

dSPACE Release

新機能と移行手順

Release 2014-B – 2014 年 11 月

dSPACE へのお問い合わせ

dSPACE Japan 株式会社

Web: <http://www.dspace.jp>
テクニカルサポート: support@dspace.jp

dSPACE サポートへのお問い合わせ

問題点やご質問を dSPACE にお問い合わせいただく場合に、
http://www.dspace.com/go/jpn_supportrequest のお問い合わせフォームにサポートのお申し込みをご入力ください。

お問い合わせフォームは、サポートチームがより迅速かつ効果的に問題点を取り扱うのに役立ちます。

ソフトウェアのアップデートとパッチ

既存の dSPACE インストールに対して、最新のパッチをダウンロードしてインストールすることを強くお勧めします。ソフトウェアのアップデートとパッチについては、以下のサイトをご覧ください。 http://www.dspace.jp/goto.cfm/ja_0903

重要なお知らせ

本書には、著作権法により保護された情報が含まれています。すべての権利は留保されています。本書は、すべての商標表示をすべての印刷コピーに保持するという条件で、個人または内部での使用を目的として印刷することができます。それ以外のすべての場合において、dSPACE GmbH の書面による事前の許可なく、本書のすべてもしくは一部を、コピー、複製、翻訳、または電子的媒体もしくは機械可読形式に変換することを禁じます。

© 2000 - 2014
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

本出版物と内容は、予告なしで変更されることがあります。

CalDesk、ConfigurationDesk、ControlDesk、MicroAutoBox、SCALEXIO、SYNECT、SystemDesk、TargetLink、および VEOS は、米国、その他の国、またはその両方における dSPACE GmbH の登録商標です。その他のブランド名または製品名は、その企業または組織の商標または登録商標です。

目次

本書について	11
本書で使用する記号と表記規則.....	11
オンラインヘルプおよび PDF ファイルの利用.....	12
dSPACE Release 2014-B の概要	15
一般的な機能拡張および変更.....	15
64 ビットバージョンの RCP and HIL Software.....	20
製品バージョンの概要.....	21
各製品の主な新機能.....	25
以前のリリースからの移行について	33
dSPACE Release 2014-B への移行.....	33
Python 2.7 ディストリビューションの変更点	35
Python 2.7 の主な変更点.....	36
dSPACE Python ディストリビューションの主な変更点.....	36
Python インストールの使用に関する一般情報.....	37
標準の Python 2.7 ディストリビューションの機能拡張.....	37
AutomationDesk	39
dSPACE Release 2014-B での AutomationDesk 4.1 の使用....	39
Automotive Simulation Model (ASM)	43
ASM Base InCylinder Blockset.....	44
ASM Base InCylinder Blockset 1.9.1 への移行.....	44
ASM Diesel Engine Blockset.....	45
ASM Diesel Engine Blockset 2.0 の新機能.....	45
ASM Diesel Engine デモモデルの変更点.....	46
ASM Diesel Engine Blockset 2.0 への移行.....	47
ASM Diesel Exhaust Blockset.....	50
ASM Diesel Exhaust Blockset 2.0.....	50
ASM Diesel InCylinder Blockset.....	51
ASM Diesel InCylinder デモモデルの変更.....	51

