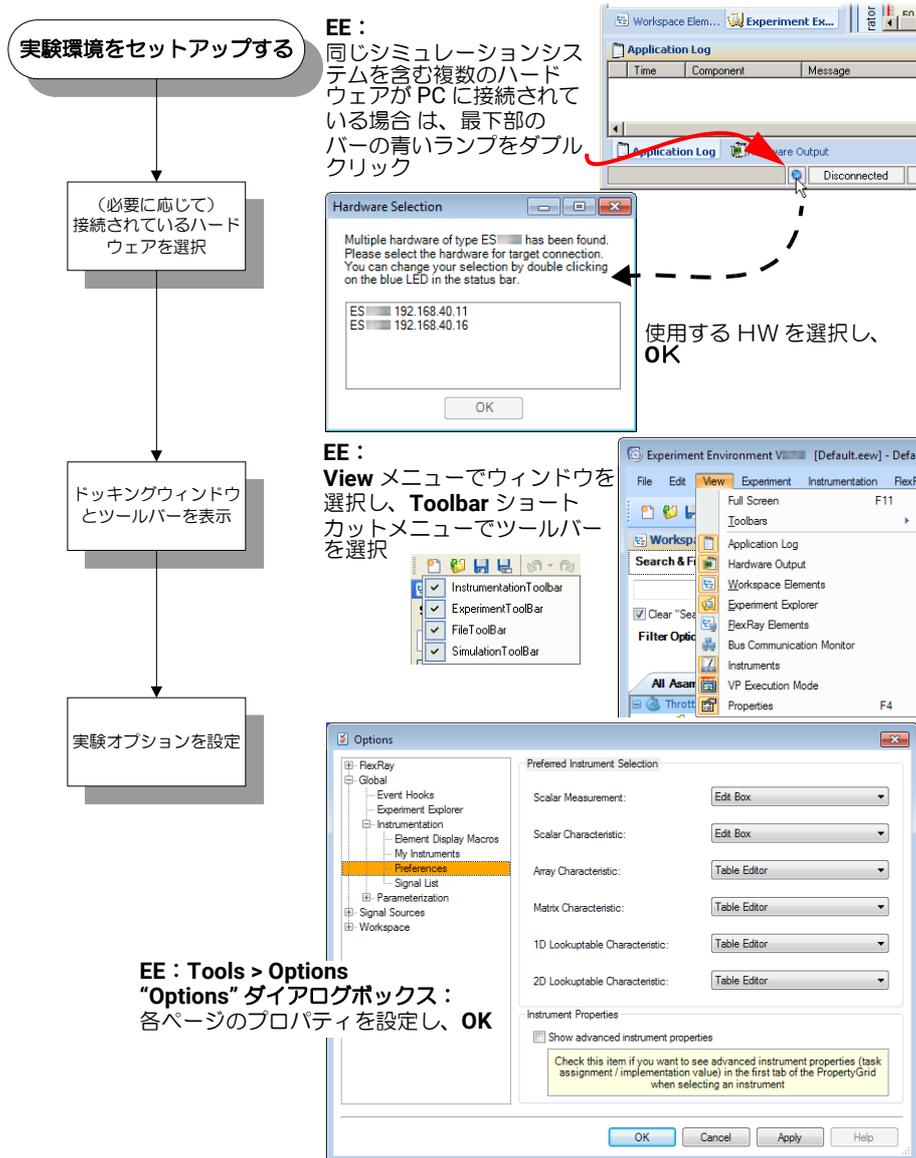


図 4-28 ETAS Experiment Environment をセットアップする

4.10.2 測定/適合ウィンドウの作成とセットアップ

ETAS Experiment Environment では、オシロスコープやその他のさまざまな測定/操作インストゥルメント（GUI エlement）を任意の数だけ作成し、そこに測定変数や適合変数を割り当てることができます。同じタイプのインストゥルメント（オシロスコープやテーブルエディタを除く）は1つの測定/適合ウィンドウにまとめて表示することもできます。

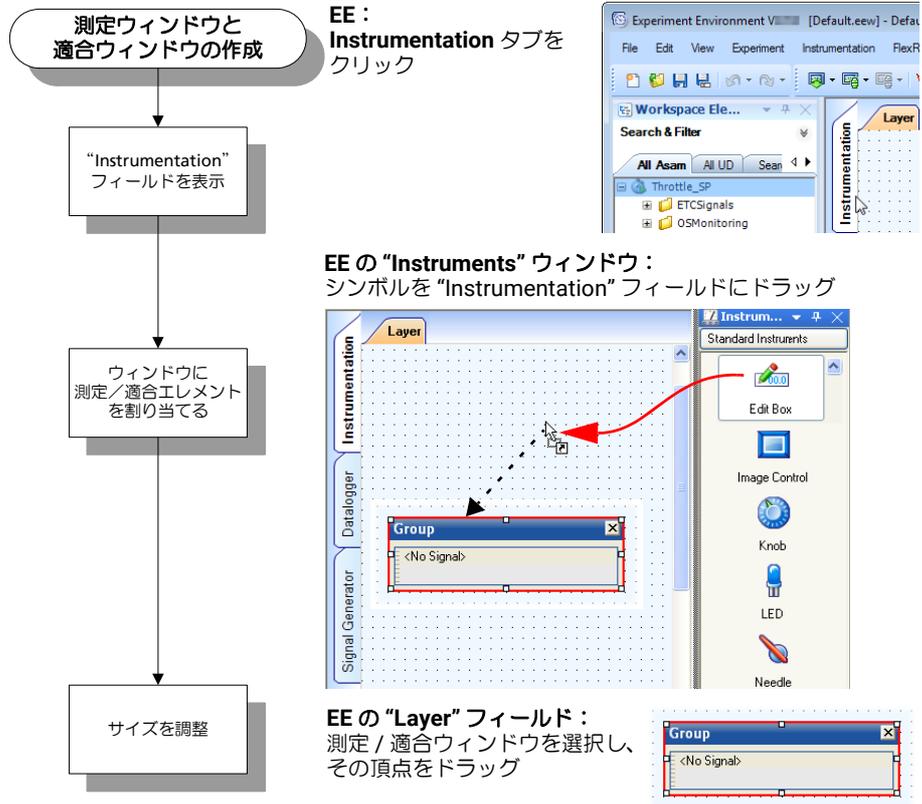


図 4-29 測定/適合ウィンドウを作成する

各インストゥルメントは、個別に表示設定を行えます。

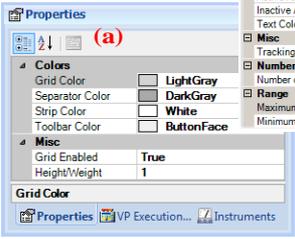
測定/適合
インストゥルメントの設定

オシロスコープを設定

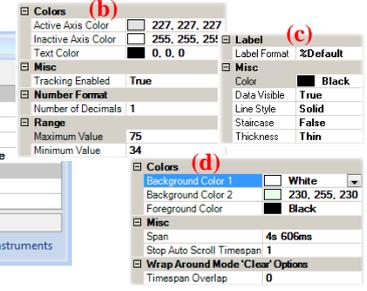
その他の
インストゥルメントを
設定

EE オシロスコープ：
 シグナル表示領域、座標軸、またはチャンネルリストを選択し、“Properties”ウィンドウにプロパティを表示する

“Properties”ウィンドウ：
 シグナル表示領域 (a)、軸 (b,d)、チャンネル (c) のプロパティを設定

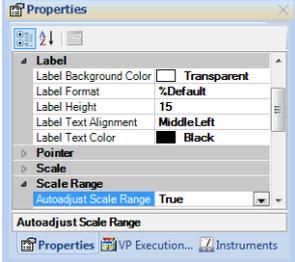


(a)



(b) (c) (d)

EE：
 インストゥルメントを選択して“Properties”ウィンドウにプロパティを表示する



“Properties”ウィンドウ：
 インストゥルメントのプロパティを設定

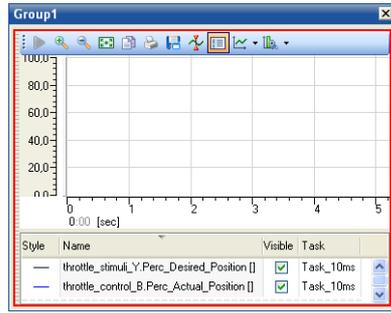
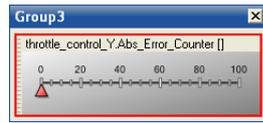



図 4-30 測定/適合インストゥルメントをセットアップする

4.10.3 測定変数と適合変数の割り当て

プロジェクトに定義されている測定変数と適合変数が“Workspace Elements”ウィンドウにリストアップされるので、ここから必要な変数を適切な測定/適合インストゥルメントに割り当てます。

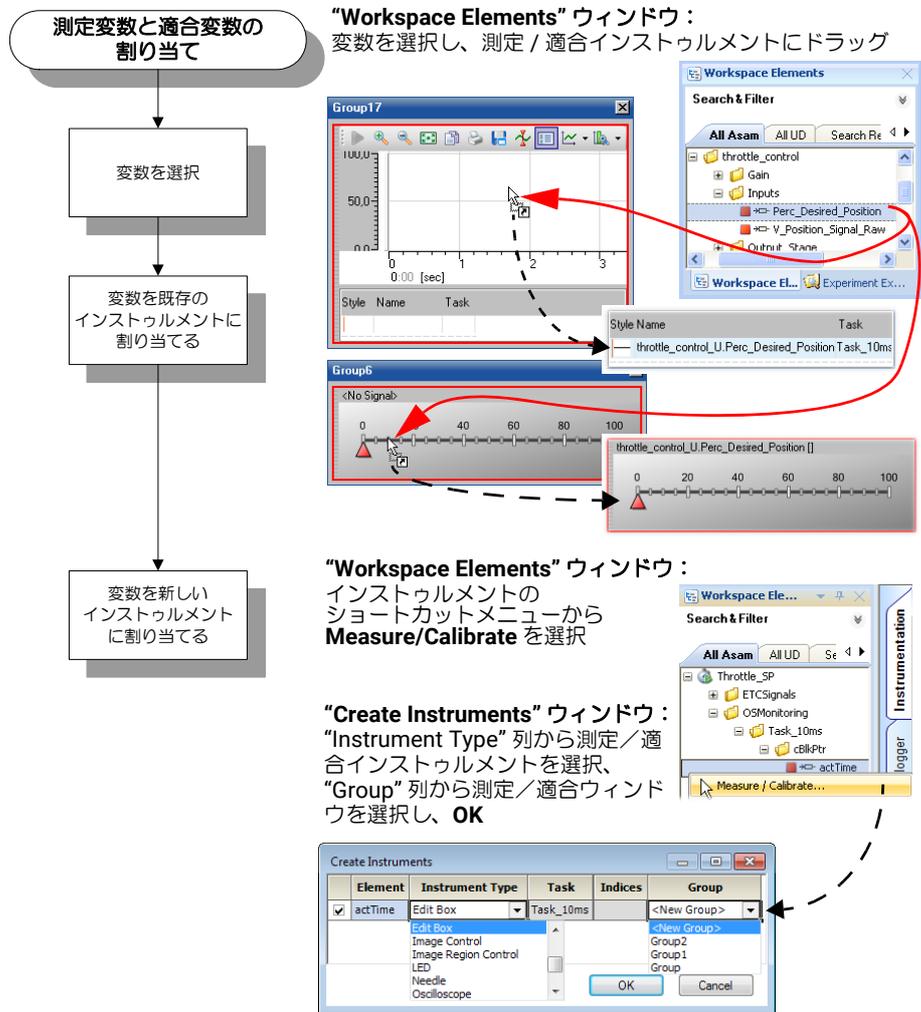


図 4-31 測定/適合変数を測定/適合インストゥルメントに割り当てる

4.10.4 レイヤの使用

デフォルト状態においては、“Instrumentation” フィールドには 1 つのレイヤ (“Layer” タブ) が含まれていて、そこに測定/適合ウィンドウを作成できるようになっています。レイヤの追加や削除、レイヤ名の変更、既存レイヤ間での測定/適合ウィンドウの交換が可能です。

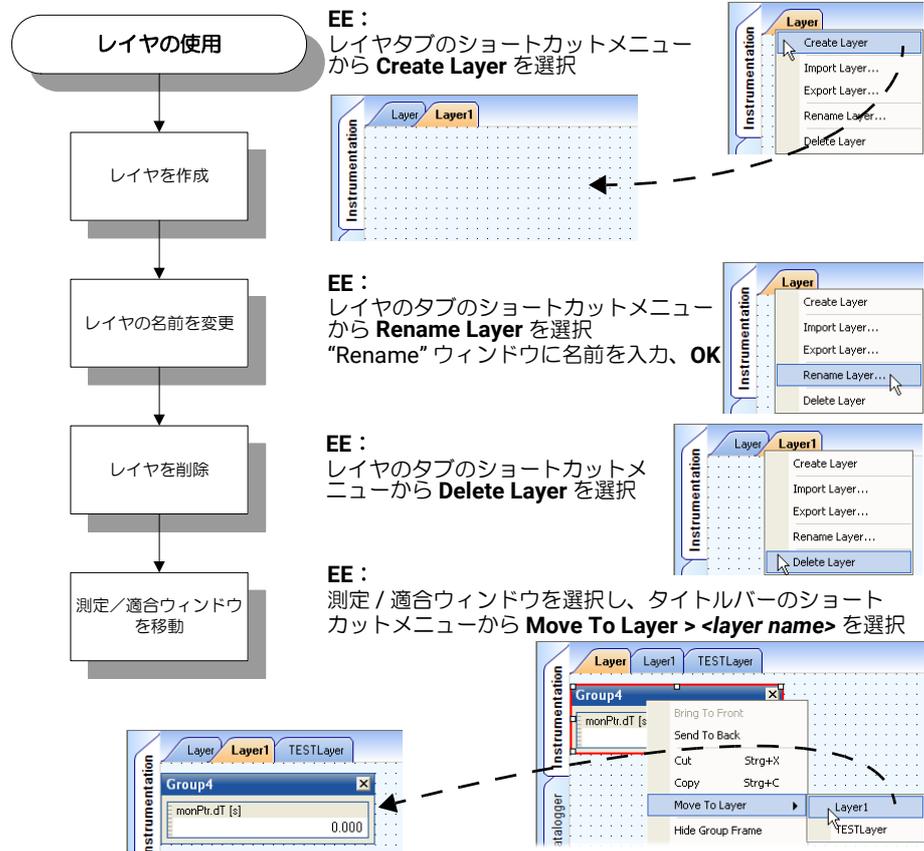


図 4-32 レイヤを使用する

4.10.5 実験の管理

実験のセットアップ（測定／適合ウィンドウとレイヤの作成、変数の割当て）が終わったら、その設定を保存します。その後は、別の実験を作成したり、既存の実験をロードしたりすることができます。

```

graph TD
    A[実験の管理] --> B[実験を作成]
    B --> C[実験をロード]
    C --> D[実験を保存]
    D --> E[実験を閉じる]
            
```

EE :
 現在開いている実験を閉じ、**File > New Experiment** ()

“New Experiment” ダイアログボックス :
 名前とパスを入力（または **Browse** ボタンで選択）、**OK**

現在開いている実験を閉じ、**File > Open Experiment** (または か、**<CTRL> + <O>**)
 実験を選択、**Open**

File > Save Experiment (または か、**<CTRL> + <S>**)、
 または **File > Save Experiment As**
 必要に応じて “Save Experiment As” ウィンドウにパスと名前を入力、**Save**

File > Close Experiment、
 現在の実験を保存 (**Yes**) または破棄 (**No**)

図 4-33 実験を管理する

4.10.6 データ収集の設定

INTECRIO では、データロガーで測定データを記録する際のさまざまな条件（保存場所や保存ファイル名、記録時間など）を設定できます。

データ収集の開始と終了を自動化するために、任意に定義されたイベントをトリガ条件として指定したり、プリトリガ時間やポストトリガ時間を指定したりすることもできます。

測定データの記録設定

“Datalogger” タブを開く

新しいデータロガーを追加

記録する測定変数を選択

記録用タスクを選択

記録ファイルを設定

トリガ条件を設定

記録時間

EE :
Data Logger タブをクリック

“Workspace Elements” 内の測定変数をデータロガーにドラッグ（または測定変数のショートカットメニューから Add to Data Logger を選択）

“Task” 列：タスクを選択

記録ファイル設定

Recording base filename: D:\ETASData\INTECRIO\Tutorial\MyLogFile.dat

Auto Increment: Auto Increment, 4 Digits

Final recording filename(s) preview: D:\ETASData\INTECRIO\Tutorial\MyLogFile001.dat

トリガエディタを開く

図 4-34 データ収集について設定する

4.10.7 実験の実行

実験を開始すると、生成されたプロトタイプが実験ターゲットに転送されます。このターゲット上で、まずリアルタイムオペレーティングシステムが稼働し、続いてシミュレーションが開始されます。

```

graph TD
    A[実験の開始] --> B[ターゲットへダウンロード  
(必要に応じてHWを選択)]
    B --> C[シミュレーション開始  
(OSを起動)]
    C --> D[測定開始]
    
```

EE : Experiment > Download > <SP>^{1/}

同じシミュレーションシステムを含む複数のハードウェアがPCに接続されている場合 ⇒ 最下部のバーの青いランプをダブルクリック

使用する HW を選択、OK

Experiment > Start Simulation > <SP>^{1/}

Experiment > Start Measurement > <SP>^{1/}

図 4-35 実験を開始する

1. <SP> はシステムプロジェクト名

測定データを記録する場合、測定を開始する前にデータロガーを起動する必要があります。データロガーが起動されると、通常は測定開始と同時に記録が開始されますが、トリガが設定されている場合はトリガ条件が満たされた時点で記録が開始されます。

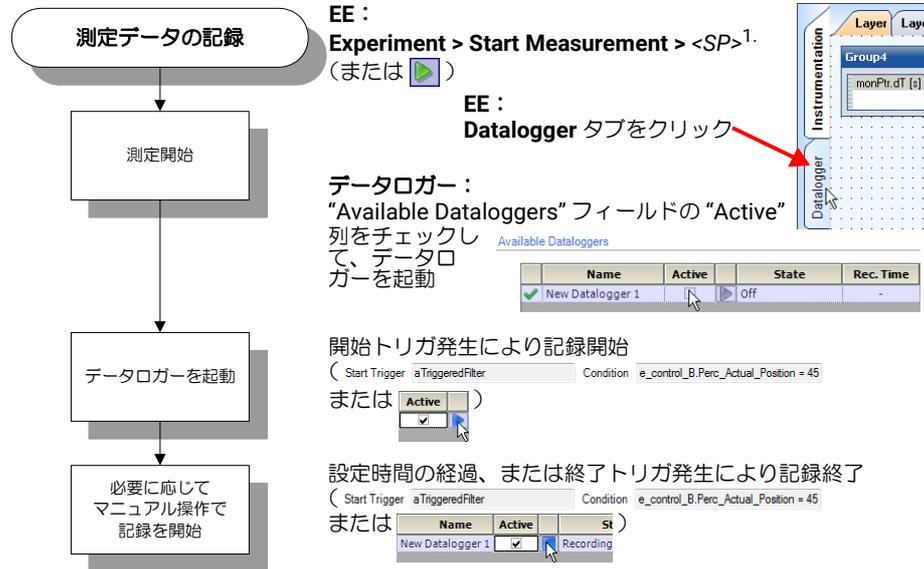


図 4-36 測定データを記録する

1. <SP> はシステムプロジェクト名

4.10.8 測定と適合

測定値は、オシロスコープやその他の各種測定インストゥルメントに表示します。オシロスコープには、測定値を分析するための機能が用意されています。

```

graph TD
    A[変数の測定] --> B[オシロスコープ上での測定データ表示の開始/停止]
    B --> C[測定カーソルをオンにして表示設定を行う]
    C --> D[測定データを分析(オシロスコープ)]
    D --> E[各種インストゥルメントを使用して測定を行う]
    
```

オシロスコープ:
表示の開始 / 再開: 、表示の休止:

カーソル ON/OFF:

カーソル上、およびチャンネルリスト内に値が表示される

マウスの左ボタンを使って任意の測定ポイントまでカーソルをドラッグ

カーソルを選択し、ショートカットメニューから **Show Cursor difference > <other Cursor>** を選択

⇒ カーソル上部にカーソル間の時間差が表示され、チャンネルリストには

時間差と値の差が表示される

測定開始 (EE の ボタン) により直ちに測定値が表示される

Style	Name	Visible	(A) 00:00:04.644	(B) 00:00:06.240
—	Perc_Actual_Position	<input checked="" type="checkbox"/>	69.8416969089319	41.0298167134427
—	Perc_Desired_Position	<input checked="" type="checkbox"/>	70	40

(A)	(B)	(A-B)
00:00:05.584	00:00:07.060	00:00:01.477
40	70	-30
49.1447513722242	67.8826806710549	-18.7379292988307

図 4-37 値を測定する

あるインストゥルメントに適合変数（特性値、特性カーブまたは特性マップ）を割り当てると、このインストゥルメントは「エディタ」として機能します。さまざまなタイプのエディタで、適合変数の値を直接変更できます。

パラメータの適合

数値エディタを使用

テーブルエディタを使用

スライダエディタを使用

適合したパラメータを保存

パラメータをロード

* 依存パラメータは適合不可

Group12

throttle_stimuli_P.StimulusSelect_Value []

3.000000

スクロールボタンを使用、またはテキストフィールドをクリックして値を入力

Group13

0	1	2	3	4
0	0.244918...	0.46211715	0.635148...	0.761594...

セルをダブルクリックして値を入力、**Apply**

Group8

throttle_control_PK_S_Value []

0 0,1 0,2 0,3 0,4 0,5

0,12

マウスを使用してポインタを移動

Save Parameters

- Gain
- Output_Stage
- Position_Filter
- Throttle_Controller
- Error_Check
- K_S
- Value
- Td_S
- PID_Control
- Td_S
- Value
- TLS
- Value
- Td_S
- Value
- Throttle_Position

パラメータの最終選択

出力ファイルを選択、**Add * オプションをオン、Save** をクリック

EE : "Workspace Elements" ウィンドウ: パラメータを選択し、ショートカットメニューから **Calibration/Parameters > Save** を選択

Experiment Explorer

- Datalogger Files
- Mapping Files
- Parameter Files
- Add File... Ins
- Set Display Order
- Download all active files
- Add New Parameter Variants... Shift+Ins

EE : "Experiment Explorer" タブ: Parameter Files フォルダを選択し、ショートカットメニューから **Add File** を選択

ファイル選択ダイアログボックス: ファイル名とフォーマットを選択して **Open** ⇒ ファイルが追加される

"Experiment Explorer" ウィンドウ: Parameter Files フォルダ内のファイルを選択してショートカットメニューから **Active** を選択 ⇒ ファイルから値が読み込まれる

Experiment Explorer

- Datalogger Files
- Mapping Files
- Parameter Files
- ParameterFile1.mpa
- Edit File...
- Add File... Ins
- Set Display Order
- Download all active files
- Active

図 4-38 エディタで適合を行う

4.10.9 ポストプロセッシング

測定結果をファイルに書き込んでおくと、そのファイルを測定データ分析ツールで読み込んでシグナル評価を行うことができます。

ETAS の MDA (Measure Data Analyzer: 測定データアナライザ) はこのような分析作業に最適なオフラインツールです。MDA には 2 種類の分析ウィンドウがあります。1 つはオシロスコープウィンドウで、これは通常オシロスコープまたは XY プロッタとして使用できます。もう 1 つは、正確な値を素早く読み取るのに適したテーブルウィンドウです。複数の測定ファイルに保存された測定シ

グナルを同じウィンドウに表示することも可能です。表示設定は任意に変更でき、設定を「分析コンフィギュレーション」として保存することもできます。分析を行う際には、測定ファイル内の任意のポイントにスムーズにアクセスするためのズーム機能や検索機能が用意されていて、値を読み取るための測定カーソル、シグナル間の差異の自動計算などの機能も利用できます。

MDA は INTECRIO、ASCET、INCA の各製品パッケージに含まれています。

4.11 仮想プロトタイピング (バーチャルプロトタイピング)

仮想プロトタイピング用のシステムプロジェクトの作成方法は、ラピッドプロトタイピング用のシステムプロジェクトを作成する場合とほとんど同じです。

VPプロジェクトの作成

↓

ソフトウェア/環境
/ハードウェア
システムを作成

↓

システムプロジェクト
を作成

↓

オペレーティング
システムを設定

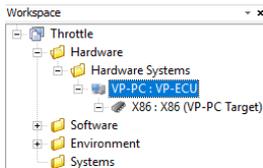
↓

コンパイラを設定
(必要に応じて)

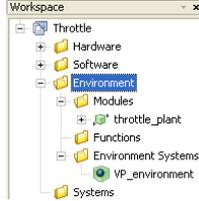
↓

デバッグの準備
(必要に応じて)

SWS :
モジュールを作成 (Simulink : **External Mode** オプションをオフ) してインポート (4.3 項参照)、ファンクションを作成して (4.4 項参照)、SWS を作成 (4.5 項参照)



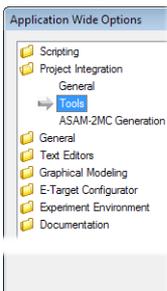
HWS :
HWS を作成してターゲットを作成 (4.6 項参照)



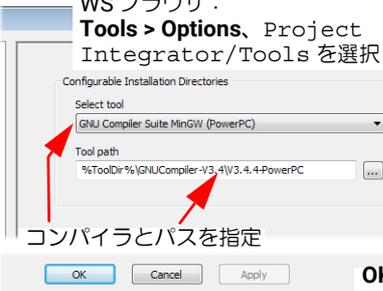
SP :
SP を作成 (4.7.1 項参照)、ES を割り当て、ES と SWS を接続 (VP の場合、HWS の設定や HW/SW の接続は不要)

OS :
OS を設定 (4.8 項参照)
(VP の場合、Event フォルダや ISR は使用しない)

コンパイラ :

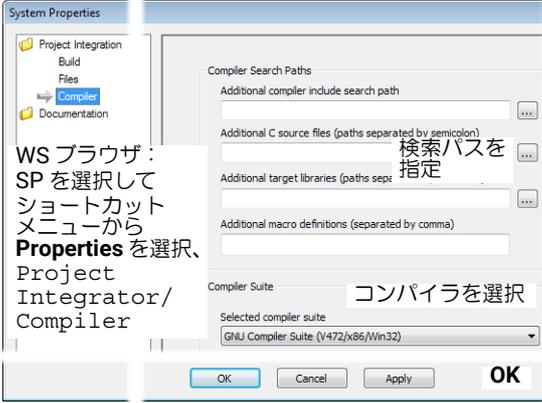


WS ブラウザ :
Tools > Options、**Project Integrator/Tools** を選択



コンパイラとパスを指定

WS ブラウザ :
SP を選択してショートカットメニューから **Properties** を選択、**Project Integrator/Compiler**



WS ブラウザ :
SP を選択して、ショートカットメニューから **Properties** を選択
Project Integration/Build: **Activate compiler debug code creation** をオン
Project Integration/Compiler: MS Visual Studio® compiler を選択

図 4-39 仮想プロトタイピング用プロジェクトを作成する

仮想プロトタイピング実験がラピッドプロトタイピング実験と大きく異なる点は、シミュレーション時間に関するモードを選択できるという点です。

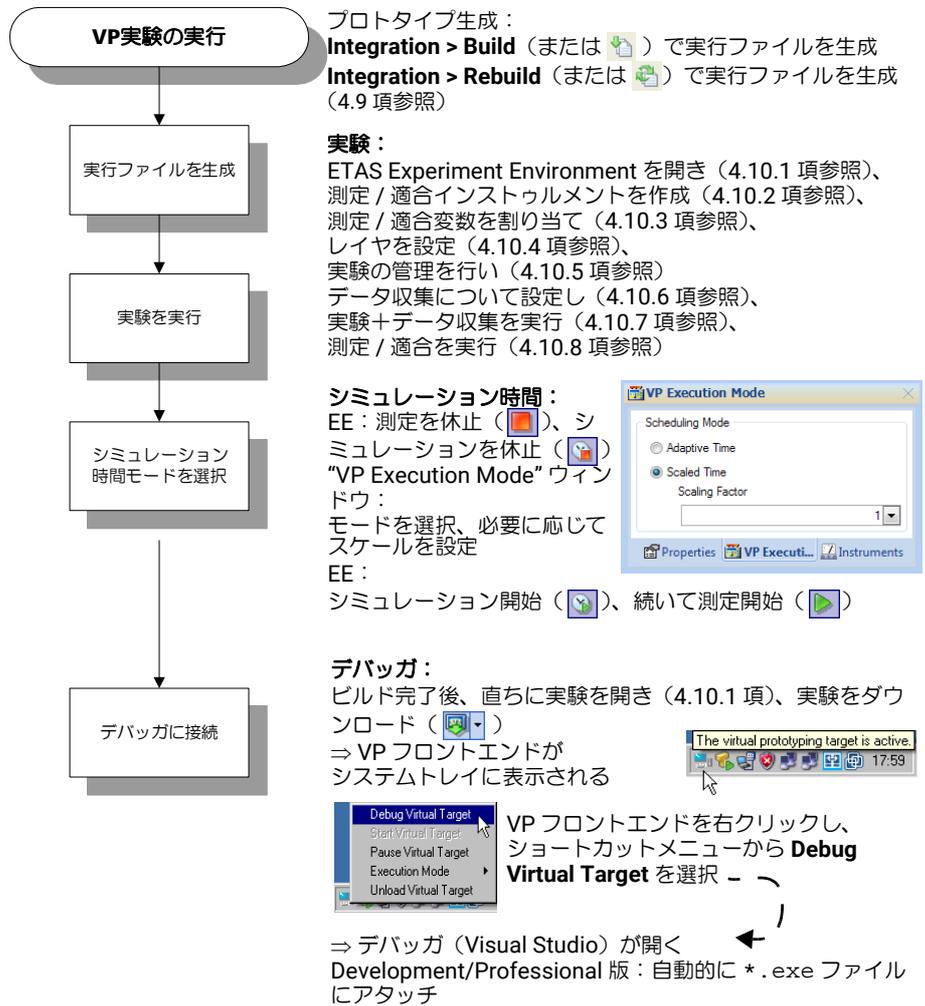


図 4-40 仮想プロトタイピング実験

4.12 ワークスペースの管理と交換

4.12.1 ワークスペースの管理

前述のように、INTECRIO では複数のワークスペースを扱うことができます。新しい空のワークスペースを作成する（43 ページの「新しいワークスペースの作成」を参照してください）だけでなく、たとえば他のユーザーが作成したワークスペースを開いて利用することもできます。保存されたワークスペースには、ワークスペース自体だけでなく INTECRIO ウィンドウのレイアウトも保存されます。

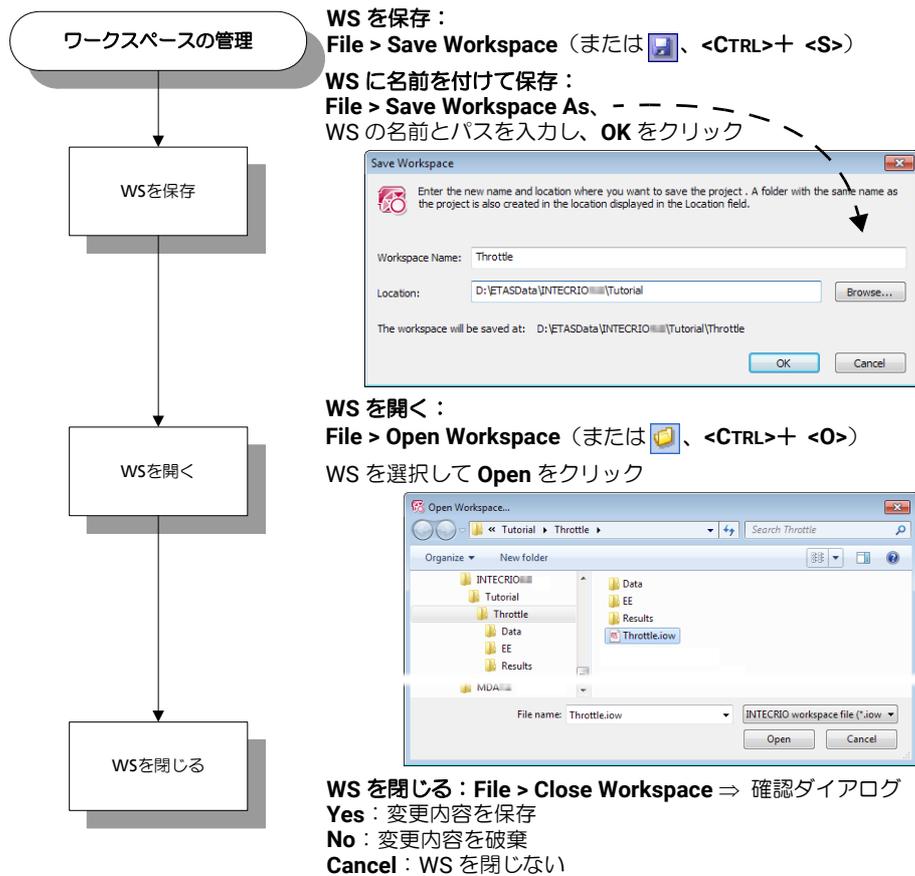


図 4-41 ワークスペースを管理する

4.12.2 インポート/エクスポートによるワークスペースの交換

WS ブラウザでは、ワークスペースのエクスポートやインポートを行えます。ワークスペースのエクスポート/インポート時には、任意のモジュールを含めることができます。この機能は、既存のワークスペースをユーザー間や PC 間で交換する際に非常に役立ちます。

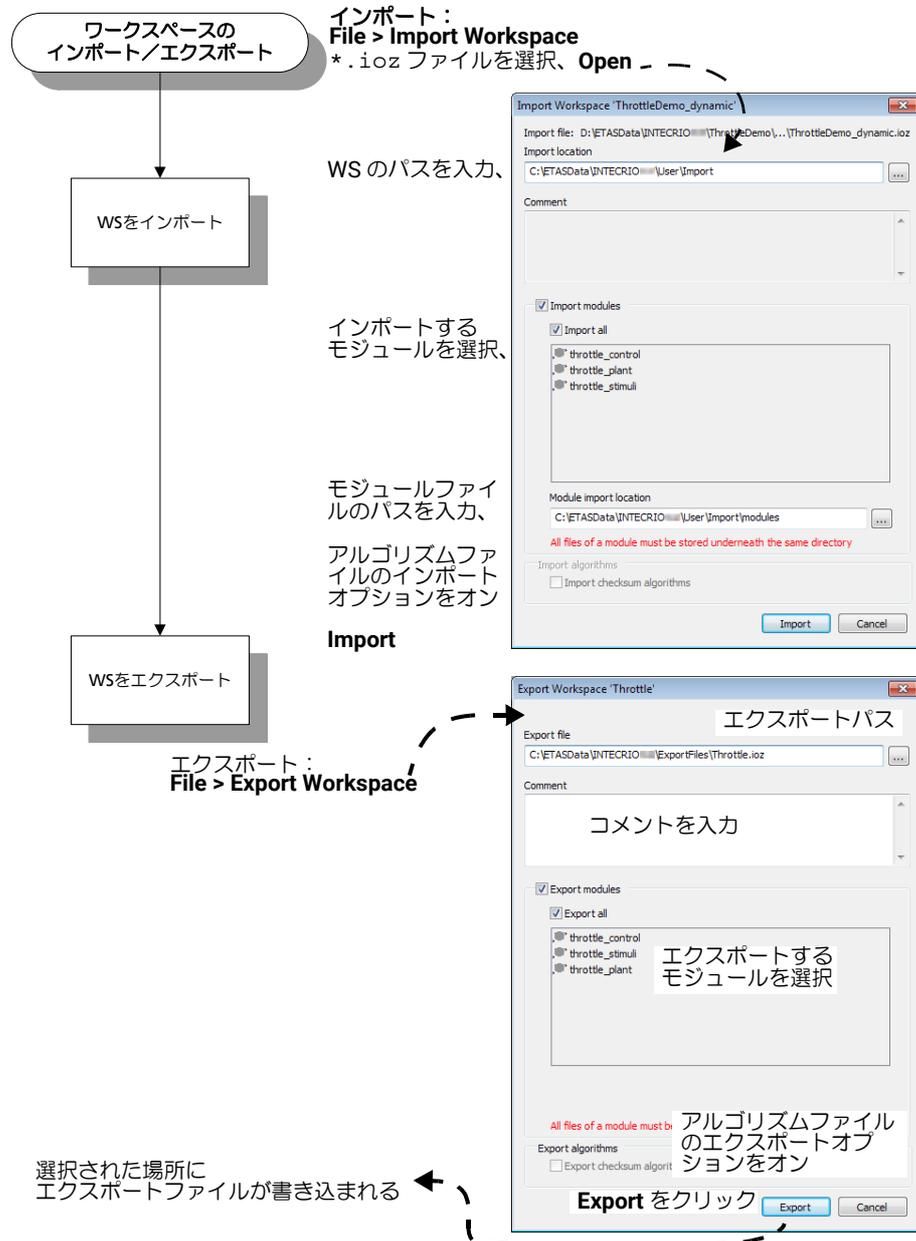


図 4-42 ワークスペースをインポート/エクスポートする

4.12.3 外部で変更されたワークスペースを開く

ワークスペースファイルを Windows エクスプローラで移動、または INTECRIO の外部でワークスペースファイルを編集したような場合には、そのワークスペースを INTECRIO で開こうとすると警告メッセージが表示されます。DTD が期限切れの場合や見つからない場合も同じメッセージが表示されます。

変更済みワークスペースを開く

File > Open Workspace、(または 、<CTRL>+ <O>)

WSを開く

↓

処理を選択

メッセージウィンドウが開く

Backup and Proceed :
WS ディレクトリ下で、Data ディレクトリのコピーを Data .bak<n> ディレクトリに格納し、WS を開く

Proceed WITHOUT Backup :
バックアップを作成せずに WS を開く

Cancel : 処理を中止する

図 4-43 外部で変更されたワークスペースを開く

上記の操作で作成されたバックアップは、以下のようにして復元できます。

バックアップファイルの復旧

Windows エクスプローラ :
...¥<workspace> ディレクトリを開き、Data フォルダを削除するか、フォルダ名を変更、

ファイルをコピー

↓

WSを開く

...¥<workspace>¥Data .bak<n> を開き、
...¥<workspace> 下に Data ディレクトリをコピー

INTECRIO :
File > Open Workspace (または 、<Ctrl>+ <O>)

図 4-44 バックアップファイルを復元する

4.13 グラフィックエディタ上のブロックのレイアウト編集

INTECRIO のプロジェクトコンフィギュレータ（PC）では、グラフィックエディタに表示されるブロックのレイアウト調整を行えます。

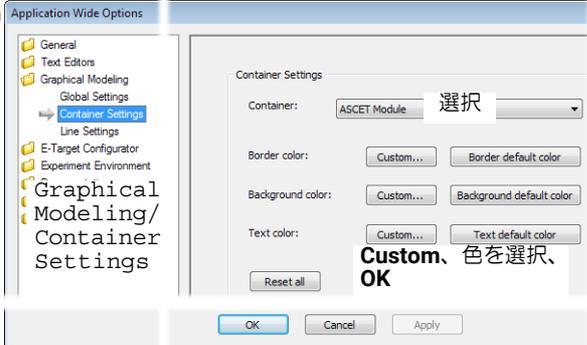
ブロックレイアウトの変更

グローバル設定を変更

ブロックレイアウトを変更

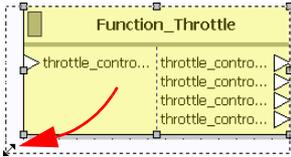
ポートを移動/ソート

Tools > Options

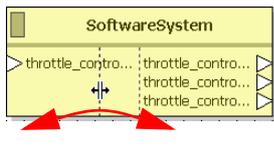


デフォルトの設定を復元：**Reset all** または *** default color**

PC：ブロックをクリック、ハンドルをドラッグしてサイズを調整



区切り線をマウスで移動



PC：ポートのショートカットメニューの **Sort > Ascending / Descending** でポートをソート、**Move Up/Down** で1つのポートを移動

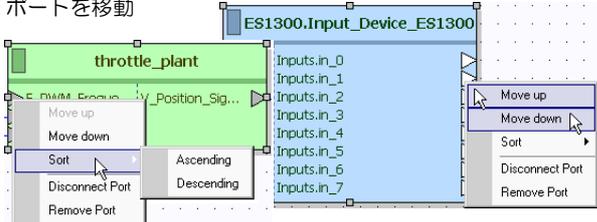


図 4-45 グラフィックエディタ上のブロックのレイアウトを編集する

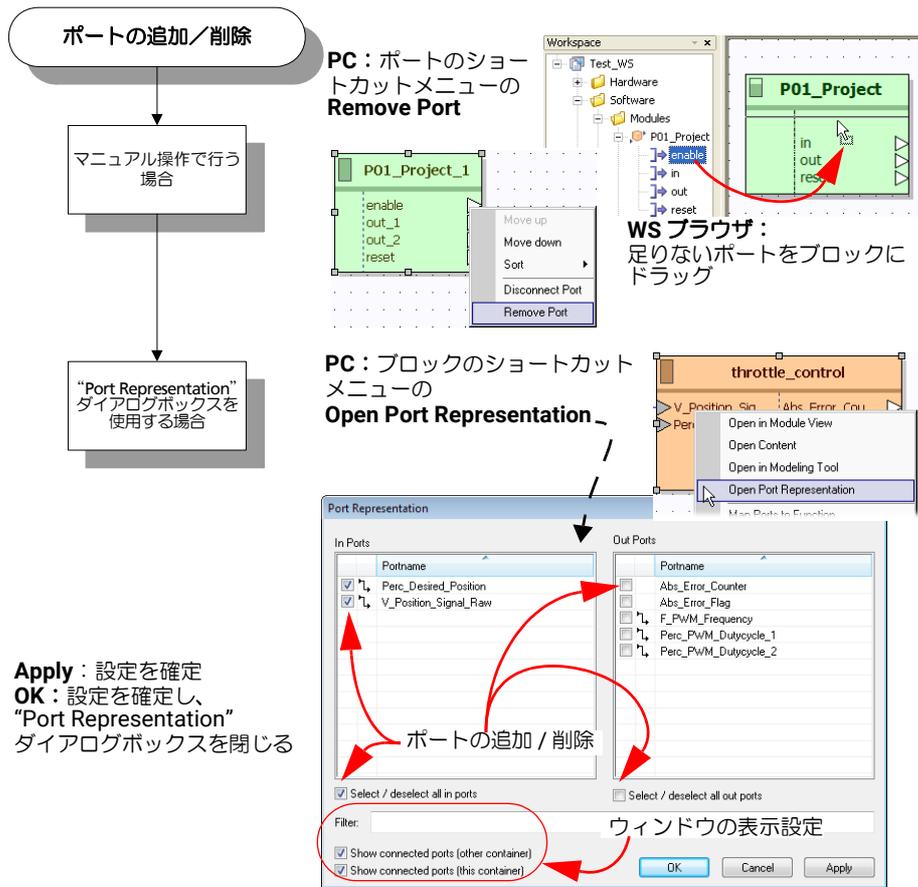


図 4-46 グラフィックエディタ上のブロックのポートを追加/削除する

4.14 ドキュメントの自動生成

INTECRIO では、システムプロジェクト（SP）のドキュメントを自動生成することができます。

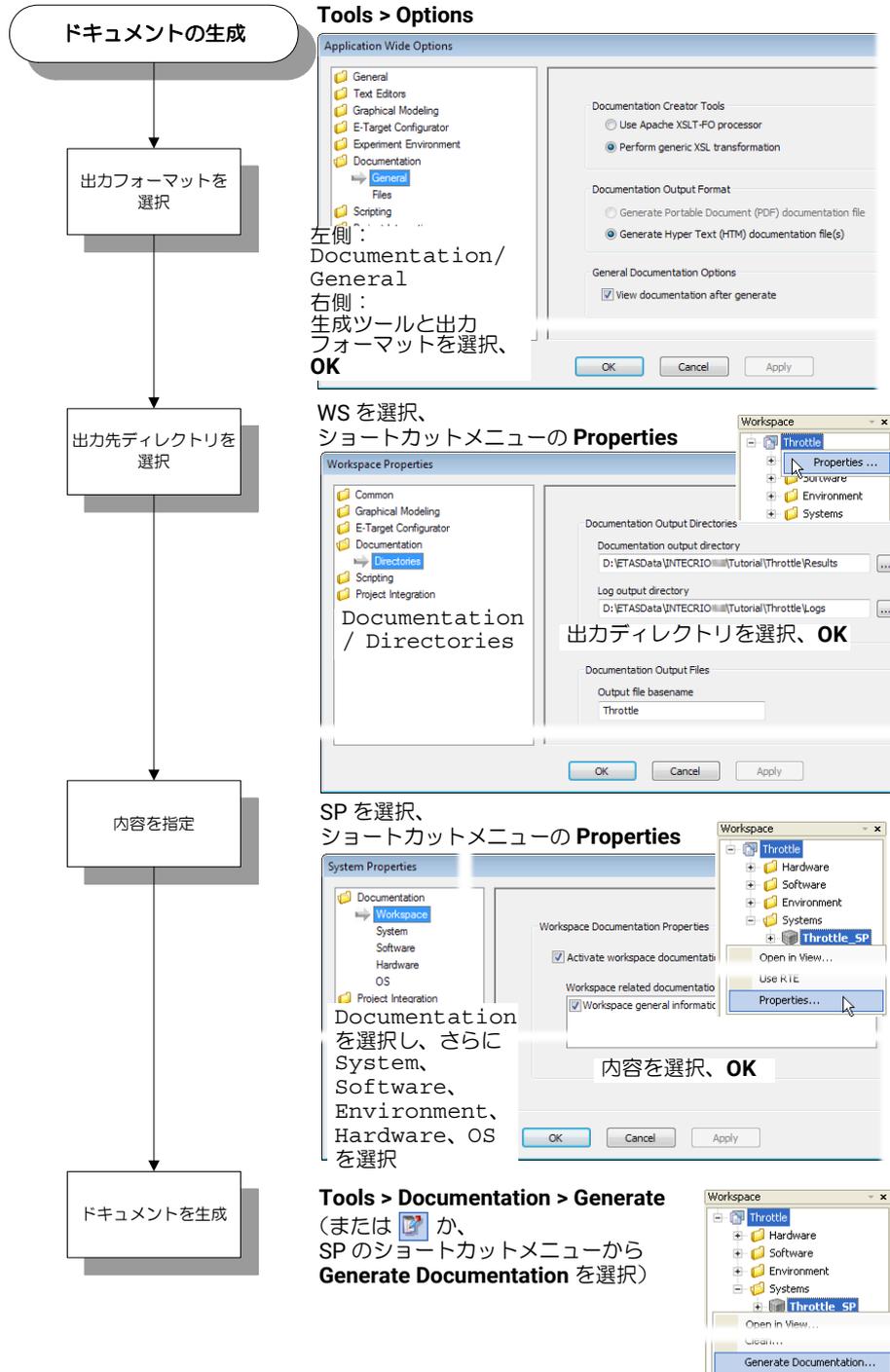


図 4-47 システムプロジェクトの自動ドキュメント生成

5 INTECRIO チュートリアル

はじめに	90
レッスン 1: INTECRIO 用モジュールの準備	96
レッスン 2: 準備	99
レッスン 3: モジュール、ファンクション、ソフトウェアシステム、環境システム	102
レッスン 4: ハードウェアシステム	111
レッスン 5: システムプロジェクト	112
レッスン 6: プロトタイプの生成	118
レッスン 7: 実験	126
レッスン 8: ES930 – 設定と操作の例	143

5.1 はじめに

本章では、スロットル制御開発の例を用いて、INTECRIO の基本的な操作方法を学習できます。このチュートリアルで学習するユーザーは、INTECRIO 自体についての詳しい知識は必要ありませんが、Windows オペレーティングシステム、および BMT（挙動モデリングツール）である MATLAB®/Simulink® に精通している必要があります。

また、チュートリアルを始める前に『INTECRIO ユーザーガイド』の「INTECRIO を理解する」という章をよく読んでご理解いただくことをお勧めします。

5.1.1 準備

実際の作業を始める前に、システムの準備が必要です。

このチュートリアルに含まれる各操作を実行するには、PC 上に INTECRIO がインストールされている必要があります。INTECRIO の起動は、デスクトップのアイコン、または Windows スタートメニューから行います。

注記

この「はじめに」の項、およびレッスン 1～7 では実際のハードウェアを使用する必要がないため、これらのレッスンはオフラインモードで実行できます。実際にハードウェアが必要なのは「レッスン 8: ES930 – 設定と操作の例」だけです。

このチュートリアルでは ETAS Experiment Environment (ETAS 実験環境) を使用しています。ES800 ハードウェアシステムをターゲットとして使用する場合は、V3.7.7 以降の ETAS Experiment Environment が必要です。

「レッスン 1: INTECRIO 用モジュールの準備」レッスン 1 では、MATLAB/Simulink (バージョン R2016a～R2021b と、INTECRIO V5.0 のリリース時点でリリースされている 32 ビット/64 ビットサービスパック) がインストールされている必要があります。旧バージョン (R2006b～R2015b) で作成された Simulink モデルは、INTECRIO へのインポートは可能です。

「レッスン 2: 準備」で必要なファイルは本製品とともに供給されているので、レッスン 1 はスキップすることができます。

以下のファイルがサンプルファイルディレクトリ (例: <sample files>\¥ThrottleDemo¥SimulinkModel、詳しくは 21 ページを参照してください) に格納されていることを確認してください。

- throttle_control.mdl

- throttle_stimuli.mdl
- throttle_plant.mdl

また、以下のファイルが throttle_*_irt_rtw というサブディレクトリに格納されていることも確認してください (* = control、stimuli、plant のいずれか)。

- throttle_*.a2l
- throttle_*.six
- throttle_*_main.c
- throttle_*_irtmacros.h

「レッスン 8: ES930 – 設定と操作の例」では以下のファイルが必要です。これらのファイルが <sample files>¥ES930Demo ディレクトリとそのサブディレクトリに格納されていることを確認してください。

- | | |
|---------------------------|------------------|
| • OutValConverter.six | • Proj_temp.six |
| • OutValConverter.a2l | • Proj_temp.a2l |
| • OutValConverterM.c | • Proj_tempM.c |
| • OutValConverterM.h | • Proj_tempM.h |
| • OutputValueConverterM.c | • TemperatureM.c |
| • OutputValueConverterM.h | • TemperatureM.h |

5.1.2 記述形式について

- 「character string と入力します。」は、character string という文字列をキーボードから入力することを意味します。ただし、以下の例外があります。

「<INTECRIO drive>¥INTECRIO5.0¥exp1.txt と入力します。」と書かれている場合、<INTECRIO drive> という文字列の代わりに INTECRIO がインストールされているドライブを表す文字列を入力してください。

たとえば、INTECRIO がドライブ C: にインストールされている場合、上記の指示は「c:¥INTECRIO5.0¥exp1.txt と入力します。」という意味になります。

- INTECRIO を起動すると表示される INTECRIO ウィンドウは、チュートリアルでは「グラフィカルフレームワーク」と呼ばれます。

5.1.3 概念

本項では、このチュートリアルに用いられている重要な用語と、その定義を紹介します。

統合

「統合」の処理には以下のステップが含まれます。

- a 制御アルゴリズムを作成するためのモデルとモデルパーツの組み立て
- b そのアルゴリズムを制御対象のハードウェアに接続
- c 実行ファイルの生成

INTECRIO

INTECRIO は、各種 BMT（挙動モデリングツール）で記述された制御アルゴリズムの各部分を結合（統合）し、さらにハードウェアシステムのコンフィギュレーション設定を行ってそのハードウェアシステムを制御アルゴリズムに接続するためのツールです。

モジュール (Module)

INTECRIO で使用される「モジュール (Module)」とは、ECU の制御機能が汎用的な形式で記述されたものを指します。モジュールは、ASCET (V5.1 以降) のプロジェクトや Simulink のモデルなどに相当します。

ASCET や Simulink でコード生成されたモデルを INTECRIO で使用するためには、C コードとともに生成されたインターフェース記述ファイル (SCOOP-IX、*.six) やデータ記述ファイル (*.a21) が必要です。モジュールをインポートする際は、インターフェース記述ファイルだけが INTECRIO に読み込まれます。データ記述ファイルと C コードが必要となるのは実行ファイルが生成される時だけです。

各モジュールは、1 つのシステムプロジェクト内において一度だけ使用できます。マルチインスタンスはサポートされていません。

ファンクション (Function)

INTECRIO の「ファンクション (Function)」は、ソフトウェアシステムを構築するための構造オブジェクトで、ファンクション自身には機能はありません。複数のモジュールを組み合わせ互いに接続し、「ファンクション」として定義することにより、各モジュールが明確に配置され、扱いやすくなります。

ソフトウェアシステム (Software System)

「ソフトウェアシステム (Software System)」は、ハードウェアに依存しないアプリケーションソフトウェアに相当し、制御アルゴリズムを構成するモジュールやファンクション、コネクションなどが含まれます。

環境システム (Environment System)

「環境システム (Environment systems)」は、仮想プロトタイピング用のプラントモデルのモデリングに用いられるものです。ソフトウェアシステムと同様にモジュールとファンクションで構成されます。

ハードウェアシステム (Hardware System)

「ハードウェアシステム (Hardware System)」には、ECU (実験ターゲット) やデバイス間インターフェースについての記述が含まれ、これによってハードウェア全体の構成や設定が定義されます。

システムプロジェクト (System Project)

「システムプロジェクト (System Project)」は、ハードウェアシステムとソフトウェアシステム、さらにシグナルマッピングやオペレーティングシステム設定を一体化したものです。実行コードの生成は、システムプロジェクトを対象として行われます。

プロトタイプ (Prototype)

「プロトタイプ (Prototype)」は、実験ターゲットシステム用の実行ファイルを指します。プロトタイプにはソフトウェアファンクションの実行コードが書き込まれていますが、これは製品用 ECU を必要としない汎用的な形式になっています。

ワークスペース (Workspace)

「ワークスペース (Workspace)」は、ソフトウェアとハードウェア、およびそれらが割り当てられたシステムプロジェクトが 1 つのファイルセットにまとめられたもので、「ワークスペースブラウザ」というツリービューにその内容が表示されます。ワークスペースは、セッション間でインポート/エクスポート、ロード/保存が可能です。ワークスペース環境から、INTECRIO の各種コンポーネントが起動されます。

グラフィカルフレームワーク

INTECRIO を起動した時に表示されるウィンドウです。INTECRIO のさまざまなコンポーネントがこのグラフィカルフレームワークに統合されています。

左上部はワークスペースブラウザで、ワークスペースのツリービューが表示されます。右上部には各種コンポーネント用のウィンドウが表示され、下部には各コンポーネントの出力メッセージが表示されるメッセージウィンドウがあります。

ソフトウェアコンフィギュレーションとプロジェクトコンフィギュレータ

システム全体を機能させるためには、INTECRIO にインポートしたモジュールを互いに接続する必要があります。またソフトウェアとハードウェアを接続するための入力と出力（シグナルシンクとシグナルソース）を作成し、ソフトウェアとハードウェアを接続します。

これらの作業は、INTECRIO のプロジェクトコンフィギュレータ機能を用いて行います。

ハードウェア接続と INTECRIO-RP / VP ハードウェアコンフィギュレータ

制御プラント（「テクニカルプロセス」とも呼ばれます）は、センサとアクチュエータのセットを表し、制御システムは、各種ハードウェアを経由してこれらのものに接続します。ここで使用されるハードウェアは、モデリングされたテクニカルプロセスに適合するよう設定する必要があります。

ハードウェア接続の作業は INTECRIO のハードウェアコンフィギュレータ（「HC」とも呼ばれます）で行います。

INTECRIO-RP のハードウェアコンフィギュレータを使用すると、該当するハードウェアシステムの構成の定義とパラメータ設定を行うことができ、ここで設定されたモデルが「コントローラ環境」にロードされ、ハードウェアシステムに組み込まれたインターフェースを制御します。

このようにして、外部ソースからのシグナルをコントローラモデル内の適切な箇所で確実に取得するためのロジックが生成されます。

INTECRIO-VP のハードウェアコンフィギュレータを使用すると、仮想プロトタイピング用ハードウェアシステムの構成の定義とパラメータ設定を行うことができ、ここで設定されたモデルが「コントローラ環境」にロードされ、VP-PC ハードウェアシステムに組み込まれたインターフェースを制御します。

このようにして仮想プロトタイピングのためのロジックが生成されます。

オペレーティングシステムのコンフィギュレーション設定

リアルタイムプロトタイプを作成する場合、オペレーティングシステムのコンフィギュレーション設定は非常に重要な作業です。オペレーティングシステムはプロセッサをめぐって競合するタスクとプロセスの処理シーケンスを決定し、必要に応じてタスクの実行を切り替えます。その際に必要な設定は、すべてオペレーティングシステムのコンフィギュレーション設定時に行われます。

OS コンフィギュレータ (OSC) と OSC エディタ

オペレーティングシステムのコンフィギュレーション設定は、INTECRIO の OSC (OS コンフィギュレータ) エディタで行います。OSC エディタは OS コンフィギュレータのユーザーインターフェースで、システムの概要を素早く把握でき、アプリケーションからの視点でコンフィギュレーションを編集することができます。

ビルド処理とプロジェクトインテグレータ

プロジェクトインテグレータ (PI) は、システム全体、つまりモジュールとファンクション、ハードウェア接続、OS コンフィギュレーションなどを 1 つの実行ファイル (「プロトタイプ」と呼ばれます) にまとめます。この際、実験用の ASAM-MCD-2MC ファイルが生成され、その中に各モジュールの ASAM-MCD-2MC ファイルの内容とプロジェクト全体についての追加情報が格納されます。

実験

ETAS Experiment Environment を使用して、プロトタイプの機能の検証実験を行います。実験を行う際は、さまざまなタイプの測定/適合インストールメントを使用できます。

実験はハードディスクに保存されますが、ワークスペースブラウザから開くことができます。適切な実験をロードすることにより、特定の作業用の実験環境を素早くセットアップすることができます。

仮想プロトタイピング (Virtual Prototyping - バーチャルプロトタイピング)

車両の電子制御ファンクションの仮想プロトタイプを作成し、PC 上で実行してテストを行います。仮想プロトタイプは以下のコンポーネントで構成されます。

- 自動車用組み込みソフトウェア (アプリケーションソフトウェアと OSEK オペレーティングシステム)
- プラントモデル (環境システム)

仮想プロトタイピングの手法を用いることにより、開発工程の初期段階において、システムモデルを用いた Model-in-the-Loop (MiL) 環境を構築し、ファンクションを検証することができます。これによって、早い段階でのファンクションとシステムの並行開発が実現でき、開発工程全体の効率が向上します。

デジチェーンコンフィギュレーション

一般的なインターフェースとは異なり、デジチェーンは INTECRIO 内で設定することができません。専用のデジチェーンコンフィギュレーションツールを用いて設定し、ツールで作成したコンフィギュレーションファイルを INTECRIO にインポートします。

ES930

ES930 マルチ I/O モジュールは、コンパクトでありながら耐久性に富んだパワフルなツールです。各種入出力チャンネルを搭載しています。

このモジュールで ES910 の機能を拡張することにより、ファンクションモデル (Simulink[®]、ASCET、AUTOSAR、C コード) から直接センサを読み取ったりアクチュエータを制御したりすることが可能になります。

5.1.4 チュートリアルの概要と目標

チュートリアルでは、INTECRIO で行われるさまざまな作業について、スロットル制御やその他の例を用いて実習します。

ガソリンエンジンの混合気中の空気量は、スロットルバルブの位置 (角度) によって変化します。つまりスロットルバルブは、エンジン制御ファンクションの一部として機能します。

エンジン制御ファンクションは、たとえばエアフロー (つまりはスロットルバルブの動作) によってエンジン出力を調整し、所定のエンジン回転数におけるエンジントルクを制御します。スロットルバルブが完全には開いていない場合、エン

ジンに吸入される空気量が絞られ、生成されるトルクが減少します。この「スロットル効果」はスロットルバルブの開度で決まる断面積に依存します。最大エンジントルクはスロットルバルブを全開にした時に得られます。

チュートリアルの各レッスンは、このような制御系のプロトタイプ開発を行う際の一般的な手順に沿って構成されています。

「レッスン 1：INTECRIO 用モジュールの準備」：スロットル制御の Simulink モデルについて必要な設定を Simulink 上で行い、INTECRIO 用に適したコードを生成します。

「レッスン 2：準備」：ワークスペースを作成しセットアップします。

「レッスン 3：モジュール、ファンクション、ソフトウェアシステム、環境システム」：スロットル制御に属するモジュールを INTECRIO にインポートし、続いて、2つのモジュールを結合して1つのファンクションを定義します。さらにファンクション内でシグナル接続を確立し、外界と接続するためのインターフェースを作成します。次にソフトウェアシステムを作成してセットアップし、最後に環境システムを作成して3つのモジュールを追加します。