ASM Drivetrain Basic Blockset.....	52
ASM Drivetrain Basic Blockset 4.0 の新機能.....	52
ASM Drivetrain Basic Blockset 4.0 への移行.....	52
ASM Electric Components Blockset.....	54
ASM Electric Components Blockset 2.7 の新機能.....	54
ASM Electric Components デモモデルの変更.....	54
ASM Engine Gasoline Basic Blockset.....	55
ASM Engine Gasoline Basic Blockset 2.0 の新機能.....	55
ASM Engine Gasoline Basic デモモデルの変更.....	56
ASM Engine Gasoline Basic Blockset 2.0 への移行.....	56
ASM Engine Gasoline Blockset.....	59
ASM Engine Gasoline Blockset 3.0 の新機能.....	59
ASM Engine Gasoline デモモデルの変更.....	60
ASM Engine Gasoline Blockset 3.0 への移行.....	61
ASM Gasoline InCylinder Blockset.....	64
ASM Gasoline InCylinder デモモデルの変更.....	64
ASM Gasoline InCylinder Blockset 1.9 への移行.....	64
ASM Parameterization Tool.....	65
ASM Parameterization Tool 1.6.5 の新機能.....	65
ASM Parameterization Tool 1.6.5 への移行.....	65
ASM Pneumatics Blockset.....	67
ASM Pneumatics Blockset 2.0 の新機能.....	67
ASM Pneumatics Blockset 2.0 への移行.....	67
ASM Traffic Blockset.....	68
ASM Traffic Blockset 3.1 の新機能.....	68
ASM Traffic Blockset 3.1 への移行.....	68
ASM Turbocharger Blockset.....	69
ASM Turbocharger Blockset 3.0 の新機能.....	69
ASM Vehicle Dynamics Blockset.....	70
ASM Vehicle Dynamics Blockset 3.0 の新機能.....	70
ASM Vehicle Dynamics デモモデルの変更.....	72
ASM Vehicle Dynamics Blockset 3.0 への移行.....	72
ConfigurationDesk	75
ConfigurationDesk – Implementation.....	76
ConfigurationDesk 5.2 (Implementation Version) の新機能...	76
ConfigurationDesk 5.2 への移行.....	81

ControlDesk Next Generation	85
ControlDesk Next Generation の新機能 (ControlDesk 5.3)	86
新しい一般機能 (ControlDesk 5.3)	86
プロジェクトおよび実験の新機能 (ControlDesk 5.3)	87
プラットフォーム管理およびプラットフォーム/デバイスの新 機能 (ControlDesk 5.3)	87
変数管理の新機能 (ControlDesk 5.3)	88
新しいビジュアル表示および計器機能 (ControlDesk 5.3)	89
新しい計測機能および記録機能 (ControlDesk 5.3)	91
Bus Navigator の新機能 (ControlDesk 5.3)	92
新しいデータセット管理機能 (ControlDesk 5.3)	92
ECU 診断の新機能 (ControlDesk 5.3)	93
Signal Editor の新機能 (ControlDesk 5.3)	93
新しい自動化機能 (ControlDesk 5.3)	94
ControlDesk Next Generation への移行 (ControlDesk 5.3)	95
ControlDesk Next Generation への移行 (ControlDesk 5.3)	95
dSPACE HIL API .NET	99
dSPACE HIL API .NET 1.7 の新機能	99
dSPACE Python Extensions	101
dSPACE Python Extensions 1.7 の新機能	101
dSPACE XIL API	103
dSPACE XIL API 2.0 の新機能	103
dSPACE XIL API 2.0 への移行	103
ECU Interface Manager	105
ECU Interface Manager 1.5 の新機能	105
ECU Interface Manager 1.5 への移行	106
Firmware Manager	107
Firmware Manager 1.2 の機能	107
dSPACE FlexRay Configuration Package	109
dSPACE FlexRay Configuration Package 3.3 の新機能	109

Model Compare	111
Model Compare 2.5 の新機能.....	111
Model Compare 2.5 への移行.....	112
ModelDesk	115
ModelDesk 4.0 の新機能.....	115
MotionDesk	117
MotionDesk 3.5 の新機能.....	117
MotionDesk 3.5 への移行.....	118
Real-Time Testing	119
Real-Time Testing 2.4 の新機能.....	119
RTI/RTI-MP および RTLib	121
RTI/RTI-MP および RTLib の新機能.....	121
RTI/RTI-MP および RTLib の移行上の注意点.....	125
RTI Bypass Blockset	129
RTI Bypass Blockset 3.3 の新機能.....	129
RTI Bypass Blockset 3.3 への移行.....	130
RTI CAN Blockset	133
RTI CAN Blockset 3.3 の新機能.....	133
RTI CAN MultiMessage Blockset	135
RTI CAN MultiMessage Blockset 4.0 の新機能.....	135
RTI CAN MultiMessage Blockset 4.0 への移行.....	136
RTI Electric Motor Control Blockset	139
RTI Electric Motor Control Blockset 1.0 の機能.....	139
RTI Ethernet Blockset	141
RTI Ethernet Blockset 1.1 の新機能.....	141