「レッスン 4：ハードウェアシステム」：仮想プロトタイピング用ハードウェアシステムを作成します。

「レッスン 5：システムプロジェクト」：システムプロジェクトを作成します。まずハードウェアシステム、ソフトウェアシステム、環境システムをシステムプロジェクト内に組み込み、続いてソフトウェアシステムと環境システムを接続します。

「レッスン 6：プロトタイプの生成」：オペレーティングシステムのコンフィギュレーション設定を行い、さらにコード生成オプションを設定して、スロットル制御用実行プロトタイプを生成します。

「レッスン 7：実験」：作成したプロトタイプを用いて実験を行います。

「レッスン 8：ES930 – 設定と操作の例」：小規模な ES930 プロジェクトをセットアップして実験を行います。

チュートリアルで習得できること

チュートリアルを終えると、以下の操作が行えるようになります。

- モジュールを INTECRIO にインポートし、それらを結合してソフトウェアシステムを作成する。
- ハードウェアシステムを作成してパラメータ設定を行い、ソフトウェアシステムと結合してシステムプロジェクトを作成する
- オペレーティングシステムのコンフィギュレーションを設定し、実行プロトタイプを生成してその実験を行う

これらの基本的なステップを学習すれば、その後はオンラインヘルプを参照しながらさらに詳細な機能を利用できるようになります。またワークスペースと実験のサンプルが用意されているので、それらを利用して実際の作業を手早く開始することもできます。

5.2 レッスン 1：INTECRIO 用モジュールの準備

スロットル制御のファンクションソフトウェアは、BMT（挙動モデリングツール）である Simulink 上で作成された複数のモジュールで構成されます。INTECRIO で処理可能なコードを BMT で生成するには、コード生成用のオプション設定が必要です。

目標：

スロットル制御の Simulink モデル (throttle_control.mdl) 用オプションを Simulink 上で設定し、INTECRIO で扱えるコードを生成します。生成したコードは以降のレッスンで使用されます。

注記

INTECRIO で扱える ASCET モジュールを作成する方法は、ASCET のマニュアルで説明しています。

モジュール自体を記述する作業は、このチュートリアルに含まれていません。さらに学習を行いたい場合は、別の Simulink モデル (throttle_stimuli.mdl、throttle_plant.mdl) を使用してこのステップを反復練習してください。

5.2.1 本レッスンで利用する重要な概念

モジュール (Module)

INTECRIO で使用される「モジュール (Module)」とは、ECU の制御機能が汎用的な形式で記述されたものを指します。モジュールは、ASCET (V5.1 以降) のプロジェクトや Simulink のモデルなどに相当します。

ASCET や Simulink でコード生成されたモデルを INTECRIO で使用するためには、C コードとともに生成されたインターフェース記述ファイル (SCOOP-IX、*.six) やデータ記述ファイル (*.a21) が必要です。

5.2.2 Simulink® モデル

本項で使用する throttle_control.mdl という Simulink モデルは、INTECRIO のサンプルファイルディレクトリ (<sample_files>¥ThrottleDemo¥SimulinkModel) に格納されています。このモデルにはすでに機能が定義されているので、ここではこのモデルを開いて INTECRIO で使用するために必要なシミュレーションパラメータを設定し、コードを生成します。

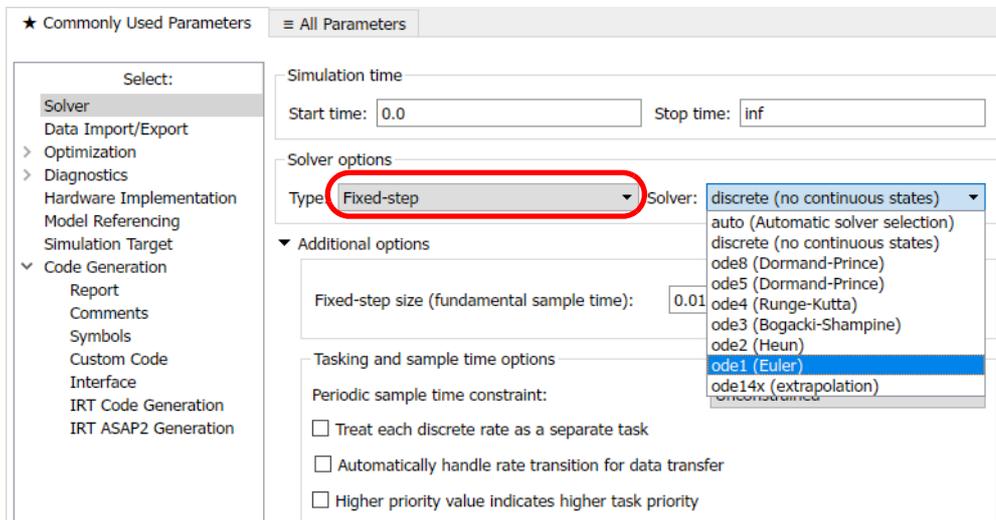
注記

本項では、MATLAB/Simulink R2016b のユーザーインターフェースを用いて説明しています。他のバージョンをご使用の場合はユーザーインターフェースが多少異なります。

操作手順：Simulink モデルのコードを生成する

1. モデルのバックアップコピーと throttle_control_irt_rtw フォルダを作成します。
2. Simulink で throttle_control.mdl モデルを開きます。

3. Simulink で “Configuration Parameters” ダイアログボックスを開きます。
4. “Simulation Parameters” ダイアログボックスの “Solver” ノードを開きます。
5. “Fixed-step” タイプのソリューションアルゴリズムを選択します。



6. "Code Generation" ノードで `irt.tlc` ターゲットを選択します。

`irt_default_tmf` テンプレートが自動的に選択されます。
ターゲットとテンプレートの組み合わせは `ier.tlc` と `ier_default_tmf` でも可能ですが、`irt*` と `ier*` を組み合わせることはできません。

Generate code only オプションは、INTECRIO ターゲットについては暗黙的に設定されるので、手動操作でオンにする必要はありません。

注記

各オプションが上図のように設定されていないと、INTECRIO 用コードは生成されません。

7. External mode インターフェイスがオフ（無効）になっていることを確認します。

External mode がオンになっていると、ASAM-MCD-2MC ファイルにパラメータが書き込まれなくなってしまいます

8. コードを生成します。

Simulink モデルが格納されているディレクトリ内に、生成されるファイルを格納するためのサブディレクトリ `hrottle_control_irt_rtw` が作成されます。生成されるファイルのうち、INTECRIO での作業に必要なのは以下のファイルです。

- `throttle_control.a2l`
INTECRIO での実験用に生成される ASAM-MCD-2MC ファイルです。
- `throttle_control.six`
インターフェース記述ファイル、つまり SCOOP-IX ファイルです。
- `throttle_control_main.c`
このファイルには、INTECRIO で統合を行う ECU ファンクションの主要部分が含まれています。ファンクションは複数の C 関数（void/void 関数）からなり、これらの関数は、INTECRIO において「プロセス」としてオペレーティングシステムに関連付けられます。
- `throttle_control_irtmacros.h`
このヘッダファイルには複数のモジュールに用いられる複数の `#define` 文が含まれています。このファイルは、`gcc -include throttle_control_irtmacros.h` というコマンドにより、他のファイルよりも先に処理されます。

`throttle_control_irt_rtw` サブディレクトリには、Simulink と Simulink® Coder™ によるコード生成時に生成されるその他のファイルも格納されますが、これらのファイルは INTECRIO での作業には無関係です。これらのファイルの内容については各ユーザーマニュアルを参照してください。

5.3 レッスン 2：準備

レッスン 1 で、実験に必要なファイルを生成したので、次に INTECRIO での作業準備を行います。まず INTECRIO を起動し、使用するワークスペースを開きます。作業を明確化するため、ここではチュートリアル専用のワークスペースを使用することをお奨めします。ここでセットアップしたワークスペースを元に、以降の作業を進めていきます。

目標：

チュートリアルに必要なワークスペースを作成し、セットアップします。

5.3.1 本レッスンで利用する重要な概念

グラフィカルフレームワーク

INTECRIO を起動した時に表示されるウィンドウです。INTECRIO のさまざまなコンポーネントがこのグラフィカルフレームワークに統合されています。

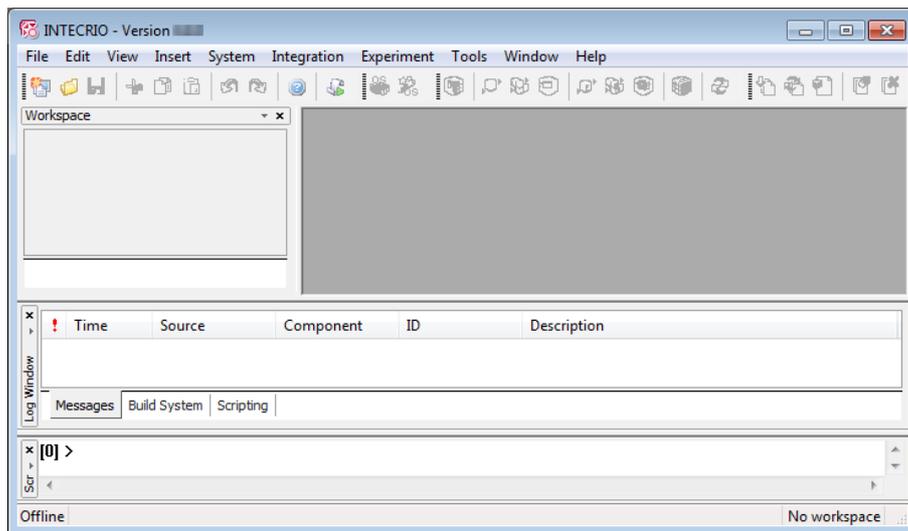
ワークスペース (Workspace)

「ワークスペース (Workspace)」には INTECRIO での作業時に作成されたすべてのデータが含まれます。INTECRIO の各種コンポーネントにはワークスペースブラウザからアクセスします。

5.3.2 INTECRIO とワークスペース

プログラムを起動（デスクトップの **INTECRIO V5.0** アイコン、または Windows スタートメニューの **ETAS INTECRIO 5.0** を選択）すると、INTECRIO のグラフィカルフレームワークが開きます。この際、“Log Window” 内の

“Messages” タブ上には、実際に起動された INTECRIO のコンポーネントがリストアップされ、それ以外のフィールドには何も表示されません。そして、新しいワークスペースを作成するかまたは既存のワークスペースを開くと、そのワークスペースがワークスペースブラウザ (“Workspace” ウィンドウ、デフォルト状態ではメインウィンドウの左側にドッキングされています) にツリー構造で表示されます。INTECRIO のさまざまな機能を実行する各コンポーネントは、このワークスペースブラウザまたはメニューバーから呼び出します。右上部のフィールドにはさまざまなデータを編集するための各種エディタが表示されます。



まず、メニューコマンド **Tools > Options** を使用してオプションウィンドウを開いてください。ここではシステム全体についてのオプション設定（プロジェクトインテグレータやテキストエディタに関する設定など）を行うことができます。各オプションについては、オンラインヘルプに詳しく説明されています。

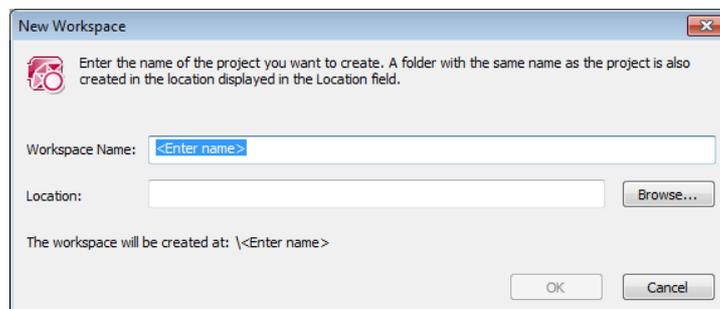
次にワークスペースを作成します。ワークスペースには、INTECRIO での作業時に作成されたデータが階層的に格納されます。ワークスペースはツリービューで表示され、ここからモジュールの統合、ファンクションの定義、ハードウェアコンフィギュレーションとシステムプロジェクトの作成などを行います。これらのアイテムの内容を定義するために使用される各種エディタも、すべてツリービューから呼び出します。

操作手順：ワークスペースを作成する



1. **New Workspace** ボタンをクリックします。

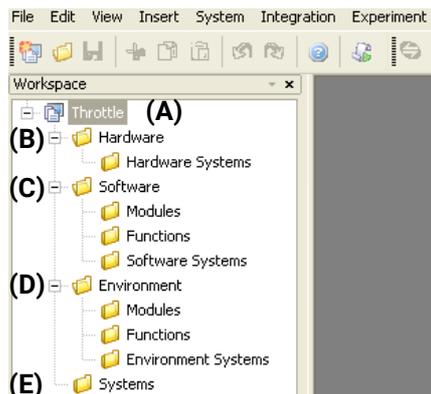
“New Workspace” ダイアログボックスが開きます。



2. “Workspace Name” フィールドに、Throttle という名前を入力します。

3. “Location” フィールドでワークスペースを保存するパス（例：
D:\¥ETASData¥INTECRIO5.0¥Tutorial）を指定しま
す。
4. **OK** をクリックします。

上記の操作により INTECRIO 上に Tutorial という名前のワークスペースが開き、同時にこのワークスペースが、ETASData¥INTECRIO5.0¥Tutorial¥Throttle ディレクトリ（上記のパスを使用した場合）に Throttle.iow という名前で保存されます。ワークスペースブラウザにはワークスペースのツリー構造が表示されますが、この時点ではまだデフォルトアイテムしか表示されません。



ワークスペースブラウザのデフォルトアイテムは以下のとおりです。

- (A) トップアイテム – ワークスペースの名前です。
- (B) Hardware – このフォルダに、ワークスペース内のすべてのハードウェアシステムが格納されます。各システムに含まれるハードウェアアイテムは、ハードウェアコンフィギュレータ（HC）により組み込まれ、Hardware Systems というサブフォルダに格納されます。
- (C) Software – 各サブフォルダに以下のものが格納されます。
 - Modules サブフォルダ：プロジェクトに属するソフトウェアモジュールと AUTOSAR ソフトウェアコンポーネント
 - Functions サブフォルダ：関連するモジュールを集めて構成されるファンクション
 - Software Systems サブフォルダ：ソフトウェア全体を表すソフトウェアシステム
- (D) Environment – このフォルダに、プロジェクトに属する環境モジュール（Modules サブフォルダ）、関連するモジュールを集めて構成されるファンクション（Functions サブフォルダ）、そして環境全体を表す環境システム（Software Systems サブフォルダ）が格納されます。
- (E) Systems – このフォルダに、システムプロジェクト、つまり実験に使用されるハードウェアとソフトウェアへの参照、ソフトウェアとハードウェアの接続情報、オペレーティングシステムのコンフィギュレーションが格納されます。

ワークスペース内にシステムプロジェクトを作成するには、Systems フォルダ内に所定のアイテムが作成されている必要があります。

ワークスペースのプロパティを設定するには、ワークスペースブラウザに表示されているワークスペース名を右クリックし、ショートカットメニューから **Properties** を選択して “Options” ダイアログボックスを開きます。設定オプションについての情報は、オンラインヘルプに記載されています。

最後にこのワークスペースを保存します。以降の各レッスンにおいても、最後に必ずワークスペースを保存してください。

5.4 レッスン 3：モジュール、ファンクション、ソフトウェアシステム、環境システム

目標：

スロットル制御に必要なモジュールを INTECRIO にインポートし、2つのモジュールを組み合わせてファンクションを作成します。ファンクション内でシグナル接続を確立し、さらに外部接続用インターフェースを作成します。続いてソフトウェアシステムを作成してファンクションを割り当て、最後に環境システムを作成して3つのモジュールを追加します。

5.4.1 本レッスンで利用する重要な概念

モジュールと環境モジュール

INTECRIO で使用される「モジュール (Module)」とは、ECU の制御機能が汎用的な形式で記述されたものを指します。モジュールは、ASCET (V5.1 以降) のプロジェクトや Simulink のモデルなどに相当します。

「環境モジュール」には仮想プロトタイピング用プラントモデルの汎用的な機能を記述したものが含まれます。

ファンクション (Function)

INTECRIO の「ファンクション (Function)」は、ソフトウェアシステムを構築するための構造オブジェクトで、ファンクション自身には機能はありません。複数のモジュールを組み合わせて互いに接続し、「ファンクション」として定義することにより、各モジュールが明確に配置され、扱いやすくなります。

ソフトウェアシステム (Software System)

「ソフトウェアシステム (Software System)」は、ハードウェアに依存しないアプリケーションソフトウェアに相当し、制御アルゴリズムを構成するモジュールやファンクション、コネクションなどが含まれます。

環境システム (Environment System)

「環境システム (Environment systems)」は、仮想プロトタイピング用のプラントモデルのモデリングに用いられるものです。ソフトウェアシステムと同様にモジュールとファンクションで構成されます。

ソフトウェアコンフィギュレーションとプロジェクトコンフィギュレータ

システム全体を機能させるためには、INTECRIO にインポートしたモジュールを互いに接続する必要があります。またソフトウェアとハードウェアを接続するための入力と出力 (シグナルシンクとシグナルソース) を作成し、ソフトウェアとハードウェアを接続します。

これらの作業は、INTECRIO のプロジェクトコンフィギュレータ機能を用いて行います。

5.4.2 モジュールのインポート

システムの挙動を決定する「モジュール」は、INTECRIO で処理できるモデルの最小単位で、インスタンス生成できません。つまりモジュールは、直接、またはファンクションの一部として、1つのシステムプロジェクト内に1回のみ割り当てられます。

スロットル制御に属するモジュールは、INTECRIO を PC にインストールした際にサンプルファイルディレクトリ (<sample files>¥ThrottleDemo¥SimulinkModel) にコピーされます。レッスン 1 をスキップした場合、そのサブディレクトリ throttle_control_irt_rtw に格納されたファイルを使用してレッスン 2 を行うことができます。

INTECRIO の観点で見たモジュールの基本構造はどれも同じで、以下のインターフェースが備わっています。

- シグナルシンク（入力）
- シグナルソース（出力）
- 起動インターフェース（プロセス）

モジュールをインポートする際には、インターフェース記述ファイル (*.six) だけが読み込まれます。実際の機能が記述されている *.c ファイルや *.h ファイルは、実行ファイルを生成する際にのみ使用されます (5.7 項)。ASAM-MCD-2MC ファイルも、実行ファイル生成時まで必要ありません。

操作手順: モジュールをインポートする

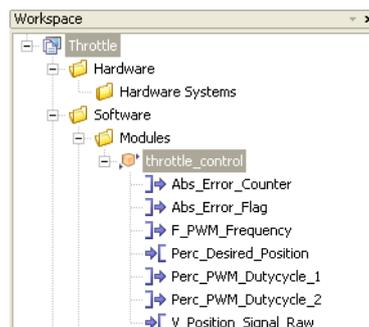
INTECRIO では、各モジュールは Software フォルダ下に保存されます。

1. ワークスペースブラウザに表示されている Modules フォルダを右クリックし、ショートカットメニューから **Import Module** を選択します。

ファイル選択ダイアログボックスが開きます。

2. <sample files>¥ ThrottleDemo¥SimulinkModel¥ throttle_control_irt_rtw というパスから throttle_control.six というファイルを選択します。
3. **Open** をクリックします。

モジュールがワークスペース内にインポートされ、ワークスペースブラウザに表示されます。



4. 同様に、throttle_stimuli という Simulink モデルもインポートします。

ツリービューを見ると、このモジュールの下にシグナルソース (]→) とシグナルシンク (→[) が表示されています。

操作手順: 環境モジュールをインポートする

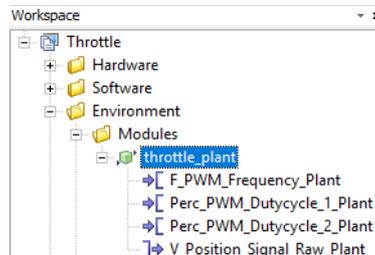
INTECRIO では、各環境モジュールは Environment フォルダ下に保存されます。

1. ワークスペースブラウザに表示されている Environment / Modules フォルダを右クリックし、ショートカットメニューから **Import Module** を選択します。

ファイル選択ダイアログボックスが開きます。

2. throttle_plant というファイルを選択します。
3. **Open** をクリックします。

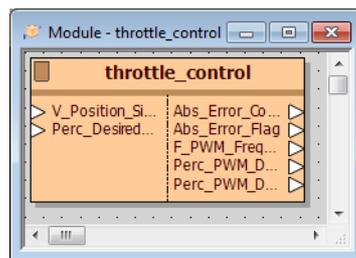
モジュールがワークスペース内にインポートされ、ワークスペースブラウザに表示されます。



操作手順: モジュールを表示する

モジュールと環境モジュールは、グラフィックエディタで表示することができます。

1. throttle_control モジュールを右クリックし、ショートカットメニューから **Open in View** を選択します。
グラフィックエディタが開き、このモジュールが表示されます。



シグナルシンクはブロックの左端に配置され、シグナルソースは右端に配置されます。どちらも白抜き三角形で示され、三角形の隣にエレメント名が表示されます。シグナルシンクまたはシグナルソースの名前が長くて表示しきれない場合は後ろの部分が省略されますが、カーソルでその名前をポイントすると名前全体がツールチップに表示されます。起動インターフェースは表示されません。

5.4.3 ファンクションの作成

ファンクションは、それ自体には機能を持たない純粋な構造化オブジェクトです。関連するモジュールを「ファンクション」としてまとめることにより、モデルの視認性や再利用性が向上します。

ファンクションは以下のコンポーネントで構成されます。

- 1つ以上のモジュール
- ファンクション内の各モジュールの入力と出力を結ぶコネクション
- ファンクションインターフェース（入力、出力、起動インターフェース）

ファンクションの入力と出力にデータやインプリメンテーション（実装情報）を持たせることはできません。ファンクションの入力と出力は手動操作で追加し、入力はファンクション内のモジュールの1つまたは複数のシンクに接続でき、出力は1つのモジュールの1つのソースだけに接続できます。

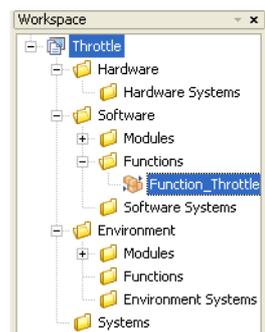
Software フォルダと Environment フォルダは完全に区別されています。Software フォルダ内のモジュールを Environment フォルダ内のファンクションに挿入することはできず、その逆もできません。

ここではまず、Software フォルダにファンクションを作成します。

操作手順: 空のファンクションを作成する

1. ワークスペースブラウザで Functions フォルダを右クリックし、ショートカットメニューから **Create Function** を選択します。
2. “Create Function” 入力ダイアログボックスに Function_Throttle という名前を入力し、**OK** をクリックします。

Function_Throttle ファンクションがワークスペースブラウザの Functions フォルダに作成されます。

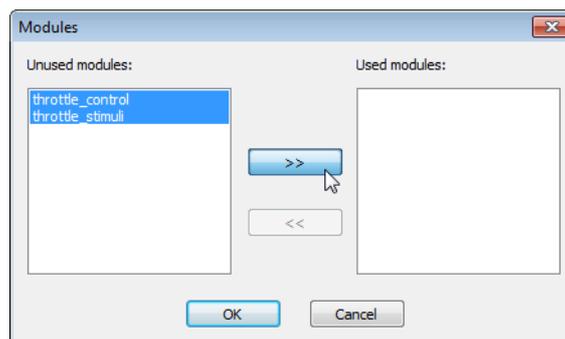


次に、このファンクションにモジュールを組み込みます。

操作手順: モジュールを追加する

1. ワークスペースブラウザで Function_throttle ファンクションを右クリックし、ショートカットメニューから **Modules** を選択します。

“Modules” ダイアログボックスが開きます。Software フォルダ内のモジュールがすべて “Unused modules” フィールドに表示されます。



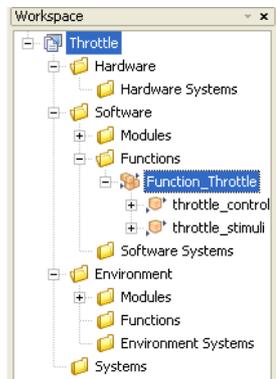
2. “Unused modules” フィールドから throttle_control と throttle_stimuli というモジュールを選択します。



3. >> ボタンをクリックします。
選択した 2 つのモジュールが “Used modules” フィールドに移動します。

4. **OK** をクリックします。

選択した 2 つのモジュールがファンクションに追加され、ワークスペースブラウザ内のファンクション用サブフォルダ内に表示されます。



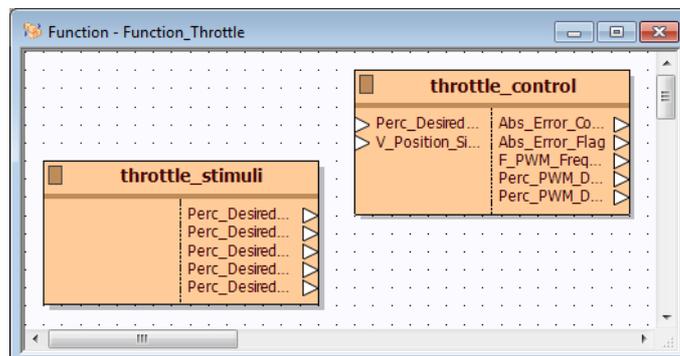
これで、2 つのモジュールがファンクションの一部となったので、次に、シグナルフローを実現する「コネクション」を明示的に確立する必要があります。INTECRIO では、たとえば同名のシグナルソースとシグナルシンクが自動的に接続されるというような暗黙的な接続は一切行われません。

ラピッドプロトタイピングにおいては「静的コネクション」と「動的コネクション」が使用され、後者はプログラム実行時に修正できます。このチュートリアルでは動的コネクションを使用しますが、実験において実際にその接続を変更する操作は行いません。「動的コネクション」や「コネクション」の扱い全般についての詳しい情報は、ユーザーガイドとオンラインヘルプを参照してください。

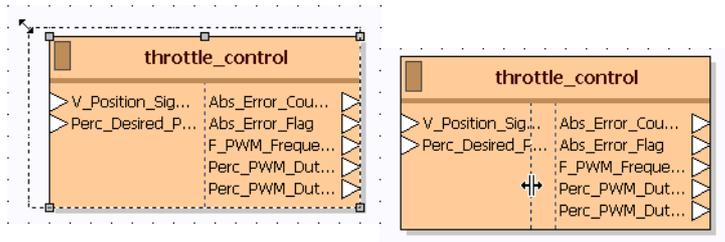
コネクションの作成はグラフィックエディタのファンクションビューで行います。新しく作成されるコネクションは、デフォルトでは動的コネクションとして作成されます。

操作手順: モジュールを接続する

1. グラフィックエディタで Function_throttle ファンクションを開きます。
2. 各モジュールの位置を調節し、接続しやすいように以下のように配置します。

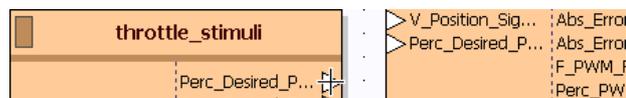


3. 必要に応じて以下のように表示を変更します。

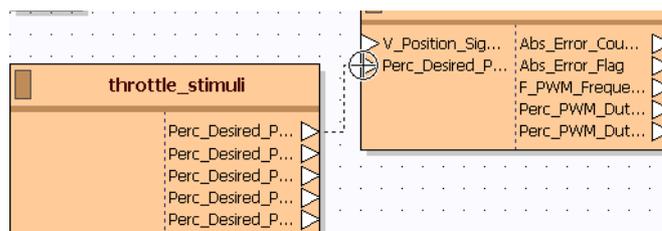


4. カーソルを throttle_stimuli モジュールの Perc_Desired_Position というシグナルソースに合わせます。

カーソルが十字カーソルに変わります。

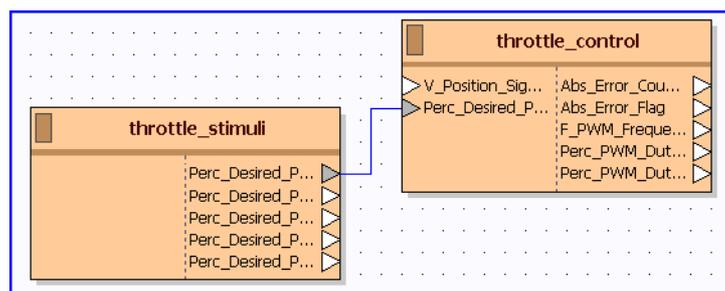


5. マウスの左ボタンを押し下げ、そのままシグナルソース Perc_Desired_Position から throttle_control モジュールの同名のシグナルシンクまで線を引きます。



6. マウスボタンから指を離すと接続が確立されます。

シグナルソースとシグナルシンクの間「コネクション」が実線で描かれ、両端の三角形は塗りつぶされます。コネクション自体は、ワークスペースブラウザには表示されません。



ファンクション内のその他のコネクションは以降のステップで定義しますが、ここでは先に、ファンクション外部とのインターフェースを定義する方法を説明します。接続されていないシグナルソースとシグナルシンク（上図の白抜きの三角形）が自動的にファンクションインターフェースとなるのではないため、手動操作でインターフェースを作成する必要があります。

これを行うには、ファンクションインターフェース（ファンクションのシグナルソースおよびシグナルシンク）を追加し、それらをモジュールのシグナルソースやシグナルシンクに接続します。ファンクションインターフェースにはデータやインプリメンテーションは定義されません。ファンクションの各シグナルシンク（図 5-1 の Sink 1 ポートを参照してください）はモジュールの 1 つまたは複数のシグナルシンクに接続でき、ファンクションの各シグナルソース（図 5-1 の

Source 1 ポートを参照してください) はファンクション内の 1 つのモジュールの 1 つのシグナルソースにのみ接続できます。ファンクションの 1 つのシグナルソースに 2 つ以上のコネクションを定義することはできません (図 5-1 の赤丸の部分)。

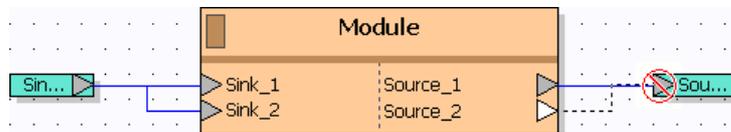


図 5-1 モジュールとポートの接続

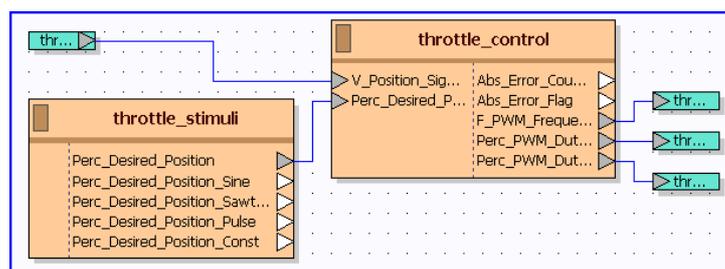
操作手順: ファンクションインターフェースを追加する

1. グラフィックエディタ内の空白部分を右クリックし、ショートカットメニューから **Create Port > Scalar In Port** を選択します。

入力ポートが生成されます。



2. 入力ポートを、適切な位置にドラッグします。
3. throttle_control モジュールの V_Position_Signal_Raw というシグナルシンクに入力ポートを接続して、ファンクションのシグナルシンクを作成します。
入力ポートの三角形が塗りつぶされます。入力ポートには接続先のシンクの名前が付きます。
4. **Create Port > Scalar Out Port** で出力ポート (out) を作成し、それを throttle_control モジュールの F_PWM_Frequency というシグナルソースに接続します。
これでファンクションのシグナルソースが作成されました。
5. ファンクションのその他のインターフェースを下図のように生成します。



これでファンクションが定義されたので、次のステップでこれを「ソフトウェアシステム」に組み込みます。

なお、グラフィックエディタの編集機能についての詳細や、ここまでの作業を別の操作で行う方法は、オンラインヘルプを参照してください。

5.4.4 ソフトウェアシステムと環境システムの作成

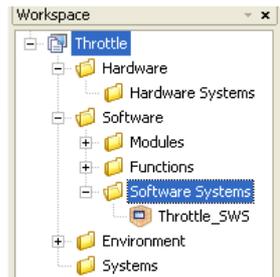
仮想プロトタイピング環境には、ECU 制御ファンクションのハードウェア非依存部分が含まれる「ソフトウェアシステム」と、プラントモデルが含まれる「環境システム」が必要です。

ここではまず、空のソフトウェアシステムを作成します。

操作手順: ソフトウェアシステムを作成する

1. ワークスペースブラウザに表示されている Software Systems フォルダを右クリックし、ショートカットメニューから **Create Software System** を選択します。
“Create Software System” ダイアログボックスが開きます。
2. Throttle_SWS という名前を入力して **OK** をクリックします。

空のソフトウェアシステムが作成されます。

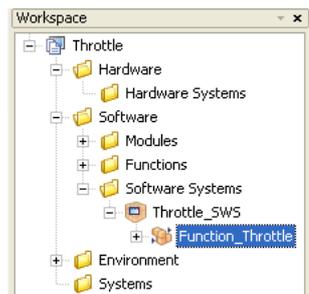


次に、スロットル制御のファンクションをこのソフトウェアシステムに追加します。この作業はワークスペースブラウザでドラッグアンドドロップ操作により行います。

操作手順: モジュールとファンクションを追加する

1. ワークスペースブラウザの Software¥Functions フォルダにある Function_throttle ファンクションを Throttle_SWS というソフトウェアシステムまでドラッグします。

このファンクションがソフトウェアシステムに割り当てられます。ファンクションは、ソフトウェアシステムにより「参照」されます。

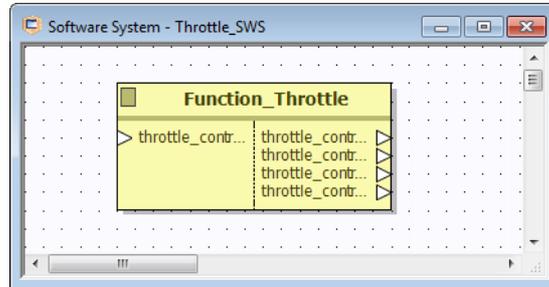


次に、ソフトウェアシステム内にハードウェアとのインターフェースを作成します。これには、ファンクションを編集する際に用いたグラフィックエディタを使用します。

操作手順：ソフトウェアシステムにインターフェースを追加する

1. Throttle_SWS ソフトウェアシステムをダブルクリックします。

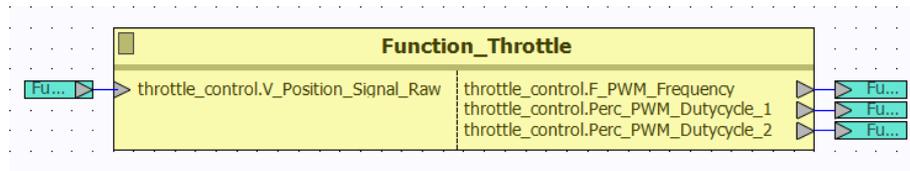
グラフィックエディタが開き、ソフトウェアシステムが表示されます。



ここで、すべてのシグナルソース/シンクにハードウェア接続用の外部インターフェースを追加し、それらをファンクションに接続します。

2. ファンクションのショートカットメニューから **Map Ports to Software System** を選択します。

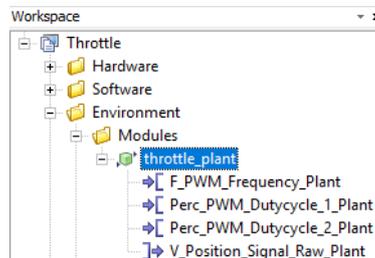
すると、下図のような結果になります。



最後に、環境システムを作成してセットアップします。

操作手順：環境システムを作成してセットアップする

1. ワークスペースブラウザの Environment Systems フォルダを右クリックして、ショートカットメニューから **Create Environment System** を選択します。
2. 環境システムの名前（例：Environment）を指定します。
3. 環境システムに throttle_plant モジュールを追加します。
4. モジュールの各ポートを環境システムに接続します。



5.5 レッスン 4：ハードウェアシステム

目標：

仮想プロトタイピング用の「ハードウェアシステム」を作成します。仮想プロトタイピングにおいては PC を実験用ハードウェアとして使用するので、ハードウェアシステムの作成はとてもシンプルです。ラピッドプロトタイピング用のハードウェアデバイスの場合はオプション設定の項目数が多くなりますが、それについては 5.9.3 項「ES910 ハードウェアシステムの作成とセットアップ」に説明されています。

5.5.1 本レッスンで利用する重要な概念

ハードウェアシステム (Hardware System)

「ハードウェアシステム (Hardware System)」には、ECU (実験ターゲット) やデバイス間インターフェースについての記述が含まれ、これによってハードウェア全体の構成や設定が定義されます。

ハードウェア接続と INTECRIO-VP ハードウェアコンフィギュレータ

ハードウェア接続の作業は INTECRIO のハードウェアコンフィギュレータ (「HC」とも呼ばれます) で行います。

INTECRIO-VP のハードウェアコンフィギュレータを使用すると、仮想プロトタイピング用ハードウェアシステムの構成の定義とパラメータ設定を行うことができ、ここで設定されたモデルが「コントローラ環境」にロードされ、VP-PC ハードウェアシステムに組み込まれたインターフェースを制御します。

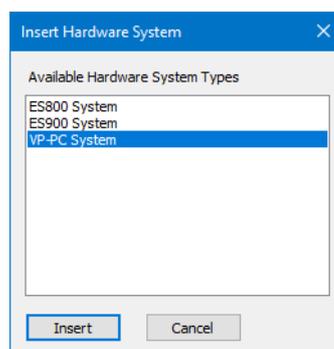
このようにして仮想プロトタイピングのためのロジックが生成されます。

5.5.2 ハードウェアシステムの作成とセットアップ

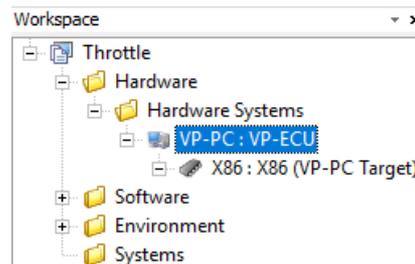
INTECRIO-VP では PC ターゲットを用いた仮想プロトタイピングを行います。ハードウェアシステムの作成方法は、ラピッドプロトタイピングの場合と同じです。

操作手順：仮想プロトタイピング用ハードウェアシステムを作成する

1. ワークスペースブラウザ上の Hardware フォルダ下の Hardware Systems フォルダを右クリックし、ショートカットメニューから **Add Hardware System** を選択します。
“Insert Hardware System” ダイアログボックスが開き、使用できるハードウェアシステムの一覧 (ES800 System、ES900 System、VP-PC System) が表示されます。



2. VP-PC を選択して **Insert** をクリックし、VP-PC システムを追加します。
“Rename” ダイアログボックスが開いてデフォルト名 VP-PC が表示されます。
3. 必要に応じて異なる名前を入力し、**OK** をクリックします。
ワークスペースブラウザに VP-PC システムが追加され、表示されます。
ここで使用できるターゲットは 1 つだけなので、直接
“Rename” ダイアログボックスが開きます。
4. 必要に応じて異なる名前を入力し、**OK** をクリックします。



5.6 レッスン 5：システムプロジェクト

レッスン 1～3 で、スロットル制御のソフトウェアモジュールをインポートして設定し、レッスン 4 でラピッドプロトタイピング実験に必要なハードウェアシステムを作成したので、今度は全体のシステムを作成します。

目標：

システムプロジェクトを作成します。まずシステムプロジェクトにハードウェアシステムとソフトウェアシステムを統合し、続いて、ソフトウェアシステム内やソフトウェアとハードウェア間のシグナル接続を確立します。

5.6.1 本レッスンで利用する重要な概念

システムプロジェクト (System Project)

「システムプロジェクト (System Project)」は、ハードウェアシステムとソフトウェアシステム、さらにシグナルマッピングやオペレーティングシステム設定を一体化したものです。実行コードの生成は、システムプロジェクトを対象として行われます。

ハードウェアシステム (Hardware System)

「ハードウェアシステム (Hardware System)」には、使用される ECU (実験ターゲット) やデバイス間のインターフェースに関する情報、といったハードウェア全体の構成や設定が含まれます。

ソフトウェアシステム (Software System)

「ソフトウェアシステム (Software System)」は、ハードウェアに依存しないアプリケーションソフトウェアに相当し、制御アルゴリズムを構成するモジュールやファンクション、コネクションなどが含まれます。

環境システム (Environment System)

「環境システム (Environment systems)」は、仮想プロトタイピング用のプラントモデルのモデリングに用いられるものです。ソフトウェアシステムと同様にモジュールとファンクションで構成されます。

統合

「統合」の処理には以下のステップが含まれます。

- a 制御アルゴリズムを作成するためのモデルとモデルパーツの組み立て
- b そのアルゴリズムを制御対象のハードウェアに接続（このレッスンでは行いません）
- c 実行ファイルの生成（次項のレッスンの一部）

プロジェクトコンフィギュレータ

システム全体を機能させるためには、INTECRIO にインポートしたモジュールを互いに接続する必要があります。またソフトウェアとハードウェアを接続するための入力と出力（シグナルシンクとシグナルソース）を作成し、ソフトウェアとハードウェアを接続します。

これらの作業は、INTECRIO のプロジェクトコンフィギュレータ機能を用いて行います。

5.6.2 システムプロジェクトの作成

始めにシステムプロジェクトを作成し、構成を定義します。

操作手順: システムプロジェクトを作成する

1. ワークスペースブラウザに表示されている Systems フォルダを右クリックし、ショートカットメニューから **Create System** を選択します。

“Create System” ダイアログボックスが開きます。

2. ダイアログボックスに Throttle_SP という名前を入力し、**OK** をクリックします。

システムプロジェクトがワークスペースに作成されます。

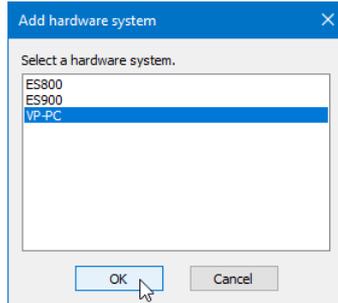


次に、5.5 項で作成したハードウェアシステムをシステムプロジェクトに統合します。

操作手順: ハードウェアシステムを統合する

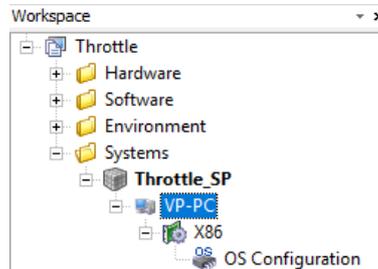
1. ワークスペースブラウザに表示されている Throttle_SP システムプロジェクトを右クリックし、ショートカットメニューから **Add Hardware System** を選択します。

“Add hardware system” ダイアログボックスが開きます。
Hardware/Hardware Systems フォルダ内に定義されているハードウェアシステムがリストアップされます。



2. VP-PC を選択して **OK** をクリックします。

ハードウェアシステムがシステムプロジェクトに統合されます。
ターゲットの下には、オペレーティングシステムコンフィギュレーションを表す OS Configuration というアイテムが自動生成されます。



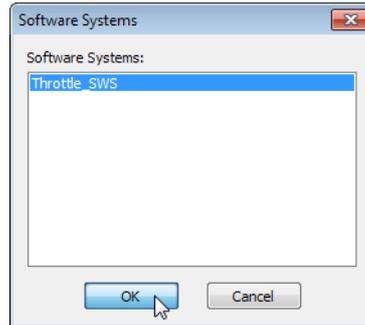
さらに、5.4.4 項で作成したソフトウェアシステムをシステムプロジェクトに統合します。

操作手順: ソフトウェアシステムを統合する

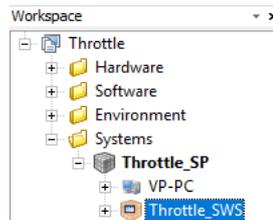
1. ワークスペースブラウザに表示されている Throttle_SP システムプロジェクトを右クリックし、ショートカットメニューから **Add Software System** を選択します。

“Software System” ダイアログボックスが開きます。

Software/Software Systems フォルダ内に定義されているソフトウェアシステムがリストアップされます。



2. Throttle_SWS を選択して **OK** をクリックします。
ソフトウェアシステムがシステムプロジェクトに統合されます。



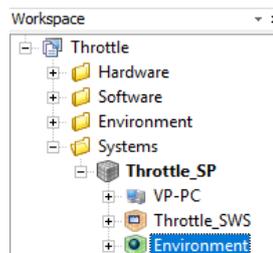
さらに、5.4.4 項で作成した環境システムをシステムプロジェクトに統合します。

操作手順: 環境システムを統合する

1. ワークスペースブラウザに表示されている Throttle_SP システムプロジェクトを右クリックし、ショートカットメニューから **Add Environment System** を選択します。

2. “Environment System” ダイアログボックスで Environment を選択し、**OK** をクリックします。

環境システムがシステムプロジェクトに統合されます。



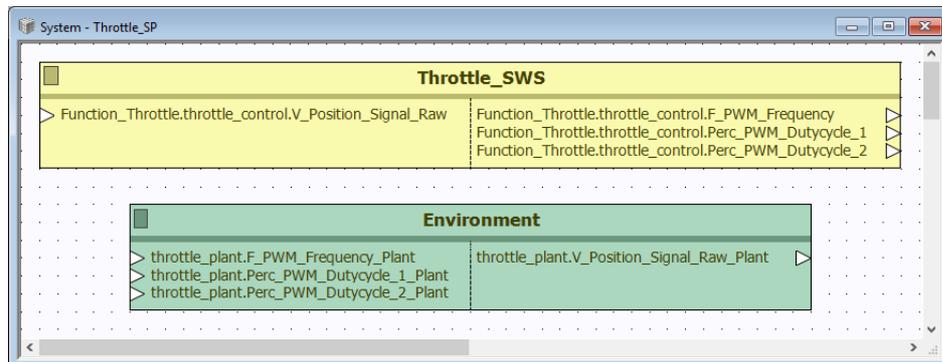
5.6.3 ソフトウェアシステムと環境システムの接続

ここではソフトウェアシステムと環境システムを接続して、完全なシグナルフローを定義します。

ソフトウェアシステムと環境システムは、118 ページのように接続します。接続には接続ウィザードを使用します。

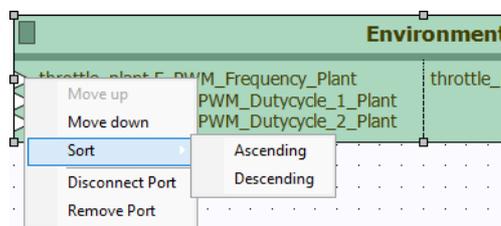
操作手順：ソフトウェアシステムと環境システムを接続する

1. Throttle_SP システムプロジェクトをダブルクリックします。
グラフィックエディタが開き、システムプロジェクトが表示されます。



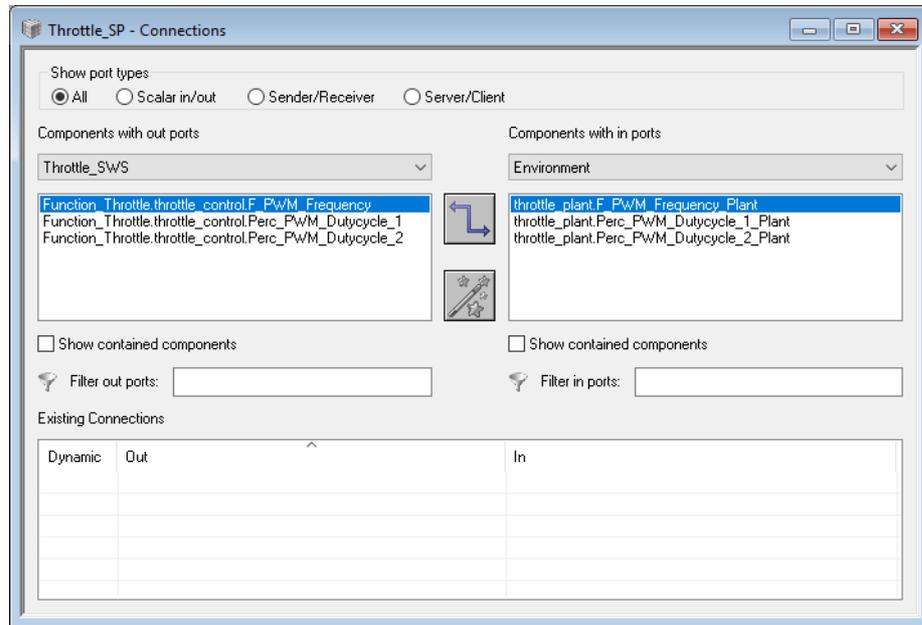
ソフトウェアシステムは黄色のブロックとして表示され、環境システムは緑のブロックとして表示されます。

2. 必要に応じて、ポートのショートカットメニューを使用してポートをソートします。



3. ワークスペースブラウザで、システムプロジェクトのショートカットメニューから **Open in Connection View** を選択します。
"Throttle_SP - Connections" ウィンドウ（接続ウィザード）が開きます。"Components with out ports" ドロップダウンリストにはシグナルソースを持つすべてのコンポーネントが表示され、"Components with in ports" ドロップダウンリストにはシ

グナルシンクを持つすべてのコンポーネントが表示されます。その下のフィールドには、選択されたコンポーネントのシグナルソース/シンクがリストアップされます。



4. ウィンドウ左側（出力側）のドロップダウンリストでソフトウェアシステムを選択し、その下のリストボックスから `Function_Throttle.throttle_control.F_PWM_Frequency` というアイテムを選択します。
5. ウィンドウ右側（入力側）のドロップダウンリストで環境システムを選択し、その下のリストボックスから `throttle_plant.F_PWM_Frequency_Plant` というアイテムを選択します。

接続する入力と出力が選択されると、ダイアログボックス中央にあるボタンが有効になります。



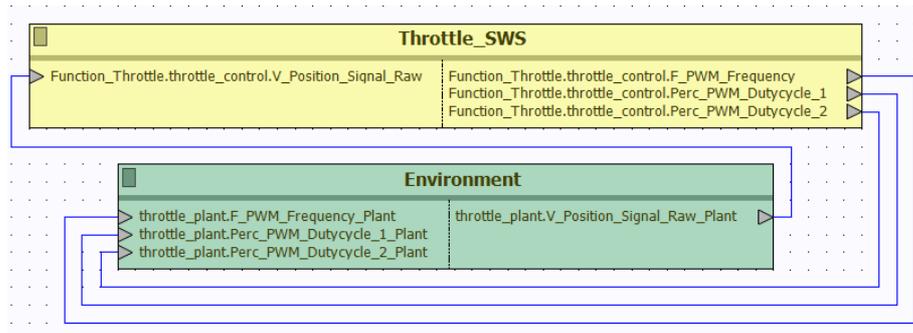
6. このボタンをクリックしてコネクションを作成します。作成された「コネクション」が“Existing Connections”という表に表示されます。“Dynamics”列で、その接続のタイプ（動的または静的）を選択できます。

Existing Connections		
Dynamic	Out	In
<input checked="" type="checkbox"/>	Function_Throttle.throttle_control.F_PWM_Frequency	throttle_plant.F_PWM_Frequency_Plant

7. さらに以下のコネクションを作成します。

Throttle_SWS	Environment
*.Perc_PWM_Dutycycle_1	*.Perc_PWM_Dutycycle_1_Plant
*.Perc_PWM_Dutycycle_2	*.Perc_PWM_Dutycycle_2_Plant
*.V_Position_Signal_Raw	*.V_Position_Signal_Raw_Plant

接続されたソフトウェアシステムと環境システムは、以下のようになります。



これで、システムプロジェクト作成に関係する作業はほぼ終わりです。次に行うのは、オペレーティングシステムの設定とプロトタイプの生成です。

5.7 レッスン 6：プロトタイプの生成

目標：

OS ウィザードを使用してオペレーティングシステムのコンフィギュレーションを設定します。そしてコード生成用オプションを設定し、最後にスロットル制御の実行プロトタイプを生成します。

5.7.1 本レッスンで利用する重要な概念

統合

「統合」の処理には以下のステップが含まれます。

- 制御アルゴリズムを作成するためのモデルとモデルパーツの組み立て
- そのアルゴリズムを制御対象のハードウェアに接続
- 実行ファイルの生成

前項までのレッスンは、上記のステップ a に相当するものです。

オペレーティングシステムのコンフィギュレーション設定

オペレーティングシステムはプロセッサをめぐって競合するタスクとプロセスの処理シーケンスを決定し、必要に応じてタスクの実行を切り替えます。その際に必要な設定は、すべてオペレーティングシステムのコンフィギュレーション設定時に行われます。

OS コンフィギュレータ (OSC)

オペレーティングシステムのコンフィギュレーション設定は、INTECRIO の「OSC (OS コンフィギュレータ)」というエディタで行います。OSC エディタは OS コンフィギュレータのユーザーインターフェースで、システムの概要を素早く把握でき、アプリケーションからの視点でコンフィギュレーションを編集することができます。

ビルド処理とプロジェクトインテグレータ

「プロジェクトインテグレータ (PI)」は、システム全体、つまりモジュールとファンクション、ハードウェア接続、OS コンフィギュレーションなどを1つの実行ファイル（「プロトタイプ」と呼ばれます）にまとめます。この際、実験用の ASAM-MCD-2MC ファイルも生成します。

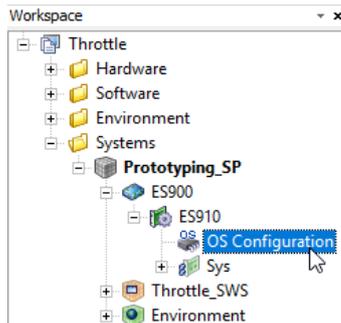
プロトタイプ (Prototype)

「プロトタイプ (Prototype)」は、実験ターゲットシステム用の実行ファイルを指します。プロトタイプにはソフトウェアファンクションの実行コードが書き込まれていますが、これは製品用 ECU を必要としない汎用的な形式になっています。

5.7.2 オペレーティングシステムのコンフィギュレーション設定

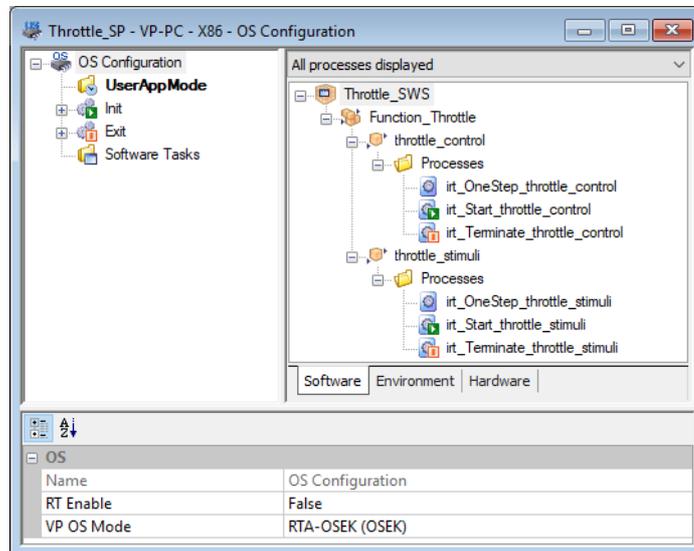
実行プロトタイプを生成するには、オペレーティングシステムのコンフィギュレーション設定を行い、各プロセスの処理シーケンスを定義する必要があります。

OS コンフィギュレーションは、OS Configuration というアイテムによって定義されます。このアイテムは、ハードウェアシステムを統合した時にシステムプロジェクト内のターゲットの下に自動的に作成されています（114 ページの「ハードウェアシステムを統合する」を参照してください）。



操作手順: OS コンフィギュレータを起動する

1. ワークスペースブラウザに表示されている OS Configuration をダブルクリックします。
OSC が開きます。



OSC エディタは 3 つの部分に分かれています。左上部は OS コンフィギュレーションビューです。右上部には 3 つのタブがあり、それぞれのタブに、ソフトウェアシステム、環境システム、ハードウェアシステムの階層構造が表示されます。ウィンドウ下部のフィールドには、OS コンフィギュレーションビューで選択されているアイテムの属性が表示されます。

OS コンフィギュレーションビューの最上位レベルにはオペレーティングシステム (OS) が表示され、その下にアプリケーションモード (UserAppMode) が表示されます。また、オペレーティングシステムの直下には、自動的に作成された「Init タスク」と「Exit タスク」も表示されます。これらのタスクには以下のような特徴があり、すべてのアプリケーションモードで使用されます。

- **Init** – アプリケーションモード開始時に一度だけ実行されます。このタスクの実行時は割り込みが無効になります。
- **Exit** – シミュレーション終了時に一度だけ実行されます。このタスクの実行時は割り込みが無効になります。

これらのタスクに加え、ユーザーはタイマタスクとソフトウェアタスクを作成することができます。タイマタスクはアプリケーションモードの下に作成され、ソフトウェアタスクは Software Tasks フォルダ内に作成されます。

操作手順: アプリケーションモードをセットアップする

アプリケーションモードについては名前の変更だけが可能です。他の設定は行えません。ここでは、チュートリアル用に設定されている名前をそのまま使用します。

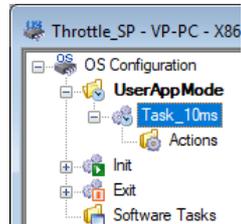
スロットル制御には Task_10ms というタイマタスクが必要です。

操作手順: タイマタスクを作成してセットアップする

1. アプリケーションモードを右クリックし、ショートカットメニューから **Add Timer Task** を選択します。
名前を指定できるダイアログボックスが開きます。

2. Task_10ms というタスク名を入力して **OK** をクリックします。

このタスクは OS コンフィギュレーションビュー内のアプリケーションモードの下に作成されます。さらに、このタスク用の Actions フォルダが自動的に作成されます。



ラピッドプロトタイピングターゲットの場合は、タイマタスク用に Events という 2 番目のフォルダも作成されます。

3. この新しいタスクを選択します。
入力フィールドに、タスク属性のデフォルト値が表示されます。

Timer Task	
Name	Task_10ms
Task ID	not yet assigned
Priority	0
Period	0.01
Delay	0
Execution Budget	0
Max. Number of Activation	1
Monitoring	False
Exclude from Tracing	True

4. 各属性に以下の値を設定します。

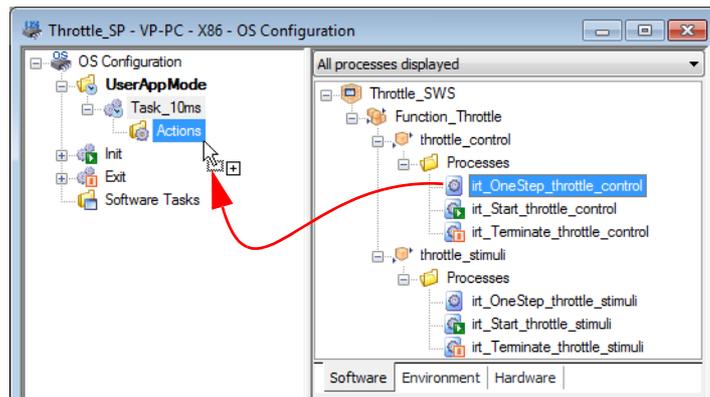
属性	説明	値
Priority	起動されたタスクの処理シーケンスを決定する、タスク優先順位	116
Period	タスクの起動間隔（単位は秒）	0.01
Delay	タスクが実際に起動されるまでの遅延時間（単位は秒）	0.0
Execution Budget	タスクに許容される最長の実行時間（単位は秒）。0 にすると、実行時間の監視が行われません。	0
Max. number of activation	タスクを同時に起動できる回数	1
Monitoring	タスク用のモニタリング変数を生成するかどうかを指定するフラグ	True
Exclude from Tracing	True の場合、システムがこのタスクを RTA-Trace によるトレース対象から除外します。	True

各属性についての詳細は、INTECRIO のオンラインヘルプまたはユーザーガイドの「OSC エディタ」の項を参照してください。

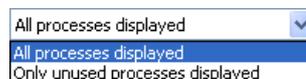
次に、プロセスをタスクに割り当てます。これには OSC 右上部のフィールドの “Software” タブと “Environment” タブを使用します。

操作手順: プロセスをタスクに割り当てる

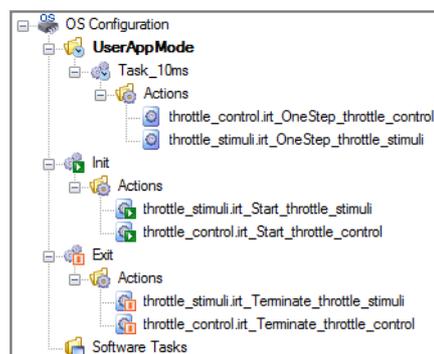
1. OS コンフィギュレーションビューを展開して、すべてのタスクが表示されるようにします。
2. “Software” タブのビューを展開して、すべてのモジュールのプロセスが表示されるようにします。