RTI FPGA Programming Blockset	143
RTI FPGA Programming Blockset 2.8 の新機能.....	143
RTI FPGA Programming Blockset 2.8 への移行.....	145
RTI LIN MultiMessage Blockset	147
RTI LIN MultiMessage Blockset 2.4 の新機能.....	147
RTI LIN MultiMessage Blockset 2.4 への移行.....	147
RTI USB Flight Recorder Blockset	149
RTI USB Flight Recorder Blockset 1.2 の新機能.....	149
SCALEXIO Firmware	151
SCALEXIO Firmware 3.1 の新機能.....	151
SystemDesk	153
SystemDesk 4.3 の新機能.....	154
新しい一般機能.....	154
ソフトウェアアーキテクチャのモデリング.....	155
システムのモデリング.....	156
ECU コンフィギュレーション.....	156
エレメントの妥当性確認.....	157
SystemDesk 4.3 への移行.....	158
SystemDesk 4.3 への移行.....	158
TargetLink	159
TargetLink 4.0 および TargetLink Data Dictionary 4.0 の新機能....	160
Simulink または Stateflow でのモデリング.....	160
行列信号のサポート.....	161
新しくサポートされる Simulink ブロック.....	161
バスサポートの向上.....	162
動的ルックアップテーブル.....	162
TargetLink のシミュレーションフレームの改善.....	163
Scaling-Invariant システムの改良.....	163
ブロックのプロパティの追加サポート.....	164
ファンクションサブシステムのシグネチャの一元的な指定.....	165

コード生成のコア機能.....	167
MISRA-C への準拠.....	167
コード効率性の向上.....	168
Data Dictionary とデータ管理.....	169
Data Dictionary の改善点.....	170
新しい DD MATLAB API コマンド.....	172
AUTOSAR.....	175
サポートされている AUTOSAR リリース.....	175
新しい AUTOSAR 機能.....	176
テストのサポート.....	177
オンラインでのパラメータ変更の改善.....	178
ターゲットシミュレーションモジュールの変更.....	178
Code Generator オプション.....	180
新しい Code Generator オプション.....	180
ツールチェーンの統合.....	182
Windows 適合性の向上.....	182
その他.....	183
一般的な機能拡張および変更.....	183
API コマンド.....	185
新しい API 関数.....	185
TargetLink 4.0 および TargetLink Data Dictionary 4.0 への移行...	187
Data Dictionary とデータ管理.....	187
TargetLink 4.0 への移行.....	188
インクルード DD ファイルのある Data Dictionary をアップグレードする方法.....	189
コードの変更.....	192
コードの変更.....	192
Code Generator オプション.....	204
Code Generator オプションに関する移行上の注意点.....	204
アクセス関数の変更.....	205
マクロアクセス関数用のデフォルトのマクロ本体.....	205
ADDRESS_BY_PARAMETER アクセス関数の変更.....	206
新しいアクセス関数固有の名前マクロ.....	207
AUTOSAR に関する移行上の注意点.....	207
2 次元行列データエレメントとオペレーション引数.....	208
AUTOSAR エクスポート.....	208
Container Manager の個別インストール.....	209