3. タブの上のドロップダウンリストで、全プロセスを表示する (All processes displayed) か、または未使用のプロセスだけを表示する (Only unused processes displayed) かを選択します。



4. irt_OneStep_throttle_control プロセスを、“Software” タブから OS コンフィギュレーションビューにドラッグし、Task_10ms タスクの Actions フォルダに格納します。
プロセスがこのタスクに割り当てられます。
5. 残りのプロセスを下図のように各タスクに割り当てます。



各プロセスは、プロセスがタスクに割り当てられた順序と同じ順序で実行されます。

6. 割り当て順が上図ようになっていない場合は、プロセスを別の位置にドラッグして、同じシーケンスになるようにしてください。

7. “Environment” タブに切り替えて、環境システムの各プロセスを適切なタスクに割り当てます。

これで、オペレーティングシステムのコンフィギュレーション設定が終わりました。

5.7.3 ビルド処理

ビルド処理は、いくつかのオプションを設定して処理を開始するだけで、ほぼ自動的に実行されます。ビルドオプションには 3 種類のレベルがあります。

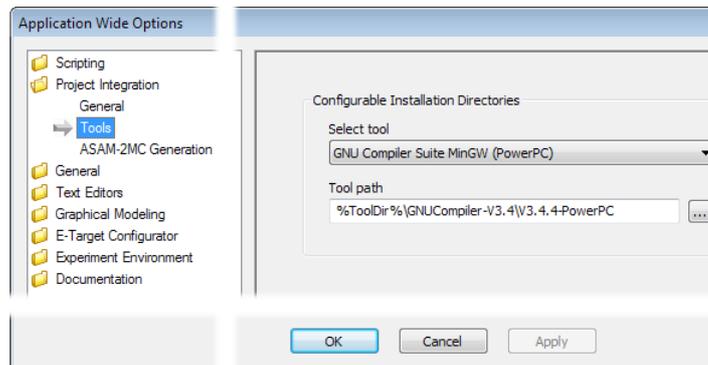
- アプリケーションレベル： **Tools > Options** で “Application Wide Options” ダイアログボックスを開いて設定します。
- ワークスペースレベル：ワークスペースのショートカットメニュー **Properties** で “Workspace Properties” ダイアログボックスを開いて設定します。
- プロジェクトレベル：システムプロジェクトのショートカットメニュー **Properties** で “System Properties” ダイアログボックスを開いて設定します。

このチュートリアルでは、コンパイラのパスとプロジェクトレベルのビルドオプションを確認します。

コンパイラのパスは、INTECRIO オプションとしてデフォルトパスが自動的に設定されていますが、必要に応じてそのパスを変更することができます。

操作手順：コンパイラのパスを確認／設定する

1. **Tools > Options** を選択します。
“Application Wide Options” ダイアログボックスが開きます。
2. Project Integration フォルダの Tools グループを開きます。



3. “Select tool” ドロップダウンリストからコンパイラを選択します。
4. このコンパイラのパスが “Tool Path” フィールドに正しく設定されていることを確認します。

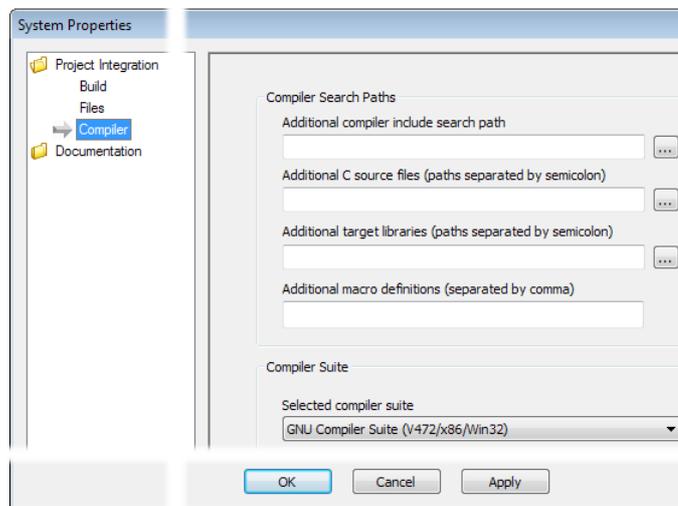
INTECRIO の製品パッケージには GNU Compiler Suite (PowerPC) が含まれているので、このコンパイラのパスが “Tool Path” フィールドに設定されています。
他のコンパイラを使用する場合は、別途インストールしてください。

5. **OK** でダイアログボックスを閉じます。

システムプロジェクトのオプション設定を変更することにより、デフォルト設定されている GNU Compiler の代わりに、PC にインストールされている他のコンパイラ（例：Microsoft Visual Studio 2008 コンパイラ）を使用することができます。GNU Compiler を使用する場合は、設定を変更する必要はありません。

操作手順：コンパイラを選択する

1. ワークスペースブラウザに表示されている Throttle_SP システムプロジェクトを右クリックして、ショートカットメニューから **Properties** を選択します。
“System Properties” ダイアログボックスが開きます。
2. Project Integration フォルダの Compiler グループを開きます。

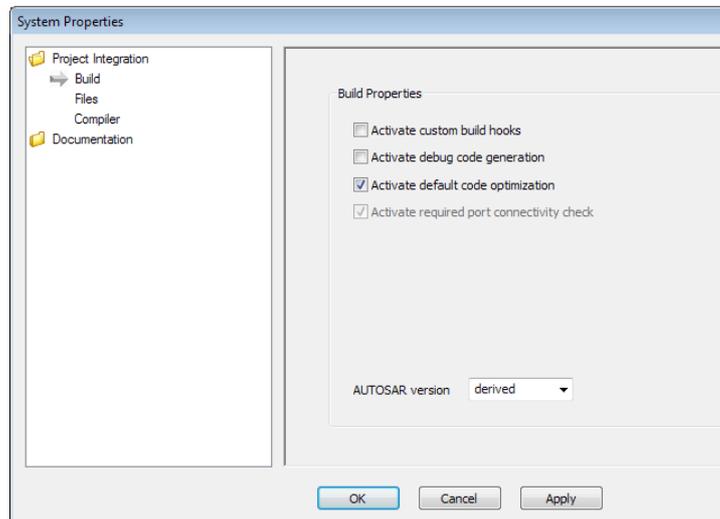


3. “Selected compiler suite” ドロップダウンリストからコンパイラを選択します。
4. 必要に応じて“Compiler Search paths” フィールドに検索先のパスを入力します。
5. **OK** でダイアログボックスを閉じます。

操作手順: ビルドオプションを設定する

1. ワークスペースブラウザに表示されている Throttle_SP システムプロジェクトを右クリックし、ショートカットメニューから **Properties** を選択します。

“System Properties” ダイアログボックスが開きます。



2. 左側のフィールドに表示されている Project Integration フォルダの Files グループを開きます。このグループのオプションが右側のフィールドに表示されます。
3. “Result File Basename” フィールドに、プロトタイプの実行ファイルと ASAM-MCD-2MC ファイルに使用されるベースネームとして、Throttle と入力します。他のグループのオプションは、デフォルト設定のままにしておいてください。
4. **OK** で設定を有効にします。

操作手順: 実行ファイルを生成する

1. 以下のいずれかを行います。
 - ワークスペースブラウザから Throttle_SP システムプロジェクトを右クリックし、ショートカットメニューから **Build** を選択します。
前回実行されたビルド処理の結果が現在も有効な場合、ビルド処理は実行されません。
 - ショートカットメニューから **Rebuild** を選択します。
前回生成されたファイルは削除され、ビルド処理が実行されてすべての出力ファイルが再生成されます。

現在の処理についての情報が “Log Window” フィールドの “Messages” タブに表示されます。また、ウィンドウ最下部のステータスバーに進捗バーが表示され、処理の進行状況を示します。

これで、実行可能なプロトタイプが生成されました。次項では、このプロトタイプを用いて実験を行います。

5.8 レッスン7：実験

目標：

実験を行い、スロットル制御モデルをテストします。ETAS Experiment Environment を使用して、さまざまな測定／適合インストゥルメントを用いた実験を行います。

5.8.1 本レッスンで利用する重要な概念

実験

ETAS Experiment Environment を使用して、プロトタイプの機能の検証実験を行います。実験を行う際は、さまざまなタイプの測定／適合インストゥルメントを使用できます。

実験はハードディスクに保存されますが、ワークスペースブラウザから開くことができます。適切な実験をロードすることにより、特定の作業用の実験環境を素早くセットアップすることができます。

仮想プロトタイピング (Virtual Prototyping - バーチャルプロトタイピング)

車両の電子制御ファンクションの仮想プロトタイプを作成し、PC 上で実行してテストを行います。仮想プロトタイプは以下のコンポーネントで構成されます。

- 自動車用組み込みソフトウェア（アプリケーションソフトウェアと OSEK オペレーティングシステム）
- プラントモデル（環境システム）

仮想プロトタイピングの手法を用いることにより、開発工程の初期段階において、システムモデルを用いた Model-in-the-Loop (MiL) 環境を構築し、ファンクションを検証することができます。これによって、早い段階でのファンクションとシステムの並行開発が実現でき、開発工程全体の効率が向上します。

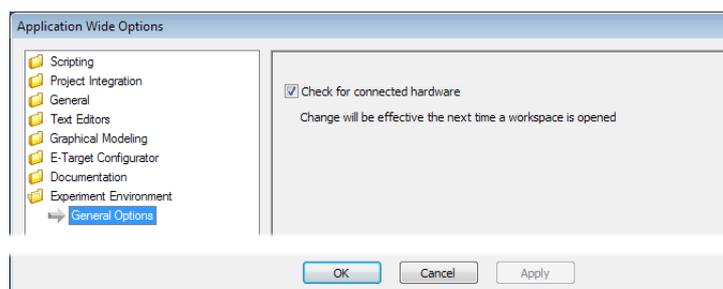
5.8.2 準備

まず実験の準備を行います。ETAS Experiment Environment を開き、いくつかのオプションを指定します。

INTECRIO の場合、実験は独立したアイテムとしては扱われないため、ワークスペースブラウザには表示されません。

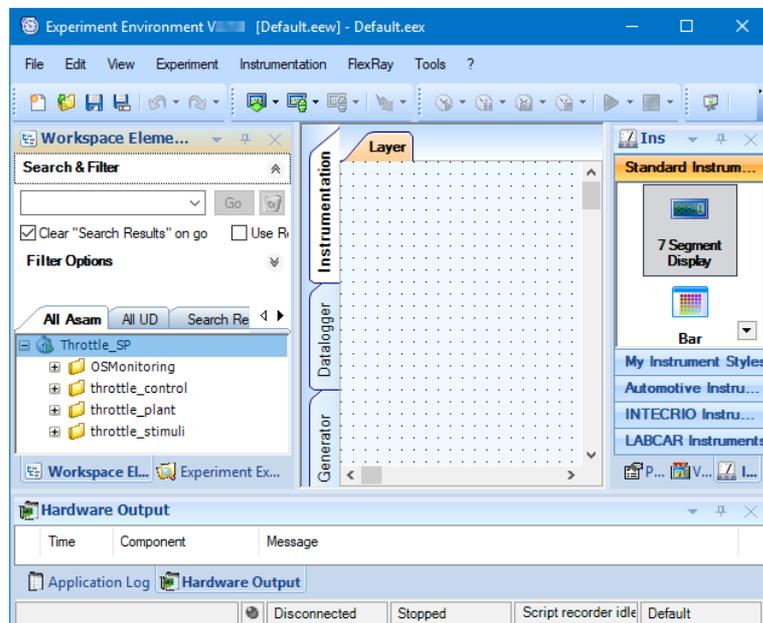
操作手順：ETAS Experiment Environment を開く

1. アプリケーションレベルオプションの **Check for connected hardware** がオンになっていることを確認します。



このオプションがオンになっていないと、ETAS Experiment Environment は開けません。

2. ワークスペースブラウザに表示されている Throttle_SP システムプロジェクトを右クリックします。
3. ショートカットメニューから **Open Experiment** を選択します。
ETAS Experiment Environment が専用のウィンドウで開き、実験がロードされます。



この時点では、実験のセットアップはまだ行われていません。そのため、エレメントリスト（デフォルトでは ETAS Experiment Environment ウィンドウの左側にドッキングされている“Workspace Elements”ウィンドウの“All ASAM”タブ）にはシステムプロジェクトの測定変数と適合変数しか表示されていません。各変数は、エレメントリスト内のシステムプロジェクトの以下のサブフォルダに、タイプごとに分類されて表示されます。

- throttle_control および throttle_stimuli – 各モジュールの測定変数と適合変数
- throttle_plant – 環境モジュールの測定変数と適合変数
- OSMonitoring – モニタリング変数

このレッスンでは、スロットル制御のプロトタイプ実験に必要な環境をセットアップします。

5.8.3 実験のセットアップ

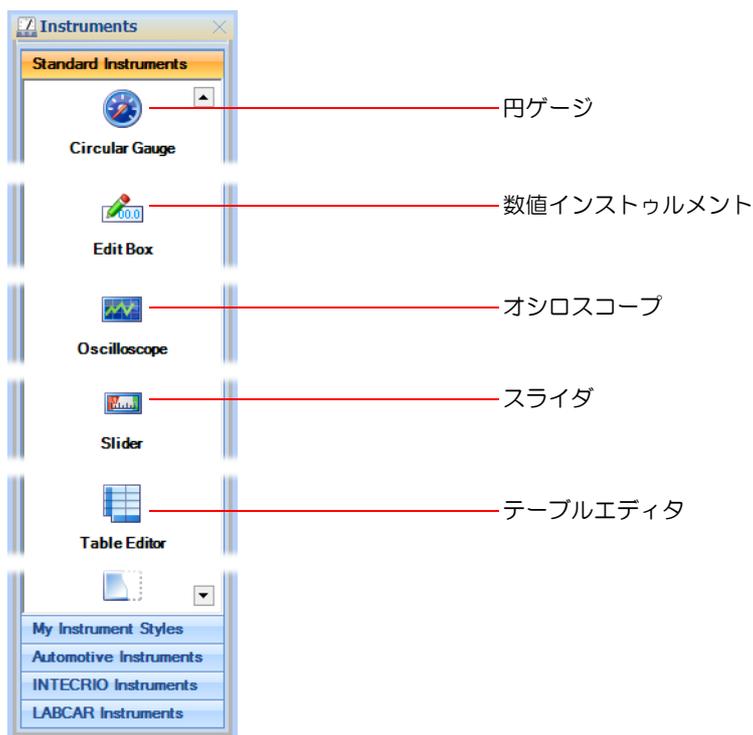
スロットル制御用に測定／適合インストゥルメントを作成し、それらに割り当てる変数を選択することにより、「実験のセットアップ」を行います。ここでは、以下の変数の測定や適合を行います。

変数	説明
Perc_Desired_Position	スロットルバルブ位置の要求値
Perc_Actual_Position	スロットルバルブ位置の実測値
K_S_Value, Td_S_Value, Ti_S_Value, Tv_S_Value	PID コントローラの制御パラメータ
Abs_Error_Counter	エラーカウント
error_tolerance_yData	エラー検知に用いられる許容値

変数	説明
Stimulus_Select_Value	スロットル位置の要求値を取得するためのステミュラスを選択する値
dT	タスク起動間の時間差
grossRunTime	タスクの総実行時間

5.8.3.1 測定／適合インストゥルメントの作成

実験ウィンドウ右側にドッキングされている“Instrumentation”ウィンドウには、実験で使用できるすべてのインストゥルメントが表示されています。このレッスンではそのうちの一部のみを使用します。



数値インストゥルメントとスライダは、割り当てられた変数のタイプ（測定変数または適合変数）に応じて、測定インストゥルメントまたは適合インストゥルメントとして使用されます。テーブルエディタは適合専用、円ゲージとオシロスコープは測定専用のインストゥルメントです。

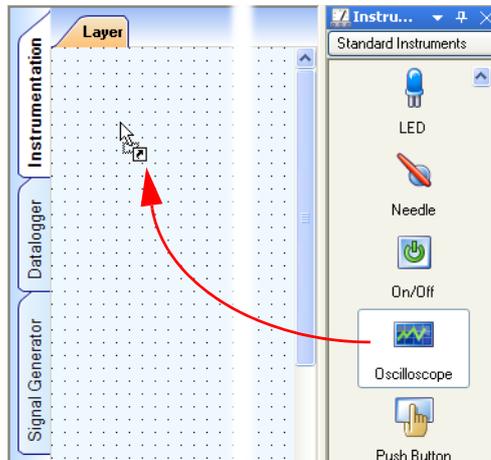
ここで行うスロットル制御の実験では、オシロスコープ、スライダ、円ゲージ、テーブルエディタを1つずつ使用し、さらに数値インストゥルメントを7つ使用します。オシロスコープ、円ゲージ、テーブルエディタ、スライダはそれぞれ専用のウィンドウに表示し、数値インストゥルメントは2つのウィンドウに配置します。

注記

2つ以上の変数を表示できるインストゥルメントは、オシロスコープだけです。他のインストゥルメントには変数を1つしか表示できません。そのため、同じタイプのインストゥルメントを複数作成しなければならない場合があります。

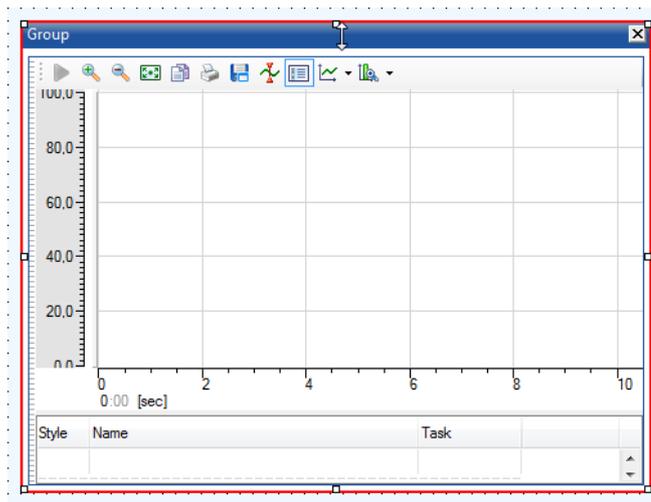
操作手順：オシロスコープを作成する

1. “Instrumentation” ウィンドウの **Oscilloscope** アイコンを、“Instrumentation” フィールドにドラッグします。



新しいオシロスコープが開きます。

2. オシロスコープのサイズを調整します。



円ゲージ、テーブルエディタ、スライダも同様に作成します。

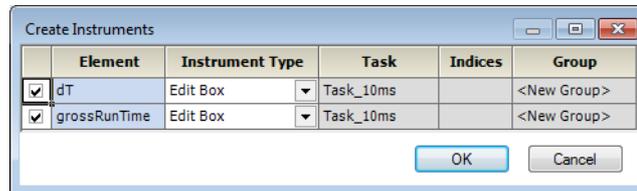
数値インストゥルメントは、測定インストゥルメントと適合インストゥルメントという2つのグループに分けて作成します。ここでは、“Instrumentation” ウィンドウではなく “Workspace Elements” ウィンドウから行う方法を紹介します。

操作手順：数値インストゥルメントを追加する

1. “Workspace Elements” ウィンドウで、`grossRunTime` と `dT` という OS モニタリング変数を選択します。

2. これらの変数のショートカットメニューから **Measure/Calibrate** を選択します。

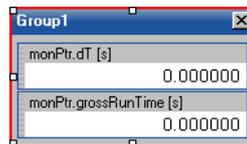
“Create Instruments” ダイアログボックスが開きます。



数値インストゥルメントを備えた新しい測定ウィンドウを作成するには、両方の変数についてインストゥルメントタイプ (“Instrument Type”) に Edit Box、グループ (“Group”) に <New Group> を選択します。

3. **OK** で選択内容を確定します。

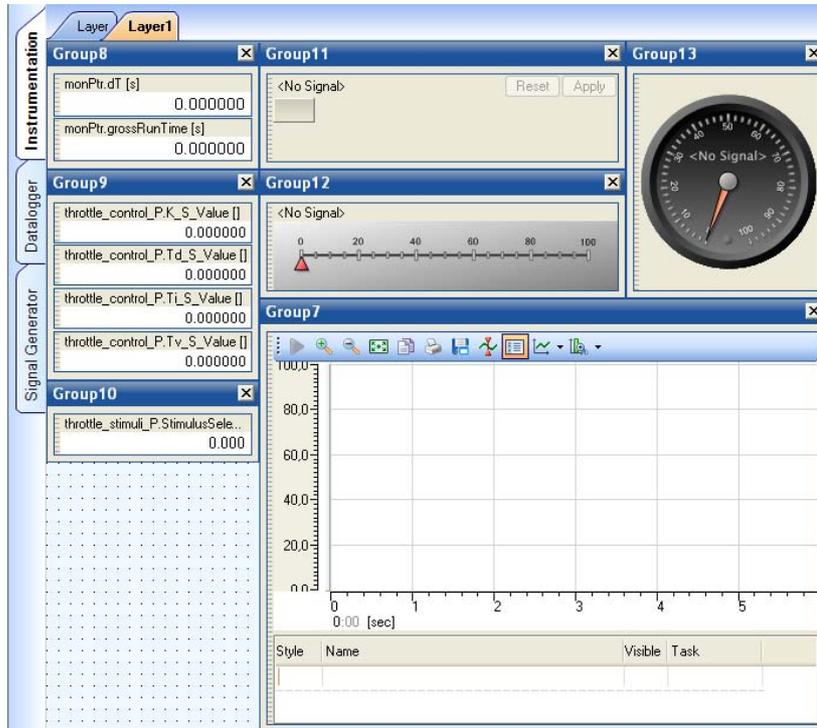
2つの数値インストゥルメントを含んだ測定ウィンドウが開きます。各インストゥルメントには2つの変数がそれぞれ割り当てられています。



4. 次に、5つのパラメータ (K_S_Value、Td_S_Value、Ti_S_Value、Tv_S_Value、Stimulus_Select_Value) 用の適合ウィンドウを作成します。

ここでは、ショートカットメニューの **Measure/Calibrate** を使わずに、変数を直接 “Instrumentation” フィールドの空いている場所にドラッグしてください。

すべてのインストゥルメントを追加したら、下図の例のように“Instrumentation”フィールド内に配置してください。



5.8.3.2 測定/適合変数の割当て

インストゥルメントを作成したので、次に、測定や適合を行う変数を各インストゥルメントに割り当てます。エレメントリストでは、測定変数は赤い正方形のシンボル (■) で示され、適合変数は青い円形のシンボル (●) で示されます。

操作手順：測定変数と適合変数を割り当てる

1. 測定変数 Perc_Desired_position を“Workspace Elements”タブからオシロスコープにドラッグします。
測定変数がオシロスコープに割り当てられます。
2. 下の表に従い、変数を残りのインストゥルメントに割り当てます。

変数	インストゥルメント
Perc_Actual_Position	オシロスコープ
Abs_Error_Counter	円ゲージ/スライダ
error_tolerance_yData	テーブルエディタ

5.8.3.3 測定/適合インストゥルメントのセットアップ

インストゥルメントのセットアップは、シミュレーションの開始前でも実行中でも任意に行えます。設定オプションについての詳細はオンラインヘルプを参照してください。

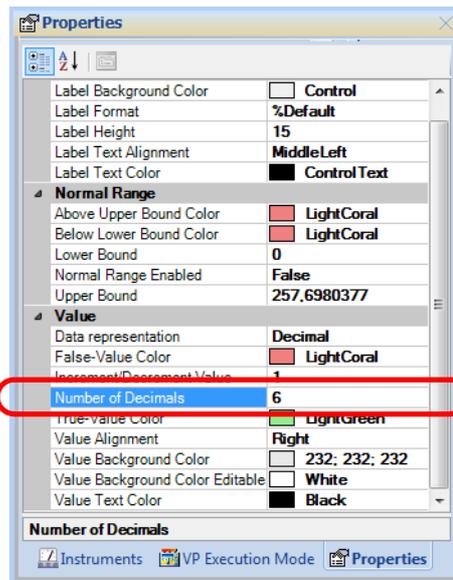
dT と grossRunTime の場合は、数値インストゥルメントの表示を調整して小数点以下の桁数を 6 桁以上にすることをお勧めします。

操作手順：数値インストゥルメントをセットアップする

1. 測定変数が割り当てられたインストゥルメントをクリックして選択します。

ここでは、インストゥルメントを含むウィンドウを選択するのではなく、ウィンドウ内のインストゥルメントを個別に選択してください。

“Properties” ウィンドウに、インストゥルメントのプロパティが表示されます。



2. “Number of Decimals” フィールドに、小数点以下の桁数を入力します。

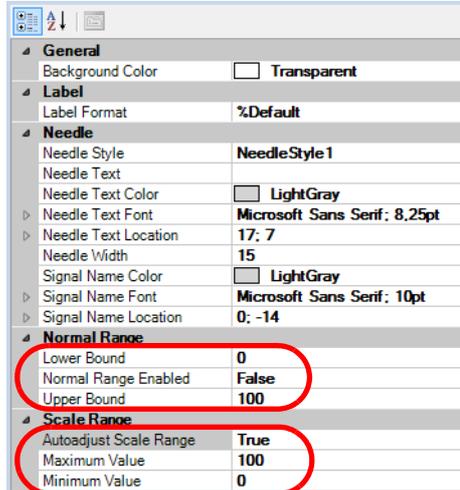
変更内容は直ちに有効になります。

続いて別のインストゥルメントをセットアップするには、セットアップするインストゥルメントをクリックし、そのインストゥルメントのプロパティを“Properties” ウィンドウに表示します。

円ゲージについては、測定変数 `Abs_Error_Counter` が適切に表示されるように、いくつかのプロパティを変更します。

操作手順: 円ゲージをセットアップする

1. “Properties” ウィンドウに円ゲージのプロパティを開きます。

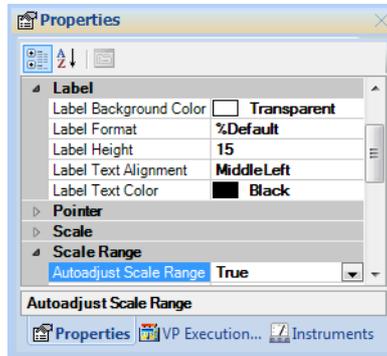


2. 値が正常範囲を外れた時に通知されるようにするには “Normal Range Enabled” フィールドの値を True にします。
3. “Upper/Lower Bound” フィールドに、正常範囲の上下限值を入力します。
4. “Autoadjust Scale Range” フィールドで、自動スケール調整を行う (True) か行わない (False) かを選択します。
True を選択すると、表示範囲の上限値が、実際の測定値に応じて大きくなります。ただし、測定値が表示範囲よりも恒常的に小さい場合、上限値は自動的に小さくならないので、必要に応じて手動操作で調整する必要があります。
5. “Maximum/Minimum Value” フィールドに表示範囲の上下限值を入力します。

スライダは、測定変数 Abs_Error_Counter が適切な形で表示されるようにいくつかのプリセット値を変更します。

操作手順: スライダをセットアップする

1. “Properties” ダイアログボックスにスライダのプロパティを開きます。



“Autoadjust Scale Range” フィールドと “Maximum/Minimum Value” フィールドの意味は、円ゲージのものと同じです。

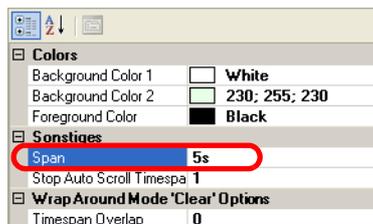
テーブルエディタについては、デフォルト設定のままにします。

オシロスコープについては、チュートリアルでは、座標軸についての最も重要な設定のみ扱います。グリッドや凡例、チャンネルについては、デフォルト設定を使用します。

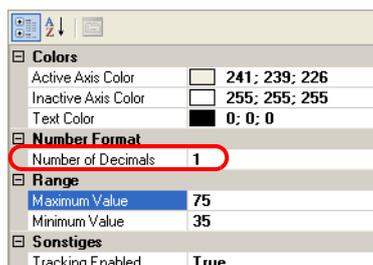
操作手順: オシロスコープをセットアップする

1. オシロスコープ内で X 軸をクリックします。

“Properties” ウィンドウに X 軸のプロパティが表示されます。



2. “Span” フィールドに 5s を入力し、時間幅の表示範囲を 5 秒に指定します。
3. オシロスコープ内で Y 軸をクリックし、“Properties” ウィンドウに Y 軸のプロパティを表示します。



4. “Number of Decimals” フィールドに 2 を入力し、小数部分の最大表示桁数を 2 桁に設定します。

Y 軸の表示領域は、ここではまだ変更しません。実際に測定を行いながら、測定値の範囲に応じて調整します。

5.8.3.4 測定/適合インストゥルメントの保存

設定がすべて完了したら、設定内容全体、つまり「実験」を保存し、後で同じ GUI 環境を再現できるようにします。

操作手順: 実験を保存する

1. **File > Save Experiment** を選択します。

測定/適合ウィンドウについての設定が Default.eex ファイルに保存されます。

このファイルはワークスペースの実験ディレクトリに格納されます。

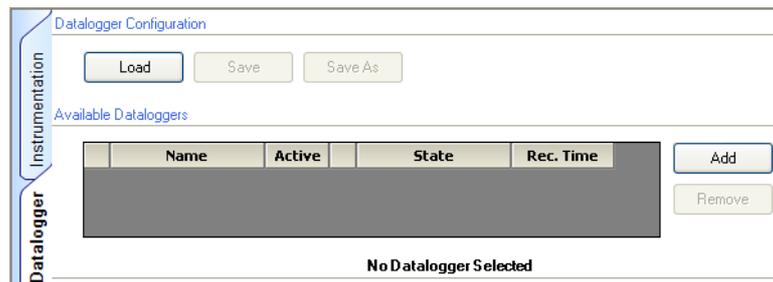
例:

```
ETASData¥INTECRIO5.0¥Tutorial¥Throttle¥EE¥
Experiments¥System¥Default¥Default.eex
```

2. 実験の名前や保存先を指定するには、**File > Save Experiment As** を選択します。

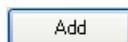
5.8.4 データ記録に関する設定

データの記録には ETAS Experiment Environment のデータロガー (“Datalogger” フィールド) を使用します。



データの記録を行うには、まず最初にデータロガーを作成します。

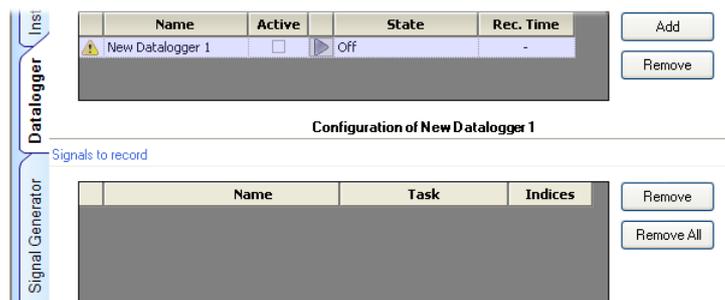
操作手順: データロガーを作成する



1. “Available Dataloggers” フィールドの **Add** ボタンをクリックします。

新しいデータロガーが作成されます。

2. データロガーの名前を変更するには、“Name” 列をクリックします。



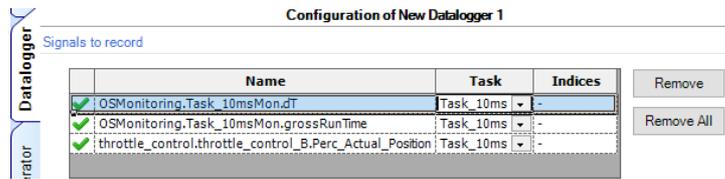
次に、データ記録に関する以下のプロパティを定義します。

- 記録する測定変数
- 記録データを保存するログファイル
- 記録開始/終了のタイミング

操作手順：記録する測定変数を選択する

1. “Workspace Elements” ウィンドウから 1 つまたは複数の測定変数を選択します。
2. 選択した測定変数を右クリックし、ショートカットメニューから **Add to Datalogger** を選択します。

測定変数がデータロガーの “Signals to record” フィールドに追加されます。



システムプロジェクトに含まれるタスクがデータ測定タスクとして設定されます。

操作手順：ログファイルを指定する

ログファイルについての設定は、データロガーの “Recording File Settings” フィールドで行います。



1. “Recording base filename” フィールドで、テキストフィールドの右側にあるボタンをクリックします。

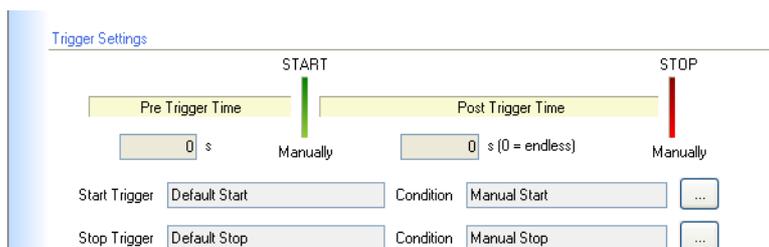
ファイル選択ダイアログボックスが開きます。

2. ログファイルのパス

(D:\¥ETASData¥INTECRIO5.0¥Tutorial など) とファイル名を入力して **Save** をクリックします。

Auto Increment オプションをオンにすると、ファイル名にシーケンス番号が付加され、保存のたびに番号がインクリメントされます。

データ記録の開始/終了のタイミングを決定する条件（「開始フィルタ」/「終了フィルタ」と呼ばれます）は、“Trigger Settings” フィールドで詳細に設定することができます。シミュレーション開始後、手動操作またはトリガによって記録を開始し、設定時間だけ記録を行ったり、トリガによって記録を終了したりすることもできます。

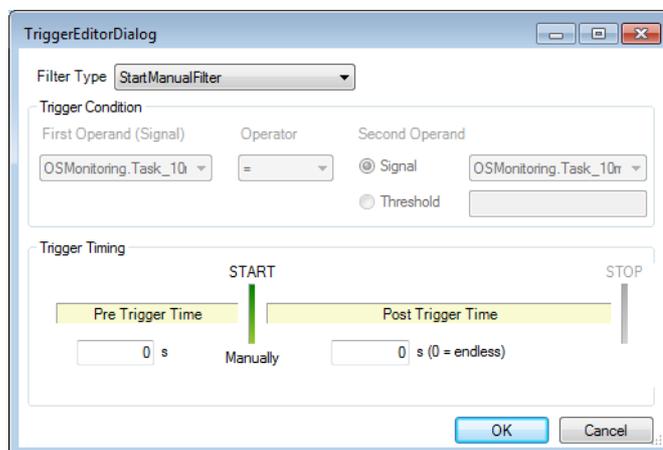


チュートリアルでは、手動操作で記録を開始し、設定時間だけ記録を行います。トリガを使用するデータ記録の方法は、ETAS Experiment Environment のオンラインヘルプを参照してください。

操作手順：記録時間を指定する



1. データロガーの“Trigger Settings”フィールドで、“Start Trigger”行の右端のボタンをクリックします。
“TriggerEditorDialog”ダイアログボックスが開きます。



2. “Filter Type”フィールドで、StartManualFilter を選択します。
3. “Pre Trigger Time” および “Post Trigger Time” フィールドに、値 0 および 10 を入力します。
この設定により、開始操作を行った後、10 秒間だけデータが記録されます。

データ記録は、データロガーが起動している間のみ行われます。上記のように設定すると、記録開始後、選択された測定変数のデータが 10 秒間記録され、指定されたファイル（141 ページの「記録」を参照してください）に保存されます。このデータは分析ツール（INTECRIO 付属の MDA など）で分析することができます。

5.8.5 測定／適合タスクの実行

本項では、これまでにセットアップした実験を使用して実験を実行します。

5.8.5.1 シミュレーションの開始

操作手順：シミュレーションと測定を開始する



1. Experiment Environment の **Download** ボタンをクリックします。



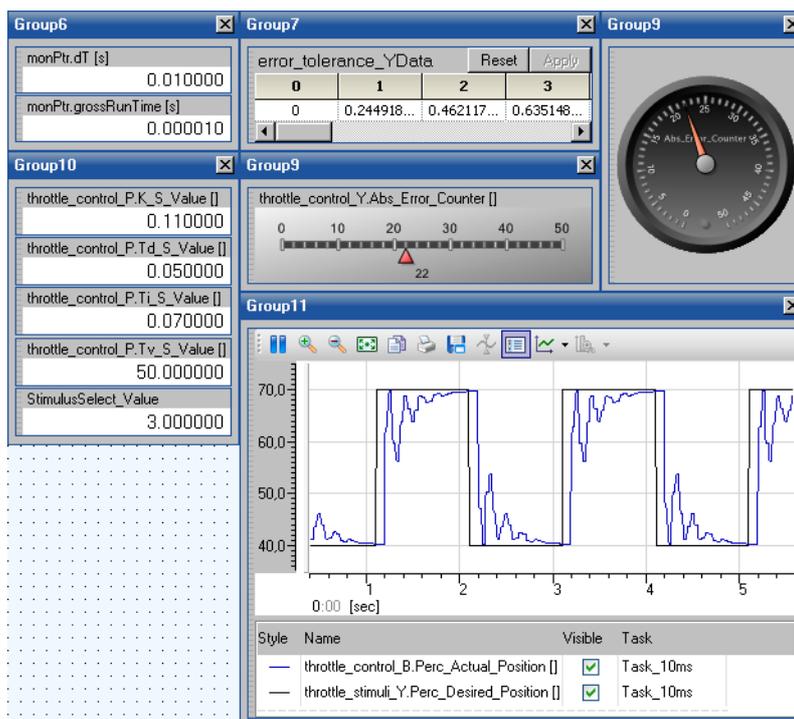
2. **Start Simulation** ボタンをクリックしてシミュレーションを開始します。



3. **Start Measurement** ボタンをクリックして測定を開始します。データロガーによるデータ収集も開始されます。データ収集は、設定した時間が経過すると終了します。

また測定値は、測定変数が割り当てられている測定インストゥルメントに表示されます。

ETAS Experiment Environment には下図のように測定値が表示されます。



5.8.5.2 測定

数値表示、円ゲージ、棒グラフ表示

これらの測定インストゥルメントには、シミュレーションと測定が開始されるとすぐに最新の測定値が表示されます。各インストゥルメントの表示設定は、シミュレーション実行中に変更できます（「測定/適合インストゥルメントのセットアップ」を参照してください）。これらのインストゥルメント内では測定データの分析は行えません。

オシロスコープ

オシロスコープもシミュレーションと測定が開始されると最新の測定値を表示します。オシロコプの場合、表示設定や値の分析を行うには、表示を休止（一時停止）する必要があります。

オシロスコープにおいて、実際の測定値が Y 軸の表示範囲から外れると、測定値のカーブの全体像を見ることができなくなります。そのような場合、以下のような方法で表示領域を変更できます。

操作手順：表示領域を変更する



1. 測定を一時停止します。

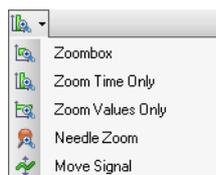


2. **Zoom In / Zoom out** ボタンをクリックして X 軸のスケールを変更します。

3. Y 軸の表示範囲を上下にシフトするには、Y 軸上でマウスの左ボタンを押下し、そのままポインタを上下に移動します。

4. Y 軸のスケールを変更するには、**<Ctrl>** キーを押したまま Y 軸上でマウスの左ボタンを押下し、そのまま上下にポインタを移動します。

5. 現在表示されている値に合わせて Y 軸の表示範囲を自動調整するには、Y 軸上でマウスを右クリックしてショートカットメニューを開き、**Zoom to Fit** を選択します。



6. その他のズーム機能を使用するには、オシロスコープのツールバーの **Zoom** ボタンをクリックします。

7. “Properties” ウィンドウで X 軸または Y 軸のプロパティを表示し、設定を任意に編集することもできます。



8. **Resume All (Tracking)** ボタンをクリックして表示を続行します。

すべての測定変数を 1 つの座標系に表示すると、表示が煩雑になって視認性が悪くなってしまう場合があります。そのような場合は、変数ごとに個別の座標軸を使用することができます。

操作手順：表示形式を設定する

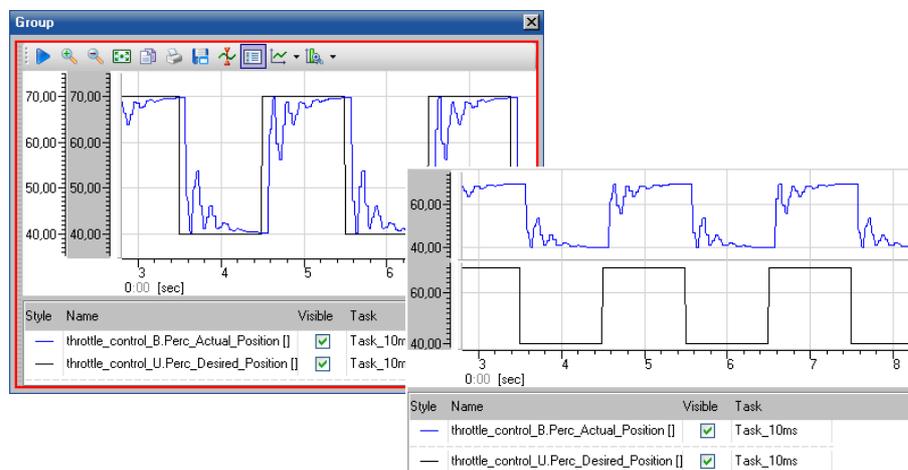
1. オシロスコープのシグナル表示領域を右クリックしてショートカットメニューを開き、**Single Axis per channel** を選択します。

2 つの測定変数用にそれぞれ個別の Y 軸が割り当てられ、各軸の表示範囲を個別に設定できるようになります。

2. オシロスコープのシグナル表示領域を右クリックしてショートカットメニューを開き、**Single Strip per channel** を選択します。

オシロスコープが上下に 2 つの領域に分割され、それぞれの領域に各測定変数が表示されます。Y 軸も個別に割り当てられます。

下図にその例を示します。



操作手順：測定データを分析する

1. オシロスコープの **Pause All** ボタンをクリックして測定データ表示を一時停止します。

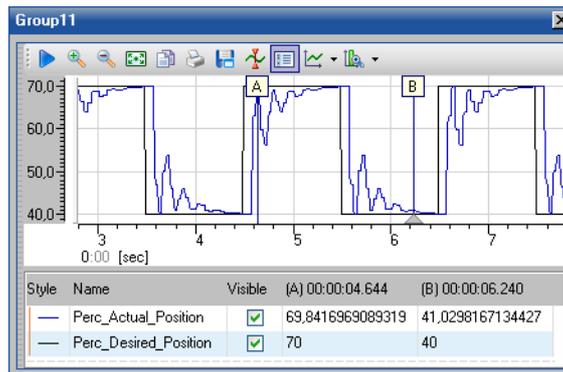


2. **Add Cursor** ボタンをクリックします。
表示領域にラインカーソルが表示されます。

3. カーソルをもう 1 本追加します。

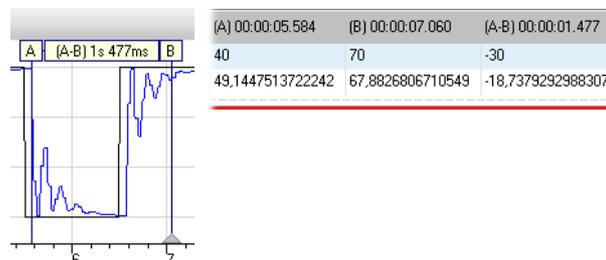
4. マウスでカーソルをドラッグし、値を読み取りたい位置まで移動します。

各カーソルの名前が表示され、チャンネルごとに、カーソル位置の Y 値が下部のチャンネルリスト内に表示されます。



5. 2 番目のカーソルを右クリックし、ショートカットメニューから **Show Cursor difference > A: <X value>** を選択します。

2 本のカーソルの時間の差異がカーソル付近に表示され、Y 値の差異がチャンネルリスト内に表示されます。

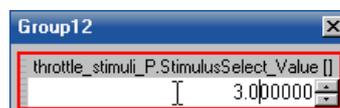
**5.8.5.3 適合**

適合変数（特性値、特性カーブ、特性マップ）をインストゥルメントに割り当てると、そのインストゥルメントはエディタとして機能します。さまざまなタイプのエディタで、適合変数の値を直接変更することができます。

操作手順：各種エディタで値を適合する

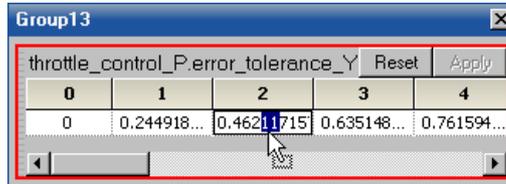
1 数値エディタで特性値を適合する場合

1. エディットボックスのテキスト領域をクリックします。
2. スクロールボタンを使用するか、または値を入力します。



新しい値が直ちに有効になります。

3. PID コントローラの制御パラメータを適合し、変数 Perc_Actual_Position の変化をオシロスコープで観察してください。
2. テーブルエディタで特性カーブ（例：error_tolerance_YData）を適合する場合
 1. テーブルのセルをダブルクリックします。
 2. 新しい値を入力します。



3. **Apply** をクリックします。
新しい値が有効になります。
4. 許容値を適合しながら、変数 Abs_Error_Counter の変化を円ゲージとスライダで観察してください。

5.8.5.4 記録

5.8.4 項においてデータロガーの設定を行ったので、測定を実行し、さらにデータ記録を開始します。

操作手順：測定データを記録する

1. データロガーを表示します。
“Available Dataloggers” フィールドの “State” 列に、データロガーの現在のステータスが表示されます。
2. “Active” 列をクリックしてデータロガーを起動します。

Available Dataloggers

Name	Active	State	Rec. Time
✓ New Datalogger 1	<input type="checkbox"/>	Off	-

チェックマークが表示されてデータロガーが起動し、さらに、右側の列の記録開始ボタンが有効になります。



3. 記録開始ボタンをクリックします。
このボタンの押下により記録開始のマニュアルトリガが発生して記録が開始され、10 秒後に自動的に停止します。

Available Dataloggers

Name	Active	State	Rec. Time
✓ New Datalogger 1	<input checked="" type="checkbox"/>	Recording	3,5

記録された測定データは、指定されたファイル（例：
D:\¥ETASData¥INTECRIO5.0¥Throttle¥MyLogfile0001.dat など）に保存されます。このデータは、分析ツールなどを用いたオフライン分析に使用することができます。

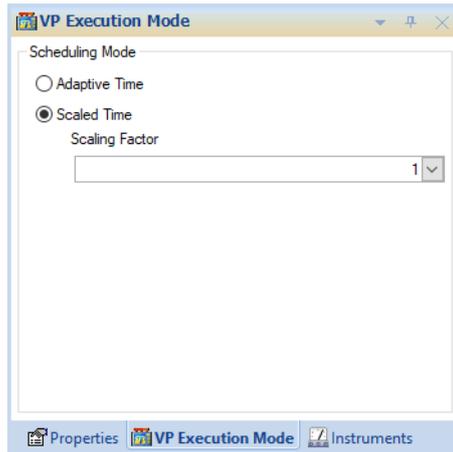
5.8.5.5 仮想プロトタイピング実験の特殊な機能と条件

ここまでの内容は仮想プロトタイピングとラピッドプロトタイピングの両方に共通の内容でしたが、ここでは仮想プロトタイピング固有の特殊な機能や条件について説明します。

VP 実験の場合は、実験環境内に “VP Execution Mode” ウィンドウ（ETAS Experiment Environment ウィンドウの右側にドッキングされています）が表示され、そこでシミュレーション時間の制御モードを選択することができます。

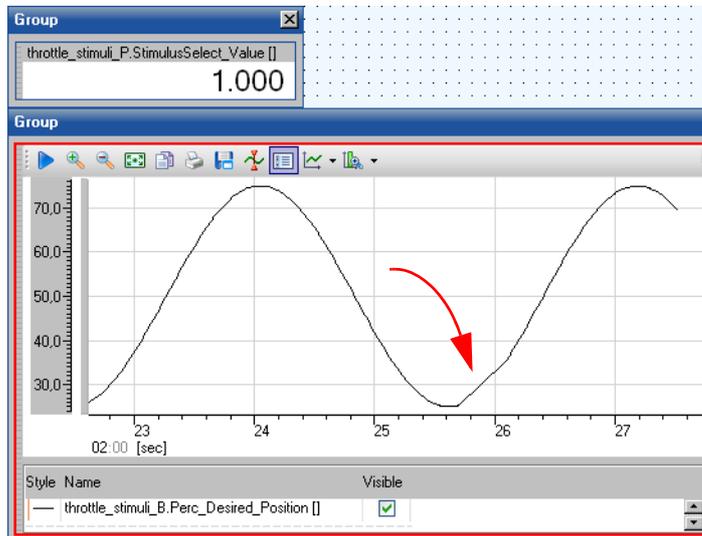
Adaptive Time モデルを可能な限り高速で、つまり最短の処理時間で実行します。

Scaled Time (デフォルト) モデルをリアルタイム (Scaling Factor = 1)、早送り (Scaling Factor < 1)、スローモーション (Scaling Factor > 1) のいずれかで実行します。



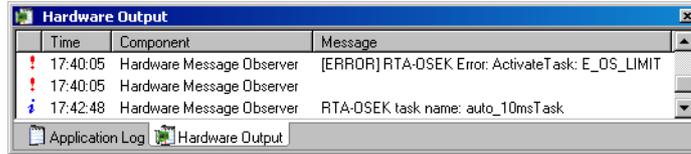
デフォルトでは **Scaled Time** が選択され、ファクタが 1 に設定されています。

Windows のリアルタイム性能には限界があるため、システム負荷が高い処理においては遅延や障害が発生する可能性があり、それによって変則的なデータが測定される場合があります（下図の矢印の箇所）。



このため、仮想プロトタイピング実験を行う時には、リソースを消費する他のプログラムを終了しておくことをお勧めします。

障害が発生すると、どのタスクが実行されなかったかを通知するメッセージが“Hardware Output” タブに出力されます。



操作手順: VP 実行モードを変更する

1. 測定とオペレーティングシステムの両方を停止します。

注記

オペレーティングシステムを実行させたままシミュレーションタイムモードを変更することはできません。

2. “VP Execution Mode” ウィンドウで以下のいずれかを行います。
 - **Adaptive Time** モードを選択する
 - **Scaling Factor** の値を変更する（142 ページ参照）
3. オペレーティングシステムを再起動して測定を再開し、データ表示の違いを確認してください。

5.9 レッスン 8: ES930 — 設定と操作の例

注記

このレッスンの実験を行うには、ES910.3 ハードウェア、HSP V12.1 以降、デジチェーンコンフィギュレーションツール、および所定のケーブルが必要です。

マルチ I/O モジュール ES930 は ES830 ターゲットにも接続できますが、その構成に関する情報は本レッスンには含まれません。

目標:

ES910 に接続された ES930 を使用する方法を学習します。ES930 モジュールを含むデジチェーンのコンフィギュレーションを手動操作で設定し、そのコンフィギュレーションを ETAS Experiment Environment でテストします。

5.9.1 本レッスンの最重要コンセプトの概要

ハードウェア接続と INTECRIO-RP / VP ハードウェアコンフィギュレータ

制御プラント（「テクニカルプロセス」とも呼ばれます）は、センサとアクチュエータのセットを表し、制御システムは、各種ハードウェアを経由してこれらのものに接続します。ここで使用されるハードウェアは、モデリングされたテクニカルプロセスに適合するように設定する必要があります。

ハードウェア接続の作業は INTECRIO のハードウェアコンフィギュレータ（「HC」とも呼ばれます）で行います。

INTECRIO-RP のハードウェアコンフィギュレータを使用すると、該当するハードウェアシステムの構成の定義とパラメータ設定を行うことができ、ここで設定されたモデルが「コントローラ環境」にロードされ、ハードウェアシステムに組み込まれたインターフェースを制御します。

このようにして、外部ソースからのシグナルをコントローラモデル内の適切な箇所で確実に取得するためのロジックが生成されます。

デージーチェーンコンフィギュレーション

一般的なインターフェースとは異なり、デージーチェーンは INTECRIO 内で設定することができません。専用のデージーチェーンコンフィギュレーションツールを用いて設定し、ツールで作成したコンフィギュレーションファイルを INTECRIO にインポートします。

ES930

ES930 マルチ I/O モジュールは、コンパクトでありながら耐久性に富んだパワフルなツールです。各種入出力チャンネルを搭載しています。

このモジュールで ES910 の機能を拡張することにより、ファンクションモデル (Simulink®、ASCET、AUTOSAR、C コード) から直接センサを読み取ったりアクチュエータを制御したりすることが可能になります。

5.9.2 ES930 デージーチェーンモジュールの設定

コンフィギュレーションの設定は ETAS のデージーチェーンコンフィギュレーションツールで行います。その後、ツールで作成した *.xml コンフィギュレーションファイルを INTECRIO にインポートします。

操作手順: 新しいデージーチェーンコンフィギュレーションを作成する

1. ES930 を ES910 に接続します。
2. ES910 を PC に接続します。
3. デスクトップアイコンなどからデージーチェーンコンフィギュレーションツールを起動します。
“ハードウェアコンフィギュレーション”ウィンドウが開きます。



注記

このウィンドウについての詳しい情報は、オンラインヘルプを参照してください。



4. “ハードウェアコンフィギュレーション”ウィンドウに ES930 モジュールを追加します。



5. チェーンの IP アドレスを入力します。
6. データ転送の遅延が最小化されるように **Hardware > Activate Rapid Prototyping** を選択します。

デフォルトでは、すべてのデジタル入力/出力、アナログ入力/出力、温度チャンネルがイネーブルになっています。このチュートリアルでは 2 つのデジタル出力チャンネルと 1 つの温度チャンネルしか使用しないので、それ以外のチャンネルはディセーブル (= 無効) にしてください。

操作手順：温度チャンネルを設定する

1. 温度センサを ES930 モジュールの Thermo チャンネルの 1 つに接続します。
2. “ハードウェアコンフィギュレーション”ウィンドウの“ネットワーク構成”ペインから、“温度”ノードを選択します。
3. “温度入力”タブで、センサを接続したチャンネルをイネーブル（チェックマークをオン）にします。
4. レート、測定範囲 min、測定範囲 max に適切な値を入力します。
5. 必要に応じて、使用していない温度チャンネルもイネーブルにして、診断を行えるようにします。

以下に設定例を示します。

温度入力

No.	選択	名前	単位	レート	センサタイプ	変換式	測定範囲 min	測定範囲 max	実際の測定範囲 min	実際の測定範囲 max	コメント
1	<input checked="" type="checkbox"/>	ES930_TH1_CH1	DegC	0.1s	K	Sensor = phys	0.00	150.00	-200.00	1372.00	
2	<input checked="" type="checkbox"/>	ES930_TH1_CH2	DegC	0.1s	K	Sensor = phys	0.00	150.00	-200.00	1372.00	
3	<input checked="" type="checkbox"/>	ES930_TH1_CH3	DegC	0.1s	K	Sensor = phys	0.00	150.00	-200.00	1372.00	
4	<input checked="" type="checkbox"/>	ES930_TH1_CH4	DegC	0.1s	K	Sensor = phys	0.00	150.00	-200.00	1372.00	

操作手順：診断チャンネルを設定する

1. “ハードウェアコンフィギュレーション”ウィンドウの“ネットワーク構成”ペインから、“診断”ノードを選択します。
2. ES930_D_TH1_CH<n>_OpenWire チャンネルをイネーブルにします。

実験中、これらのチャンネルは対応する温度チャンネルが使用されていないときには true を返し、使用されているときには false を返します。



3. “ハードウェアコンフィギュレーション”ウィンドウでオンライン測定を開始します。

下のスクリーンショットはオンライン測定実行中の“診断”タブを示しています。

診断

No.	選択	名前	値	単位	レート	コメント
27	<input checked="" type="checkbox"/>	ES930_D_TH1_CH1_OpenWire	True	bit	ES930_DI1_CH1 0.1s	
28	<input checked="" type="checkbox"/>	ES930_D_TH1_CH2_OpenWire	True	bit		
29	<input checked="" type="checkbox"/>	ES930_D_TH1_CH3_OpenWire	False	bit		
30	<input checked="" type="checkbox"/>	ES930_D_TH1_CH4_OpenWire	True	bit		
31	<input type="checkbox"/>	ES930_D_D01_CH1_MinPulseAd	---	bit		
32	<input type="checkbox"/>	ES930_D_D01_CH2_MinPulseAd	---	bit		

操作手順：デジタル出力チャンネルを設定する

1. “ハードウェアコンフィギュレーション”ウィンドウの“ネットワーク構成”ペインから、“デジタル出力”ノードを選択します。

2. 第 1 チャンネルと第 2 チャンネルをイネーブルにします。
3. 両方のチャンネルについて、**PWM 出力モード** を選択します。
4. 各チャンネルのアクティブステートを選択します。
5. 各チャンネルにユーザー LED を 1 つずつ割り当てます。
6. **アクティブ時間** と **パルス周期** の初期値を定義します。
以下に設定例を示します。

No.	設定	選択	HWチャンネル	アクティブステート	モード	名前	更新モード	アライメント	LED	電力線	初期値			
											出力	アクティブ時間	パルス周期	単位
1		<input checked="" type="checkbox"/>	CH1	High	PWM出力	ES930_D01_CH1_ActiveTime	個別	N/A	U1	なし	N/A	500.00	N/A	msec
2		<input checked="" type="checkbox"/>		ES930_D01_CH1_PeriodTime	個別	N/A	N/A	N/A	N/A	N/A	N/A	2000.00	N/A	msec
3		<input type="checkbox"/>		N/A	N/A	ES930_D01_CH1_Enable	N/A	N/A	N/A	N/A	イネーブル	N/A	N/A	bit
4		<input checked="" type="checkbox"/>	CH2	Low	PWM出力	ES930_D01_CH2_ActiveTime	個別	N/A	U2	なし	N/A	1000.00	N/A	msec
5		<input checked="" type="checkbox"/>		ES930_D01_CH2_PeriodTime	個別	N/A	N/A	N/A	N/A	N/A	イネーブル	N/A	2000.00	msec
6		<input type="checkbox"/>		N/A	N/A	ES930_D01_CH2_Enable	N/A	N/A	N/A	N/A	イネーブル	N/A	N/A	bit
7		<input type="checkbox"/>	CH3	High	デジタル出力	ES930_D01_CH3	個別	N/A	なし	なし	非アクティブ	N/A	N/A	bit
8		<input type="checkbox"/>		ES930_D01_CH3_PeriodTime	個別	N/A	N/A	N/A	N/A	N/A	イネーブル	N/A	N/A	msec

操作手順：デジチェーンのコンフィギュレーションを完成させる



1. 設定が終わったら、デジチェーンを初期化します。



2. コンフィギュレーションファイルを保存します。

指定されたディレクトリに *.xml ファイルと *.gcf ファイルが書き込まれます。INTECRIO で ES930 を使用する際は、この *.xml ファイルが必要です。

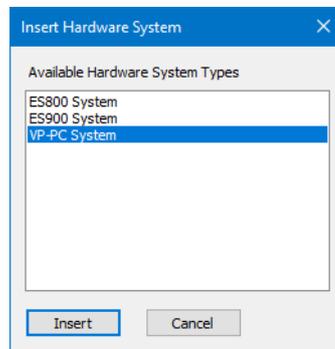
デジチェーンコンフィギュレーションツールで行う作業はここまでです。

5.9.3 ES910 ハードウェアシステムの作成とセットアップ

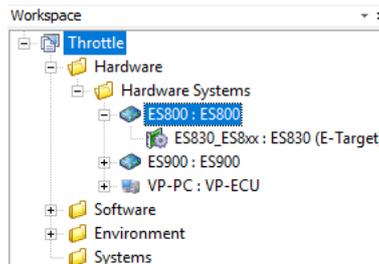
5.9.3.1 ES900 システム

操作手順：ES900 システムを作成する

1. ワークスペースブラウザ上の Hardware フォルダ下の Hardware Systems フォルダを右クリックし、ショートカットメニューから **Add Hardware System** を選択します。
“Insert Hardware System” ダイアログボックスが開き、使用できるハードウェアシステムの一覧 (ES800 System、ES900 System、VP-PC System) が表示されます。



2. ES900 System を選択して **Insert** をクリックし、ES900 システムを追加します。
“Rename” ダイアログボックスが開いてデフォルト名 ES900 が表示されます。
3. 必要に応じて異なる名前を入力し、**OK** をクリックします。
ワークスペースブラウザに ES900 システムが追加され、表示されます。



空のシステムが作成されたので、ここにスロットル制御に必要なデバイスを追加する必要があります。ここでは ES910 ターゲットとデジチェーンを順に追加します。

5.9.3.2 ES910 シミュレーションターゲット

ES910 シミュレーションターゲットは、ハードウェアコンフィギュレータで作成するハードウェア構成全体のマスタデバイスとして機能します。INTECRIO V5.0 以降では、1 つのターゲットしかサポートしていないハードウェアシステムについては、自動的にターゲットが追加されるので、その下にその他のデバイスを追加していきます。

ターゲットを削除してしまった場合は、手動で追加することができます。詳細は INTECRIO オンラインヘルプを参照してください。

操作手順：実験ターゲットを設定する

1. ワークスペースブラウザに表示されている ES910 をダブルクリックします。

ES910 用のハードウェアコンフィギュレータが開き、デバイスパラメータが表示されます。

パラメータ名	意味
Name	デバイスの名前
IO Failure Behavior	I/O 初期化時にエラーが発生した際の対処方法

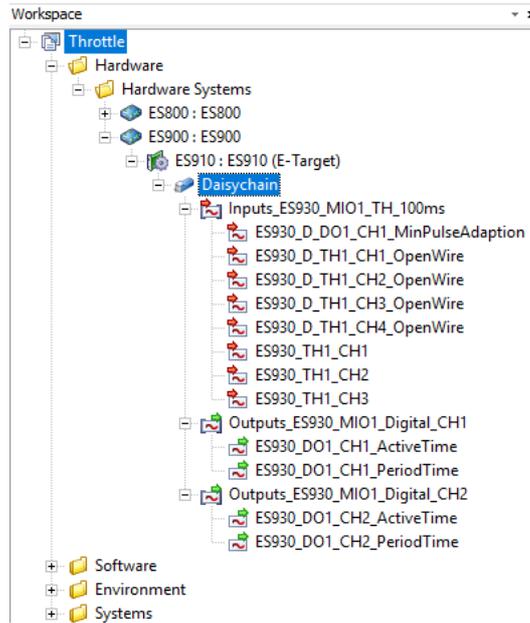
2. ここではプリセットされているデフォルト値をそのまま使用します。

5.9.3.3 デジチェーンデバイス

操作手順：デジチェーンデバイスを作成する

1. ワークスペースブラウザ内の ES910 を右クリックし、ショートカットメニューから **Insert** を選択します。
2. “Insert new Item” ダイアログボックスで Daisychain を選択します。

3. **Insert** をクリックします。
4. デフォルトの名前を変更する場合は “Rename” ダイアログボックスに新しい名前を入力し、**OK** をクリックします。
ターゲットに Daisychain デバイスが追加されますが、その内容はまだ空の状態です。
5. Daisychain デバイスを右クリックし、ショートカットメニューから **Import Daisychain Configuration** を選択します。
インポートウィザードが開きます。
6. ウィザードの指示に従い、5.9.2 項で作成した *.xml デイジーチェーンコンフィギュレーションファイルをインポートします。



次に、このデバイスを「アクティブ化」します。

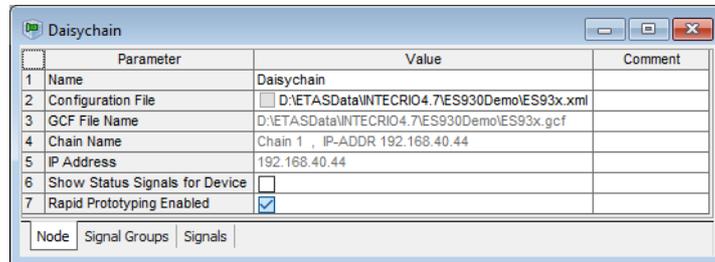
操作手順：デイジーチェーンデバイスをアクティブにする

1. Daisychain デバイスのショートカットメニューを開きます。
Enabled メニューコマンドの横にチェックマークが付いている場合、このデバイスはアクティブになっています。
2. そうでない場合は **Enable** を選択してデバイスをアクティブにします。

操作手順：デイジーチェーンデバイスを設定する

1. ワークスペースブラウザ内の Daisychain を右クリックし、ショートカットメニューから **Open** を選択します。

デイジーチェーンデバイス用のハードウェアコンフィギュレータが開き、"Node" タブにデバイスパラメータが表示されます。



"Node" タブには一般的なデバイスパラメータが表示されます。各パラメータについてはオンラインヘルプに詳しく説明されています。

値を編集するには、その値のセルをクリックします。パラメータの種類により、直接入力する場合と、ドロップダウンリストから値を選択する場合があります。

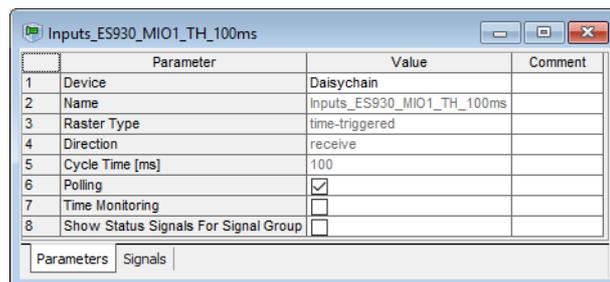
灰色の背景で表示されたフィールドは編集できません。

2. デバイスのステータスシグナルをアクティブにします。
3. 必要に応じて、パラメータの "Comments" 欄にコメントを入力します。

操作手順：シグナルグループを設定する

1. ワークスペースブラウザ内のシグナルグループ Inputs_ES930_MIO1_TH_100ms をダブルクリックします。

シグナルグループのハードウェアコンフィギュレータが開き、シグナルグループのパラメータが表示されます。各パラメータについてはオンラインヘルプに詳しく説明されています。



灰色の背景で表示されたフィールドは編集できません。

2. 必要に応じて、パラメータの "Comments" 欄にコメントを入力します。
3. シグナルグループをアクティブにします。
4. 他のシグナルグループを設定します。

上記の方法以外に、デイジーチェーンのハードウェアコンフィギュレータ（149 ページ参照）を開いて "Signal Groups" タブを選択すると、そこでもシグナルグループの設定（コメント以外）が行えます。

操作手順：シグナルを設定する

1. 以下のいずれかを行います。
 - デイジーチェーンのハードウェアコンフィギュレータで "Signals" タブを選択します。
全シグナルグループに含まれるすべてのシグナルのパラメータが表示されます。
 - シグナルグループのハードウェアコンフィギュレータで "Signals" タブを選択します。
そのシグナルグループに含まれるシグナルのパラメータが表示されます。
 - ワークスペースブラウザ上のシグナルをダブルクリックします。
そのシグナルのパラメータを含むハードウェアコンフィギュレータが開きます。
シグナルパラメータについてはオンラインヘルプに詳しく説明されています。
灰色の背景で表示されたフィールドは編集できません。
2. 必要に応じて、シグナルパラメータの "Comments" 欄にコメントを入力します。

これでハードウェアシステムの定義が完了したので、これをシステムプロジェクトに追加することができます。

その他のハードウェアデバイスの使用方法については、オンラインヘルプを参照してください。

5.9.4 ES930 による実験**5.9.4.1 システムプロジェクトの作成**

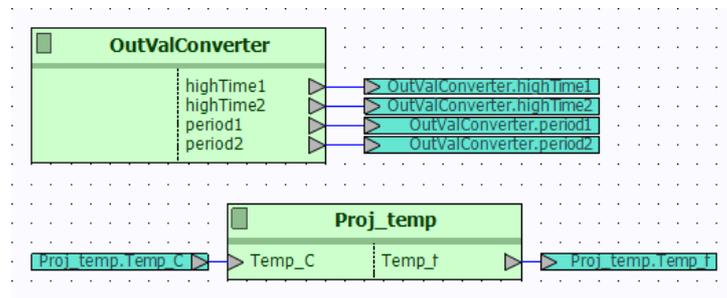
実験を行うには完全なシステムプロジェクトが必要です。このために、以下の2つのソフトウェアモジュールが用意されています。

- OutValConverter は、周波数とデューティサイクルを使用して、ES930 のデジタル出力チャンネルの HIGH タイム（アクティブタイム）とパルス周期の値を算出します。
- Proj_temp は Thermo チャンネルで測定された温度を °C から °F に変換します。

操作手順：システムプロジェクトを作成する

1. 103 ページの説明に従い、ソフトウェアモジュール OutValConverter および Proj_temp をインポートします。
*.six ファイル、およびその他の必要なファイルは <sample files>¥ES930Demo ディレクトリのサブディレクトリ OutValConverter および Proj_temp にあります。
2. ソフトウェアシステム（109 ページ参照）を作成し、それに名前を付けます（例：SWS_ES930）。

3. インポートしたモジュールをソフトウェアシステムに追加し（109 ページ参照）、必要なインターフェースを下図のように追加します（108 ページと 110 ページ参照）。

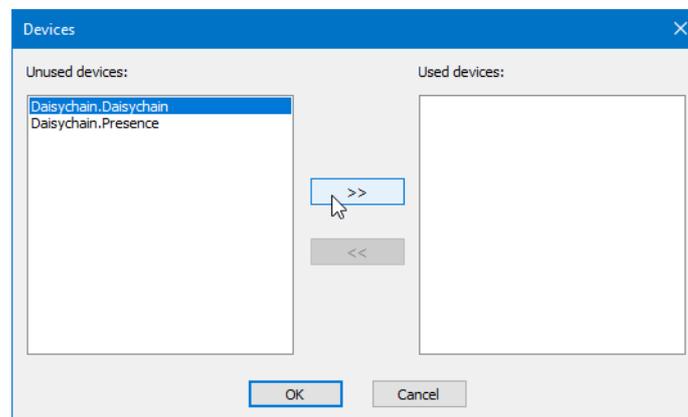


4. 113 ページの説明に従ってシステムプロジェクトを作成し、それをアクティブシステムにして名前（例：SP_ES930）を付けます。
5. ハードウェアシステムとソフトウェアシステムをシステムプロジェクトに統合します（114 ページと 115 ページ参照）。

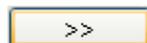
操作手順: ハードウェアシステムをセットアップする

1. ワークスペースブラウザで SP_ES930 システムプロジェクトを右クリックし、ショートカットメニューから **Devices** を選択します。

“Devices” ダイアログボックスが開きます。



このダイアログボックスの使用方法は、105 ページの “Modules” ダイアログボックスと同じです。ここにはハードウェアシステムに含まれるデバイスが表示されます。



2. “Unused modules” フィールド内の Daisychain.Daisychain というエントリを “Used devices” フィールドに移動します。
Daisychain.Presence というエントリにはデバイスの状態を表すシグナルが含まれますが、このチュートリアルでは使用しません。
3. **OK** をクリックします。
デージーチェーンがシステムプロジェクトで使用できるようになりました。

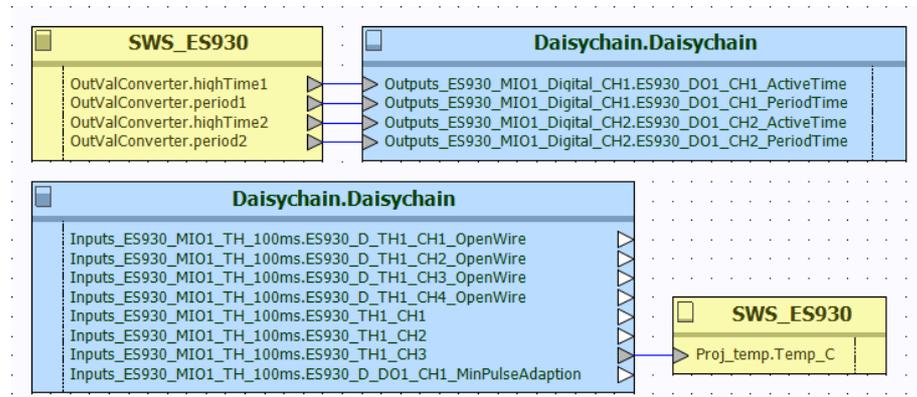
操作手順: ハードウェアとソフトウェアを接続する

1. グラフィックエディタで SP_ES930 システムプロジェクトを開きます。

システムプロジェクトがブロック図で表示されます。

ハードウェアデバイスは青いブロックのセットとして表示されます。ソフトウェアシステムのシグナルシンクに接続できるハードウェアチャンネルは、1つのブロックの右側にまとめられ、ソフトウェアシステムのシグナルソースに接続できるハードウェアチャンネルは、もうひとつのブロックの左側にまとめられています。

2. 下図のように接続します (116 ページ参照)。



表示を見やすくするため、使用されていない SWS ポートは非表示になっています。

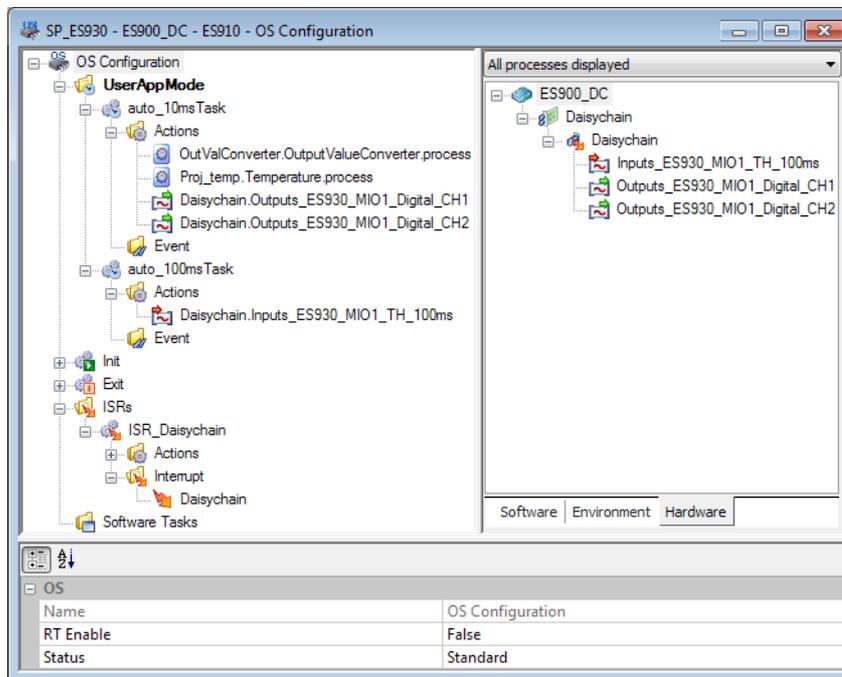
操作手順: オペレーティングシステムを設定する

1. OSC を開きます。



2. **OS Auto Mapping** ボタンをクリックして、OS コンフィギュレーションの自動設定を行います。

必要なタスクと ISR が作成され、“Software” タブと “Hardware” タブのプロセスが自動的に割り当てられます。



システムプロジェクトを作成してコンパイラを選択したら、実行ファイルを生成します。

操作手順: プロトタイプを生成する

1. ビルドオプションを設定します (125 ページ参照)。
2. プロトタイプを生成します (125 ページ参照)。

実行ファイル `SP_ES930.rta` が生成され、ワークスペースの出力用ディレクトリ (例: `ETASData\INTECRIO5.0\Tutorial\ES930_demo\Results`) に格納されます。

5.9.4.2 ES910 の設定

デージーチェーン用の IO インターフェースを使用するには、ES910 のポート設定をグラフィックユーザーインターフェースで確認する必要があります。この確認を行うには、ES910 を PC に接続してウェブブラウザアプリケーションを使用します。

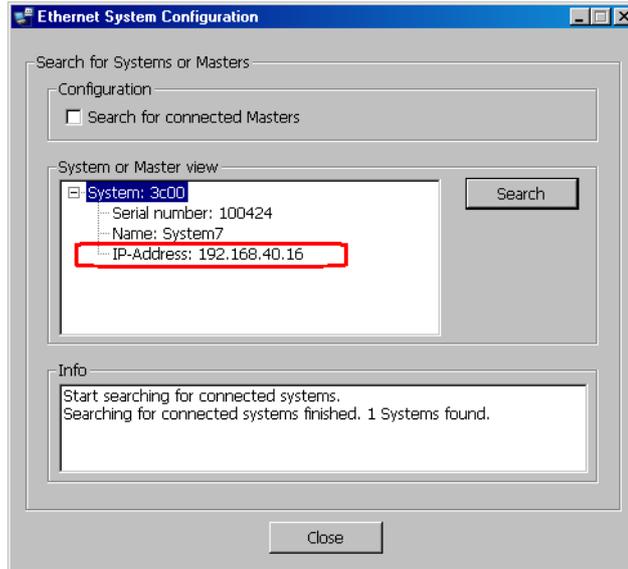
ES910 ユーザーインターフェースは、下記とは異なる方法で開くこともできます。詳しくは ES910 ユーザーガイドを参照してください。

操作手順：IP アドレスを確認してコンフィギュレーションインターフェースを開く

1. Windows のシステムトレイに表示されている **IP-Manager** アイコンを右クリックして、ショートカットメニューから **Ethernet System Configuration** を選択します。

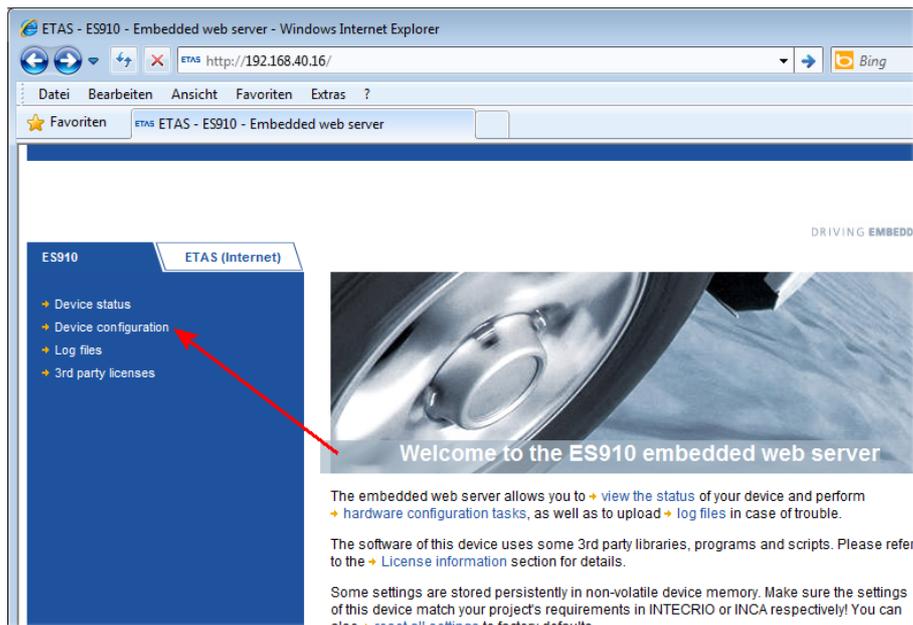


“Ethernet System Configuration” ダイアログボックスが開きます。



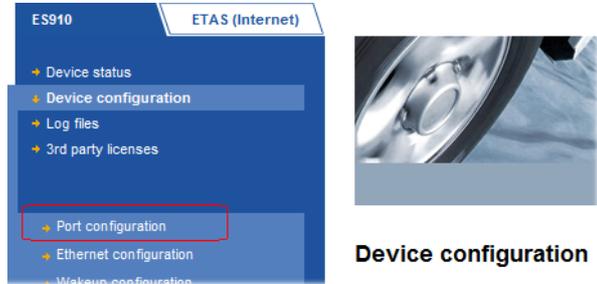
“System or Master view” フィールドに IP アドレスが表示されます。

2. この IP アドレスを Web ブラウザに入力し、ES910 ユーザーインターフェースを開きます。



操作手順: ES910 を設定する

3. ES910 ユーザーインターフェースのスタートページにある、[Device configuration](#) リンクをクリックします。
4. “Device configuration” ページの “Device configuration” というキャプションの下にある [Port configuration](#) リンクをクリックします。



5. Micro I/O interface の **ES400 daisy chain** オプションがオンになっていることを確認します。



6. コンフィギュレーションを保存します。

5.9.4.3 実験の実行

以上の設定で、実験の準備は完了です。6 個の数値インストゥルメントと 4 個の LED インストゥルメントを使用して、以下の変数の測定と適合を行います。

変数	ソース	備考
dutycycle1	SW	OutValConverter
frequency1	モジュール	モジュールの入力
dutycycle2	パラメータ	
frequency2		
highTime1	SW	OutValConverter
period1	モジュール変数	モジュールの出力
highTime2		
period2		
ES930_TH1_CH3	HW 温度チャンネル	測定された温度
Temp_C	SW	Proj_temp モジュール
Temp_F	モジュール変数	の入力/出力
ES930_D_TH1_CH1_OpenWire	HW	温度チャンネルが接続
ES930_D_TH1_CH2_OpenWire	診断チャンネル	されている (false)
ES930_D_TH1_CH3_OpenWire		かない (true) かを
ES930_D_TH1_CH4_OpenWire		調べます。

操作手順: 実験をセットアップする

1. ETAS Experiment Environment を開きます (126 ページ参照)。

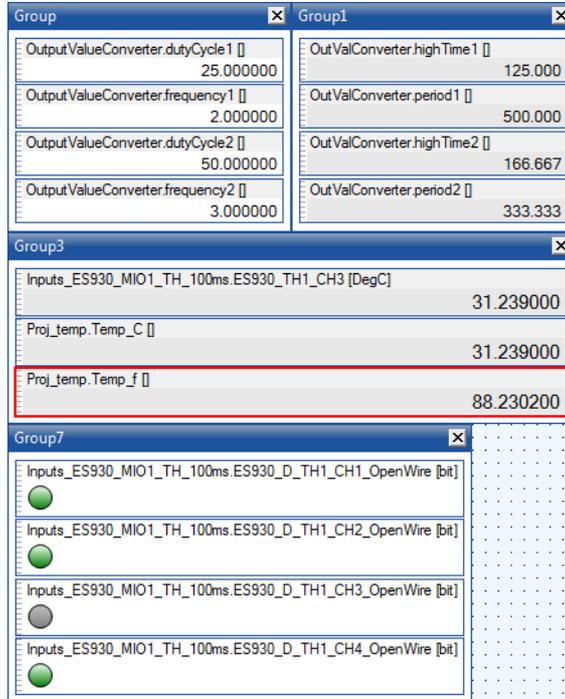
2. 以下の変数用に 3 つのグループの数値インストゥルメントを作成します (129 ページ参照)。

グループ	変数
1	dutycycle1
	frequency1
	dutycycle2
	frequency2
2	highTime1
	period1
	highTime2
	period2
3	ES930_TH1_CH3
	Temp_C
	Temp_F

3. 変数 ES930_D_TH<n>_CH<n>_OpenWire 用に、4 個の LED インストゥルメントからなるグループを 1 つ作成します。
4. 実験を保存します (135 ページ参照)。

操作手順：実験を実行する

1. シミュレーションと測定を開始します（137 ページ参照）。
 数値ディスプレイと LED インストゥルメントに測定値が表示されます。



ES930 のフロントパネルにあるユーザー LED（U1 と U2）が点滅します。



2. 第 1 グループのパラメータを編集し、ES930 上にあるユーザー LED の点滅パターンの変化を観察します。

6 ETAS ネットワークマネージャ

ネットワークアダプタの設定は、ETAS ネットワークマネージャを使用して行います。ETAS ネットワークマネージャは、ETAS IP マネージャが使用するコンフィギュレーション（ネットワーク構成）を作成するためのツールで、IP マネージャは、ネットワーク接続された ETAS ハードウェアのダイナミック IP アドレッシングを行います。

ETAS ネットワークマネージャは、Windows スタートメニューのアプリ一覧から表示できます。例：**E > ETAS > ETAS Network settings**

ETAS ネットワークマネージャについての詳しい情報は、ネットワークマネージャのオンラインヘルプを参照してください。オンラインヘルプを開くには、ネットワークマネージャで **<F1>** キーを押します。

7 一般的なトラブルシューティング

この章では、個々のソフトウェアやハードウェアに依存しない一般的な問題が発生した場合の対処法について説明します。

7.1 ETAS ネットワーク用のネットワークアダプタを選択できない

原因：APIPA が無効になっている

IP アドレッシングの代替メカニズムである APIPA は、Windows XP / Vista 環境では、通常は有効に設定されていますが、時にはネットワークセキュリティポリシーによって無効となっている場合もあります。そのような場合、DHCP 設定のネットワークアダプタ（ネットワークカード）を ETAS ハードウェアのアクセスに使用することはできず、そのアダプタを選択すると ETAS ネットワークマネージャは警告メッセージを表示します。

無効になっている APIPA メカニズムを有効にするには、Windows のレジストリを編集する必要がありますが、これを行うには管理者の権限が必要です。ネットワーク管理者の方にご相談のうえ行ってください。

操作手順：APIPA メカニズムを有効にする

1. 以下のように操作します。
 - i. <Windows ロゴ> + <R> を押します。
 - ii. regedit と入力し、OK をクリックします。レジストリエディタが開きます。

2. 以下のフォルダを開きます。

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\
```

3. **編集 > 検索** を選択して以下のキーを検索します。

```
IPAutoconfigurationEnabled
```

注記

APIPA メカニズムが無効になっていない場合、レジストリ内にこのキーは存在しません。このキーが存在する場合のみ、以下の操作で APIPA メカニズムを有効にしてください。

4. APIPA メカニズムが有効になるように、見つかったキーの値をすべて 1 に変更します。

Windows のレジストリ内には、この名前のキーがいくつか含まれている場合があります。これらは一般的な TCIP/IP サービス用のものと、個別のネットワークアダプタ用のものです。ETAS ネットワーク用に使用するアダプタについてのみ値を変更してください。

5. レジストリエディタを閉じます。
6. 変更されたレジストリの内容を有効にするため、PC を再起動します。

7.2 PC に接続されたイーサネットハードウェアが検出できない

接続エラーの原因としてはさまざまなものが考えられますが、一般的には Windows やハードウェアの設定が正しく行われていないことが原因となることが多く、これらの原因が特定できれば容易にエラーを解決することができます。

以下に原因を特定して解決するためのヒントをいくつかご紹介します。

原因：ハードウェアバージョンと ETAS ソフトウェアのバージョンに互換性がない

ETAS ハードウェアを ETAS ソフトウェアに接続して使用する場合、ETAS の HSP アップデートツールを利用して、以下のようにハードウェアのファームウェアバージョンをチェックすることができます。

- 最新の HSP (Hardware Service Pack : ハードウェアサービスパック) に含まれる HSP アップデートツールを使用します。
- HSP アップデートツールを使用して、ハードウェアのバージョンがソフトウェアのバージョンと互換性があるかどうかをチェックします。
- PC ドライバが必要なハードウェアの場合、ドライバが正しくインストールされているかどうかをチェックします。

最新バージョンの HSP は、ETAS ホームページのダウンロードセンター (https://www.etas.com/ja/products/download_center.php) からダウンロードできます。

最新の HSP アップデートツールでも検出されない場合は、そのハードウェアが Web インターフェースを実装しているかを確認してください。実装している場合は Web インターフェース経由で検索し、それでも見つからない場合は、以下の原因が考えられます。

原因：パーソナルファイアウォールによる通信のブロック

パーソナルファイアウォールは、ETAS のイーサネットハードウェアへのアクセスを妨害する場合があります。そのような場合、ハードウェアの自動検索時に、コンフィギュレーションが正しく設定されているにもかかわらずイーサネットハードウェアがまったく検出されない、という状態が発生する可能性があります。

また、ファイアウォールが適切に設定されていないと、ETAS ソフトウェアにおける特定の操作 (例 : ASCET で実験を開く、INCA や HSP でハードウェアを検索する、など) を実行する際に不具合が発生する場合があります。

ファイアウォールによって ETAS ハードウェアとの通信がブロックされる場合は、ETAS ソフトウェア使用中はファイアウォールソフトウェアを無効にするか、またはファイアウォールの詳細設定を行って以下のアクセスを許可するようにしてください。

ETAS ハードウェアに関するアクセス許可

- UDP を経由する、デスティネーション IP アドレス 255.255.255.255 への送信用 IP ブロードキャスト (デスティネーションポート : 17099 または 18001)
- UDP を経由する、ソース IP アドレス 0.0.0.0 からデスティネーション IP アドレス 255.255.255.255 への受信用 IP ブロードキャスト (デスティネーションポート : 18001)

- UDP を経由する、ETAS ネットワークへの直接 IP ブロードキャスト（デスティネーションポート：17099 または 18001）
- UDP を経由する、ETAS ネットワーク内のすべての IP アドレスへの送信用 IP ユニキャスト（デスティネーションポート：17099 ～ 18001）
- UDP を経由する、ETAS ネットワーク内のすべての IP アドレスからの受信用 IP ユニキャスト（ソースポート：17099 ～ 18020、デスティネーションポート：17099 ～ 18020）
- ETAS ネットワーク内への送信用 TCP/IP 接続（デスティネーションポート：18001 ～ 18020）

注記

実際のポート番号は、使用するハードウェアに応じて異なります。ポート番号についての詳しい情報は、ハードウェアのドキュメントを参照してください。

XCP on Ethernet に関するアクセス許可

- UDP 経由の XCP スレーブ検知のための、ETAS ネットワーク内のすべての IP アドレスへの送信用 IP マルチキャスト（デスティネーション IP 239.255.0.0、ポート：5556）
- UDP 経由の XCP スレーブ検知のための、ETAS ネットワーク内のすべての IP アドレスからの受信用 IP マルチキャスト（デスティネーション IP 239.255.37.45、ポート：3745）

アクセス許可についての詳細（実際にどのような許可が必要か、またその許可を与えることは可能であるか、など）は、IT 担当の方にご確認ください。

ご注意ください！

ファイアウォール構成の変更によるシステムの安全性の低下への影響

ファイアウォール設定を変更して PC をネットワークに再接続する際には、前もって社内の IT セキュリティポリシーをご確認ください。その際には IT 担当の方にご相談いただくことをお勧めします。

原因：リモートアクセス用クライアントソフトウェアによる通信のブロック

ETAS ハードウェアネットワーク外で使用されている PC には、リモートアクセス用クライアントソフトウェアがインストールされているものがあり、それによって ETAS ハードウェアへのアクセスが妨害される場合があります。それには以下のような状況が考えられます。

- イーサネットメッセージをブロックするファイアウォールが使用されている（160 ページの「原因：パーソナルファイアウォールによる通信のブロック」を参照してください）
- トンネリングに使用されている VPN クライアントソフトウェアが誤ってメッセージをフィルタリングしてしまうことがあります。たとえば Cisco VPN クライアントの V4.0.x より前のバージョンでは、特定の UDP ブロードキャストを不正にフィルタしてしまう、というケースがありました。

このケースに該当する場合は、VPN クライアントソフトウェアのバージョンをアップデートしてください。

原因：ETAS ハードウェアのフリーズ

ETAS ハードウェアが何らかの理由でフリーズしてしまった可能性もあります。この場合は、ハードウェアの電源を切ってから再投入してください。これによってハードウェアは再初期化されるので、多くの場合、正常に戻ります。

原因：ETAS ハードウェアがスリープモードになっている

ETAS ハードウェアには、省電力の目的で、他のデバイスは PC に接続されていないときにはスリープモードに切り替わるものがあります。

スリープモードに切り替わらないようにするには、PC のイーサネットケーブルをハードウェアの“HOST” / “Sync In” ポートに接続し、ハードウェアの起動後に Web インターフェースでハードウェアに接続し、設定を変更してください。詳しくはハードウェアのマニュアルを参照してください。

原因：ネットワークアダプタへの IP アドレス割り当てが一時的に失われた

PC の接続を、DHCP が使用されている社内 LAN から ETAS ハードウェアに切り替える際、PC が ETAS ハードウェアを検知できるようになるまでに約 60 秒かかります。これはオペレーティングシステムが DHCP プロトコルから ETAS ハードウェア用の APIPA に切り替わる処理に要する時間です。

原因：ETAS ハードウェアが他の論理ネットワークに接続されている

1 つの ETAS ハードウェアに対して複数の PC からアクセスする場合、各 PC で使用されるネットワークアダプタは、同じ論理ネットワークを使用するように設定しておく必要があります。このように設定することが不可能である場合、他の PC を切り替える前に ETAS ハードウェアの電源を切って再投入してください。

原因：ネットワークアダプタ用のデバイスドライバが起動していない

ネットワークアダプタ用のデバイスドライバが起動していない可能性があります。その場合は、ネットワークアダプタを一旦無効にしてから再度有効にしてください。

操作手順：ネットワークアダプタを無効にして、再度有効にする

1. Windows の **コントロールパネル** を開きます。
2. **ネットワークと共有センター** を開き、さらに **アダプターの設定の変更** リンクをクリックします。
3. **ネットワークの接続** ウィンドウで、ETAS ネットワーク用に使用されているデバイスを右クリックし、ショートカットメニューから **無効にする** を選択します。
4. 続いて同じショートカットメニューから **有効にする** を選択し、アダプタを有効にします。

原因：ラップトップ PC の電源管理システムによってネットワークアダプタが無効になっている

ラップトップ PC の電源管理システムにより、ネットワークアダプタが無効になっている場合があります。その場合は、ラップトップ PC の電源管理を無効にしてください。

操作手順: ラップトップ PC の電源管理を無効にする

1. Windows のスタートメニューから、コントロールパネル > ハードウェアとサウンド > デバイスマネージャ を選択します。
2. デバイスマネージャ ウィンドウで ネットワークアダプタ のツリーを展開します。
3. 使用するネットワークアダプタを右クリックし、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. 電源の管理 タブを選択し、コンピュータでこのデバイスの電源をオフできるようにする オプションをオフにします。
5. 詳細設定 タブを選択し、プロパティ に **Autosense** が含まれている場合は、これを無効 (**Disabled**) にします。
6. **OK** をクリックして設定を有効にします。

原因: ネットワークの自動切断

ネットワークアダプタのデータトラフィックが一定の時間途絶えると、ネットワークアダプタが自動的にイーサネット接続を切断する場合があります。これは、レジストリの autodisconnect キーを設定することによって避けることができます。

操作手順: レジストリキー autodisconnect を設定する

1. レジストリエディタを開きます。
2. HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\lanmanserver\parameters というフォルダに含まれるレジストリキー autodisconnect の値を 0xffffffff に変更します。

8 お問い合わせ先

製品に関するご質問等は、各地域の ETAS 支社までお問い合わせください。

ETAS 本社

ETAS GmbH

Borsigstrasse 24	Phone:	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	WWW:	www.etas.com/

日本支社

イータス株式会社

〒 220-6217		
神奈川県横浜市西区	Phone:	(045) 222-0900
みなとみらい 2-3-5	Fax:	(045) 222-0956
クイーンズタワー C 17F	WWW:	www.etas.com/

その他のお問い合わせ先

上記以外のお問い合わせ先につきましては、ETAS ホームページをご覧ください。

各国支社	WWW:	www.etas.com/ja/contact.php
技術サポート	WWW:	www.etas.com/ja/hotlines.php

9 用語集

この用語集では、INTECRIO のマニュアルに使用されている専門用語と略語について解説します。ここに紹介する多くの語は一般的な用語としても用いられていますが、この用語集では、各語が特に INTECRIO について使用される場合の意味が説明されています。

用語はアルファベット順および 50 音順に並んでいます。

9.1 略語

API

Application Programming Interface (アプリケーション開発者向けプログラミングインターフェース)

ASAM-MCD

自動化および測定システムの標準化を行うための、測定、適合、および診断の作業部会 (ドイツ語: **Arbeitskreis zur Standardisierung von Automations- und Mess-systemen, mit den Arbeitsgruppen Messen, Calibrieren und Diagnose**)

ASAM-MCD-2MC

ECU ソフトウェアに含まれる適合変数と測定変数についての情報を記述するためのファイルフォーマットで、実験用インターフェースについての情報も保存されます。実験を行う際、このファイル (拡張子は A2L) から実験に必要な情報を読み込みます。

INTECRIO V5.0 は、ETK AML の V1.1 (フックベースバイパスのみ) と V1.2 ~ V1.7 (フックベースバイパス/サービスベースバイパス)、XETK AML の V2.5 まで、ASAP1B_Bypass AML の V1.0 以降、SBB AML の V2.0、V3.0、V3.1 をサポートしています。

詳しい情報は <https://www.asam.net> を参照してください。

ASCET

ECU ソフトウェア開発用の ETAS 製品ファミリー – ASCET のモデルを INTECRIO にインポートできます。

ASCET-MD

ASCET Modeling & Design (ASCET 製品ファミリーに含まれる BMT)

ASCET-RP

ASCET Rapid Prototyping (ASCET 製品ファミリーのラピッドプロトタイプングツール)

AUTOSAR

Automotive Open System Architecture (自動車用オープンシステムアーキテクチャ、<https://www.autosar.org> を参照してください)

BMT

Behavioral Modeling Tool (挙動モデリングツール) – BMT では、モデルの挙動を編集し、シミュレーションやアニメーションを行い、ファンクションコードを生成することができます。

BR_XETK

車載イーサネットインターフェースを採用した ETK (エミュレータテストプローブ)

BR_XETK を使用するには、ES882 / ES886 を含む ES800 ハードウェアシステムが必要です。

BSW

Basic software (基本ソフトウェア) – 通信や I/O の機能のほか、各種ソフトウェアコンポーネントに必要な諸機能を提供します。

CAN

Controller Area Network (コントロールエリアネットワーク) – 強靱な車両バス規格のひとつ。ホストコンピュータがなくてもマイクロコントローラと各種デバイスのアプリケーション同士が互いに通信し合うことができます。

CANdb

CAN database – Vector Infomatik 社の CANdb データ管理プログラムで作成された CAN ディスクリプションファイル

INTECRIO V5.0 は CANdb V2.3 以降をサポートしています。

CPU

Central Processing Unit (中央演算処理装置) – INTECRIO では、マイクロコントローラを意味します。

DISTAB

Display Table – ETK バイパス実験において実験ターゲットと ECU との間のデータ交換に使用されるデータ交換メソッド

INTECRIO V5.0 は、DISTAB12 以降 (XETK AML : DISTAB13) を用いたフックベースバイパスと、DISTAB13 を用いたサービスベースバイパスをサポートしています。

INTECRIO V5.0 では、ES830 または ES910.3 を使用することにより DISTAB17 を用いたバイパスも利用可能です。

ECU

Electronic Control Unit (電子制御ユニット) – CPU と周辺機器からなる、小型の組み込み型コンピュータシステム

一般的に、すべてのコンポーネントが 1 つの筐体に収められています。

ETK

Emulator-Taskopf (「エミュレータテストプローブ」を表すドイツ語)

FETK

ES89x ECU / バスインターフェースモジュール対応の ETK

FIBEX

Field Bus Exchange – XML スキーマをベースとするデータ交換フォーマットで、車載通信ネットワーク全体を記述するものです。

FIBEX はさまざまなネットワークタイプ (CAN、LIN、MOST、FlexRay) 用に定義されていて、バスアーキテクチャ、信号、ネットワークノードのプロパティなどについての情報を記述できるようになっています。

FIBEX ファイルフォーマットは ASAM (Association for Standardization of Automation and Measuring Systems : 自動化 / 測定システム標準化委員会) により標準化されています。

INTECRIO V5.0 は FIBEX ベースラインバージョンの FIBEX V2.0.x と V3.1.0 (V3.1.0 については制限事項がありますので、詳しくはオンラインヘルプを参照してください) をサポートしています。

詳しい情報は <https://www.asam.net> を参照してください。

FIFO

First in, first out (先入れ先出し)

HBB

hook-based bypass (フックベースバイパス)

HC

Hardware Configurator (ハードウェアコンフィギュレータ)

IER

INTECRIO Embedded Coder Real-Time Target (INTECRIO に Simulink モデルをインポートするための Embedded Coder リアルタイムターゲット)

INCA

Integrated Calibration and Acquisition Systems (ETAS の測定/適合システム)

INTECRIO V5.0.1 には INCA V7.2.17 以降が対応しています。

INCA-EIP

INCA のアドオンです。INCA からラピッドプロトタイピングターゲット (ES910、ES830) や仮想プロトタイピングターゲット (VP-PC) へのアクセスを実現します。

INTECRIO V5.0.1 には INCA-EIP V7.2.17 以降が対応しています。

INTECRIO-RP

INTECRIO Rapid Prototyping パッケージ — ラピッドプロトタイピングへの接続機能を提供します。

INTECRIO-VP

INTECRIO Virtual Prototyping パッケージ — 仮想プロトタイピングへの接続機能を提供します。

IRT

INTECRIO Real-Time Target (INTECRIO に Simulink モデルをインポートするための Simulink Coder リアルタイムターゲット)

LDF

LIN description File (LIN ディスクリプションファイル) — LIN コントローラ用コンフィギュレーションファイル

INTECRIO V5.0.1 は LDF V1.3、V2.0、V2.1、V2.2 をサポートしています。

LIN

Local Interconnect Network (ローカル相互接続ネットワーク) — 車載コンポーネント間の相互通信に使用されるシリアルネットワークプロトコル
CAN ほどの帯域幅や汎用性を必要としない用途において使用され、一般的な例としては、電気自動車のドアやシートのネットワーク接続などが挙げられます。

LSB および lsb

大文字表記の場合は最下位バイト (**Least Significant Byte**) — 小文字表記の場合は最下位ビット (**least significant bit**)

MDA

Measure Data Analyzer (測定データアナライザ) — 保存された測定データを表示して分析するための ETAS オフラインツール

MSB および msb

大文字表記の場合は最上位バイト (**Most Significant Byte**) — 小文字表記の場合は最上位ビット (**most significant bit**)

OIL

OSEK Implementation Language (OSEK 実装言語) — ECU ネットワーク用記述言語

OIL はオペレーティングシステムを間接的に構成するパーツで、ECU ネットワークの静的情報 (通信接続や ECU のプロパティなど) の記述に使用されます。

OS

Operating System (オペレーティングシステム)

OSC

Operating System Configurator (OS コンフィギュレータ)

OSEK

Offene Systeme für die Elektronik im Kraftfahrzeug (自動車エレクトロニクス向けオープンシステムのための作業部会のドイツ語表記)

PDU

Protocol data unit (プロトコルデータユニット) — プロトコルスタック内のレイヤ間で受け渡しされる情報の単位で、ペイロードと制御情報が含まれます。

INTECRIO V5.0 では、FlexRay の PDU はシグナルグループに相当します。

RE

Runnable Entity (ランナブルエンティティ) — 実行時において RTE からトリガされる一連のコード。INTECRIO で扱われる従来の「プロセス」に相当します。

RP

Rapid prototyping(ラピッドプロトタイピング):173 ページ「ラピッドプロトタイピング」を参照してください。

RTA-OSEK

ETAS のリアルタイムオペレーティングシステム — AUTOSAR-OS V1.0 (SC-1) と OSEK/VDX OS V2.2.3 を実装し、MISRA コンポーネントに完全準拠しています。

RTA-OS

ETAS のリアルタイムオペレーティングシステム — AUTOSAR R3.0 OS と OSEK/VDX OS V2.2.3 を実装し、MISRA コンポーネントに完全準拠しています。

RTA-RTE

ETAS が提供する AUTOSAR ランタイム環境

RTE

AUTOSAR runtime environment (AUTOSAR ランタイム環境) — SWC (ソフトウェアコンポーネント)、BSW (基本ソフトウェア)、OS (オペレーティングシステム) 間のインターフェースとなるものです。

RTIO

Real-Time Input-Output (リアルタイム I/O)

SBB

Service-based bypass (サービスベースドバイパス)

SBC

Sensotronic Brake Control (電気油圧式ブレーキシステム)

SCOOP

Source Code, Objects, and Physics (ソースコード、オブジェクト、および物理的現象)

SCOOP-IX

SCOOP Interface Exchange language (SCOOP インターフェース交換言語)

INTECRIO V5.0 は SCOOP-IX の V1.0、V1.1、V1.2、V1.4、V1.5 をサポートしています。

SP

Service point : 171 ページ「サービスポイント」を参照してください。

SWC

AUTOSAR Software Component (AUTOSAR ソフトウェアコンポーネント) – AUTOSAR アプリケーションにおけるソフトウェアの最小単位

UDP

User Datagram Protocol

UML

Unified Modeling Language (統一モデリング言語)

VFB

AUTOSAR の **Virtual function bus** (仮想ファンクションバス)

VP

Virtual prototyping (仮想プロトタイピング) 171 ページ「仮想プロトタイピング」を参照してください。

XCP

Universal measurement and calibration protocol (汎用測定/適合プロトコル) – x は、さまざまな種類の転送層を使用できることを表します。正式名称は ASAM MCD-1 XCP です。

INTECRIO V5.0 は XCP V1.0 のほか、V1.0 に互換なすべての上位バージョンをサポートしています。さらに、V1.1 以降の XCPplus キーワードもサポートしています。

XETK

イーサネットインターフェースを採用した ETK (エミュレータテストブローブ)

XML

Extensible Markup Language (拡張可能なマークアップ言語)

9.2 用語**AUTOSAR** ソフトウェアコンポーネント

「SWC」を参照してください。

Embedded Coder®

Simulink® Coder™ 用アドオン。Simulink Coder の機能を拡張して、組み込み型ターゲット上での量産アプリケーションの記述、統合、デプロイ、およびテストを可能にします。

FlexRay

FlexRay は高い伝送速度で決定論的データ交換を行うための、拡張可能な耐障害性の通信システム。時分割の手続きにより、モジュール性に優れ安全性の高い分散システムを構築でき、2チャンネル上で10MBaudの高周波数帯域を実現します。現代の自動車は革新的な電子システムの搭載量が年々多くなり、ネットワーク負荷も増大しつつありますが、高周波数帯域の使用によって、このような状況への対応も可能となります。

この通信システムの仕様は、世界中の自動車メーカーとサプライヤに広くサポートされている「FlexRay コンソーシアム」という委員会により策定されました。

INTECRIO

各種 BMT（挙動モデリングツール）で作成された制御アルゴリズムのパーツを統合し、プロトタイピング実験の環境を構築するツール。ハードウェアシステムのコンフィギュレーションを設定し、実験用ハードウェアと制御アルゴリズムを接続することができます。

MATLAB[®]

技術計算用の高性能言語。1つの環境に、数学計算、視覚化、プログラミングなどの機能が備わっています。

MATLAB[®] Coder[™]

MATLABコード用のコードジェネレータ

OS コンフィギュレータおよび OSC

オペレーティングシステムに含まれるタスクは、INTECRIO 内の OS コンフィギュレータによって設定されます。OSC エディタは、この設定を行うためのエディタです。このエディタでは、システム構成を素早く把握でき、アプリケーションからの視点で表示されたソフトウェアの要素を、容易にオペレーティングシステムに組み込むことができます。

RTA-Trace

ECU への汎用インターフェースを介してシステム挙動を監視できるソフトウェアトレースツール。現在、生産中止していますが、既存のインストーラーは現在も使用できます。

Simulink[®]

動的システムのモデリング、シミュレーション、分析を行うためのツール。Simulink のモデルを INTECRIO にインポートできます。

Simulink[®] Coder[™]

Simulink / Stateflow モデル用のコードジェネレータ。MATLAB[®] Coder[™] が必要です。

Stateflow[®]

複雑なイベントドリブンシステムのモデリングとシミュレーションを行うためのツール。MATLAB/Simulink にシームレスに統合されています。

Xパス実験

バイパス実験とフルパス実験を組み合わせた実験。実験ターゲット（ES900、ES800）は、バイパスフックを持つ ECU を外部インターフェースとして利用します。

アクチュエータ

実行ハードウェアユニット。電子信号処理とメカニクスとの物理的なインターフェースです。

アプリケーションモード

アプリケーションモードはオペレーティングシステムの実行モードです。EEPROM プログラミングモード、初期化モード、通常運転モードなど、システムのさまざまな状態に対応します。

イベント

オペレーティングシステムのアクション（タスクなど）を開始するための外部トリガ。

イベントインターフェース

「プロセス」を参照してください。

インプリメンテーション

物理的に定義されたモデル値から、実際に実行されるプログラム用の固定小数点値への変換規則を記述したものの。1 つの線形変換式とモデル値の範囲制限とで定義されます。

仮想プロトタイピング

自動車の電子制御ファンクションの仮想プロトタイプを作成することにより、PC 上で制御ロジックのテストを行えます。

環境システム

環境システムは、仮想プロトタイピングにおいてプラントモデルをモデリングするためのものです。ソフトウェアシステムと同様、モジュールとファンクションで構成され、作成方法もソフトウェアシステムの場合と同じです。

基本ソフトウェア

「BSW」を参照してください。

グラフィカルフレームワーク

INTECRIO のメインウィンドウ。このウィンドウでさまざまな INTECRIO コンポーネントを統合します。

クロスバー

AUTOSAR に該当しないモジュール、ファンクション、ハードウェアの間で行われる通信について、その管理と制御を行うものです。

検証

ある開発フェーズの成果が、そのフェーズの仕様を満たしているかどうかを判断するために、システムまたはコンポーネントを評価するプロセス。ソフトウェアの検証においては、特定の開発ステップについて定義された仕様がソフトウェアに正しく実装されているかを検証します。

コネクション（静的）

静的コネクションとして定義されたシグナルソースとシグナルシンク間の接続は、ランタイムにおいて変更できません。

コネクション（動的）

動的コネクションとして定義されたシグナルソースとシグナルシンク間の接続は、ビルド処理を行わなくてもランタイムにおいて変更できます。

サービスポイント

ECU ソフトウェア内のプロセスをカプセル化したもので、ターゲットシステムとのデータ転送アクションを行います。これらのアクションはユーザーが有効化したり設定したりすることができます。

サービスポイントクラスタ

ECU において同じ優先度で実行される複数のサービスポイントのグループ（同じ ECU タスク内に配置された複数のサービスポイント）

サービスポイントクラスタグループ

複数のサービスポイントクラスタのグループ。グループには、ECU 内で同時に呼び出される可能性のあるすべてのタスクのすべてのサービスポイントが含まれます。

システムプロジェクト

ハードウェアシステム、ソフトウェアシステム、環境システム（仮想プロトタイピングの場合）、シグナルのマッピング、およびオペレーティングシステムのコンフィギュレーションを、1つのプロジェクトとして一体化したものです。実行コードは、システムプロジェクト単位で生成されます。

センサ

センサは、物理的または化学的な量（通常は電氣的でない量）を電氣的な量に変換します。

ソフトウェアシステム

ソフトウェアシステムには ECU ファンクションの汎用部分（モジュール、ファンクション、接続情報）が含まれています。

タスク

タスクは複数のプロセスの実行シーケンスを定義するもので、オペレーティングシステムにより起動されます。タスクの属性には、アプリケーションモード、起動トリガ、優先度、スケジューリングモードなどがあります。タスクが起動されると、タスクに割り当てられた一連のプロセスが、指定の順序で実行されます。

妥当性確認

作成されたシステムまたはコンポーネントが、アプリケーションに対するユーザー要件を満たすものであるかどうかを判断するための評価プロセス。ここでは、ユーザーの承認がとれるシステムまたはコンポーネントが作成されたかどうかを確認します。

統合

複数の会社においてさまざまなツールを用いて開発されたモデルコードを結合して制御アルゴリズムを構築し、ターゲットハードウェア用にそのアルゴリズムを調整して、最終的に実行ファイルを作成すること。

ハードウェアシステム

ハードウェアシステムは実験に使用するハードウェア全体を定義するもので、ECU（実験ターゲット）についての定義や ECU 間インターフェース（バスシステム）の記述などで構成されます。

バイパス実験

バイパス実験においては、ECU プログラムの一部が実験ターゲット（ES900、ES800）上で実行されます。バイパス実験を行うためには、ECU プログラム内に専用のフックルーチンを挿入する必要があります。INTECRIO V5.0 は、CAN / UDP を用いた XCP バイパスや、フックドベース/サービスベースの ETK / XETK / FETK バイパスをサポートしています。

ファンクション

ソフトウェアシステムをグルーピングするためのアイテムで、ファンクション自体に機能はありません。1つのファンクション内には複数のモジュールが組み込まれ、接続されています。関連するモジュールをファンクションにまとめることにより、モデル機能を明確に把握でき、再利用もしやすくなります。

フルパス実験

フルパス実験においては、ECU プログラム全体が実験ターゲット上で実行されます。

プロジェクトインテグレータ

プロジェクトインテグレータは、システムのすべてのコンポーネント（モジュールとファンクション、ハードウェアインターフェース、OS コンフィギュレーションなど）を結合して1つの実行ファイルを生成します。

プロジェクトコンフィギュレータ

INTECRIO の統合プラットフォームの一部。ソフトウェアシステムとシステムプロジェクトを設定する際に使用されます。

プロセス

オペレーティングシステムにより起動され、並行的に実行できる機能単位。プロセスはモジュール内に記述し、引数（入力）や戻り値（出力）を持ちません。

プロセッサ

「CPU」を参照してください。

プロトタイプ

実験ターゲットシステム用の実行ファイル。プロトタイプには ECU ファンクションが実行可能なコードとして書き込まれていますが、その構成と目的は、最終的な製品用のものとは異なります。

モジュール

INTECRIO のモジュールは、ECU ファンクションが汎用的に記述されたものです。1つのモジュールは、1つの ASCET プロジェクトや Simulink モデルに相当します。

ラピッドプロトタイピング

実験ターゲット（車両インターフェースを持つコンピュータ）上でソフトウェアを実行して、制御アルゴリズムの検証と妥当性評価を行うこと。

ランタイム環境

「RTE」を参照してください。

ランナブルエンティティ

「RE」を参照してください。

レガシー AUTOSAR モジュール（またはレガシー SWC）

V5.0.0 以前の INTECRIO にインポートされた AUTOSAR モジュール

ワークスペース

ワークスペースにおいて、INTECRIO での作業で生成されたすべてのデータが管理され、結合されます。WS ブラウザ（ワークスペースブラウザ）のツリービューから、INTECRIO のすべてのアイテムを開くことができます。



☒ 4-1	ワークスペースを作成する	43
☒ 4-2	モジュールを作成する	44
☒ 4-3	モジュールをインポートする	45
☒ 4-4	ファンクションを作成する	46
☒ 4-5	SWC を含まないファンクションを定義する	47
☒ 4-6	SWC を含むファンクションを定義する	48
☒ 4-7	ソフトウェアシステム/環境システムを作成する	49
☒ 4-8	SWC を含まないソフトウェアシステム/環境システムを定義する	50
☒ 4-9	SWC を含むソフトウェアシステムを定義する	51
☒ 4-10	ハードウェアシステムを作成して設定する	52
☒ 4-11	HWX2 フォーマットのハードウェアコンフィギュレーション (*.hwx) をインポートする	53
☒ 4-12	デ이지チェーンの作成と設定	54
☒ 4-13	LIN コントローラの作成と設定	55
☒ 4-14	LIN ノードタイプの変更	56
☒ 4-15	バイパスデバイスのセットアップ	57
☒ 4-16	サービスポイントのセットアップ (サービスベースバイパス)	58
☒ 4-17	フックベースサービスポイントのセットアップ (フックベースバイパス + DISTAB17)	59
☒ 4-18	CAN コンフィギュレーションファイルのインポート	60
☒ 4-19	CAN データベースのエクスポート	61
☒ 4-20	システムプロジェクトを作成する	62
☒ 4-21	ハードウェアとソフトウェアを接続する	
☒ 4-22	オペレーティングシステムを自動設定する	64
☒ 4-23	オペレーティングシステムを手動設定する	65
☒ 4-24	タスクを手動設定する	66
☒ 4-25	ISR を手動設定する	67
☒ 4-26	実行ファイル (プロトタイプ) を生成する	68
☒ 4-27	ETAS Experiment Environment (ETAS 実験環境) を開く	69
☒ 4-28	ETAS Experiment Environment をセットアップする	70
☒ 4-29	測定/適合ウィンドウを作成する	71
☒ 4-30	測定/適合インストゥルメントをセットアップする	72
☒ 4-31	測定/適合変数を測定/適合インストゥルメントに割り当てる	73
☒ 4-32	レイヤを使用する	74
☒ 4-33	実験を管理する	75
☒ 4-34	データ収集について設定する	76
☒ 4-35	実験を開始する	
☒ 4-36	測定データを記録する	
☒ 4-37	値を測定する	79
☒ 4-38	エディタで適合を行う	80

☒ 4-39	仮想プロトタイピング用プロジェクトを作成する	82
☒ 4-40	仮想プロトタイピング実験	83
☒ 4-41	ワークスペースを管理する	84
☒ 4-42	ワークスペースをインポート/エクスポートする	85
☒ 4-43	外部で変更されたワークスペースを開く	86
☒ 4-44	バックアップファイルを復元する	86
☒ 4-45	グラフィックエディタ上のブロックのレイアウトを編集する	87
☒ 4-46	グラフィックエディタ上のブロックのポートを追加/削除する	88
☒ 4-47	システムプロジェクトの自動ドキュメント生成	89
☒ 5-1	モジュールとポートの接続	108

索引

A		OS のマニュアル設定 65
AUTOSAR		P
SWC を含むソフトウェアシステム . 51		PC 接続機能
SWC を含むファンクション 48		特殊機能 25
C		ループバックネットワークアドレス 22
CAN		R
コンフィギュレーションファイルのイ ンポート 60		RTA-OSEK for PC
データベースのエクスポート 61		インストール 25
D		RTA-OSEK ツール
DISTAB17		インストール 25
フックベースバイパス 59		S
E		Simulink モデル
ES4xx 54		コードの生成 96
ES63x 54		ソリューションアルゴリズム 97
ES930 143		ターゲット 98
ETAS Experiment Environment		V
associate with INCA 31		Virtual Prototyping
アンインストール 39		→ 「仮想プロトタイピング」参照
インストール 28		Z
免責条項 30		PC 接続機能
ライセンス契約 30		ループバックネットワークアドレス 22
ETAS ネットワークマネージャ 158		あ
Experiment Environment		アンインストール
→ 「ETAS Experiment Environment」 参照		ETAS Experiment Environment .. 39
H		INTECRIO 39
HWX インポート 53		ターゲットサーバー 40
I		安全に関する注意事項 7
INTECRIO		本製品に関する特殊な要件 9
アンインストール 39		い
インストール 17		インストール
コマンドラインインストール .. 25, 33		ETAS Experiment Environment .. 28
MATLAB/Simulink との関連付け (自 動) 23		INTECRIO 17
インストール開始 17		PC 接続機能 25
機能範囲指定 20		RTA-OSEK for PC 25
コンポーネント選択 20		RTA-OSEK ツール 25
パス設定 21		安全に関する注意事項 19
L		開始 17
LIN		仮想プロトタイピングパッケージ .. 25
コンフィギュレーションのインポート 55		旧バージョンのアンインストール .. 18
ノードタイプ変更 56		コマンドライン 25, 33
O		コンポーネント選択 20
OSC		サイレント 26, 33
OS の自動設定 64		システム要件 17
OS のマニュアル設定 65		パス設定 21
OS の自動設定 64		ライセンスに関する合意 19
		ループバックネットワークアドレス 22
		インポート
		AUTOSAR CAN コンフィギュレーショ ン (*.arxml) 60

CAN データベース (*.dbc)	60	と	
ES4xx/ES63x コンフィギュレーション	54	問い合わせ先	164
LIN コンフィギュレーション (*.ldf)	55	トラブルシューティング	159 ~ 163
デジチェーンコンフィギュレーション	54	ETAS ネットワーク用のネットワークアダプタを選択できない ..	159
ワークスペース	85	PC に接続されたイーサネットハードウェアが検出できない ..	160
え		ね	
エクスポート		ネットワークアダプタ	158
CAN データベース (*.dbc)	61	は	
ワークスペース	85	バーチャルプロトタイピング	
か		→「仮想プロトタイピング」参照	
仮想プロトタイピング		ハードウェアコンフィギュレータ	
実験	83	CAN	60, 61
特殊機能	25	ES4xx	54
プロジェクトの作成	82	ES63x	54
こ		LIN	55
コードの生成 (BMT)		デジチェーン	54
Simulink モデル	96	ハードウェアシステムのインポート (*.hwx)	53
個人情報保護	9	バイパス	57
コマンドラインインストール (INTECRIO)	33	ハードウェアシステム	
サイレント	26, 33	インポート (*.hwx)	53
さ		バイパス	
サービスベースバイパス	58	サービスポイントのセットアップ ..	58
サイレントインストール (INTECRIO) ..	26, 33	設定	57
し		フックベースサービスポイントのセットアップ	59
実験環境		ふ	
→「ETAS Experiment Environment」参照		フックベースバイパス	
そ		DISTAB17	59
ソフトウェアシステム		よ	
SWC を含む~	51	用語集	165 ~ 173
ち		用語	169 ~ 173
チュートリアル	90 ~ 157	略語	165 ~ 169
ES930	143	わ	
OS コンフィギュレーション	119	ワークスペース	
環境システム	108	インポート	85
コード生成	118	エクスポート	85
システムプロジェクト	112	バックアップの復元	86
実験	126 ~ 141	変更されたワークスペースを開く ..	86
ソフトウェアシステム	108		
ハードウェアシステム	111		
ビルド	123		
ファンクション	104		
モジュール	102		
モジュールの準備	96		
て			
デジチェーン	54, 143		