アプリケーションデータタイプ.....	210
配列および行列タイプの制約.....	210
置き換えられた IsQueued プロパティ.....	211
その他.....	211
移行に関するその他の注意点.....	212
バス診断のより厳密な設定.....	213
プロットチャンネルの指定.....	215
信号プロパティの継承.....	216
API コマンド.....	216
TargetLink と TargetLink Data Dictionary API 関数の変更.....	216
廃止事項.....	217
廃止された TargetLink の機能.....	217
廃止された Data Dictionary 機能.....	217
廃止された制限事項.....	218
廃止された API 関数.....	218
メッセージ.....	219
メッセージの変更.....	219
Stateflow 関連の変更.....	220
エクスポートされたグラフィカル関数.....	220
Stateflow の行列.....	221
TargetLink の今後のバージョンでの変更予定.....	222
廃止予定.....	222
VEOS	223
VEOS 3.3 の新機能.....	223
VEOS 3.3 への移行.....	224
互換性情報	225
サポートしている MATLAB リリース.....	226
オペレーティングシステム.....	227
dSPACE ソフトウェアのランタイム互換性.....	230
Windows (64 ビット版) で dSPACE ソフトウェア (32 ビットバージョン) を使用する場合の制限事項.....	231
64 ビット dSPACE DVD に含まれる製品についての制限事項.....	231
Windows 7 の場合の制限事項.....	235
索引	237

本書について

目次 本書では、Release 2014-B に含まれるすべての dSPACE ソフトウェア製品の新機能について説明します。以前の dSPACE リリースからの変更がない、または変更が少ないソフトウェア製品についても概要を示します。また、以前の dSPACE リリース、特に以前の製品バージョンからの移行手順についても、必要に応じて説明します。

項目の一覧




本章の内容

本書で使用する記号と表記規則	11
オンラインヘルプおよび PDF ファイルの利用	12

本書で使用する記号と表記規則

記号

本書では次の記号を使用します。

	人身傷害につながる一般的な危険があることを示します。本書の指示に従って危険を回避しないと、けがをする可能性があります。
	感電の危険があることを示します。本書の指示に従って危険を防止しないと、死亡または重傷を負う可能性があります。
	物的な損害の危険があることを示します。本書の指示に従って危険を防止しないと、物的損傷を招く可能性があります。



注意すべき重要な情報(故障を回避するための注意など)を示します。




作業を円滑に進めるのに役立つヒントを示します。


表記規則

本書では次の省略表記と書式を使用します。

%name% パーセント記号で囲まれた名前は、ファイルとパス名の環境変数を表します。

<> 山形括弧で囲まれた表記は、任意のファイル名やパス名などを表すワイルドカード文字またはプレースホルダを示します。

 リンク先が別のドキュメントを参照する場合にドキュメントタイトルの前に付記されます。

 リンク先が dSPACE HelpDesk で提供されている別のドキュメントを参照していることを示します。

特別なフォルダ

ControlDesk Next Generation や AutomationDesk などの一部のソフトウェア製品では、次の特別なフォルダを使用します。

共通プログラムデータフォルダ アプリケーション固有の設定データ用の標準フォルダで、すべてのユーザが使用します。

```
%PROGRAMDATA%\dSPACE\\
```

ドキュメントフォルダ ドキュメント用の標準フォルダで、各ユーザ固有のフォルダです。

```
%USERPROFILE%\My Documents\dSPACE\\  
<VersionNumber>
```

ローカルプログラムデータフォルダ アプリケーション固有の設定データ用の標準フォルダで、現在の非ローミングユーザが使用します。

```
%USERPROFILE%\AppData\Local\dSPACE\\  
<ProductName>
```

オンラインヘルプおよび PDF ファイルの利用

目的

dSPACE ソフトウェアをインストールすると、インストールした製品に関するドキュメントがオンラインヘルプまたは Adobe® PDF ファイルとして参照できるようになります。

オンラインヘルプ

オンラインヘルプ (dSPACE HelpDesk) を使用するには

Windows の [スタート]メニュー [スタート]メニューから[(すべての)プログラム] - [<製品名>] - [dSPACE HelpDesk (<製品名>)]を選択して、選択した製品のスタートページから dSPACE HelpDesk を開きます。また、インストールされている他のソフトウェア製品およびそれにサポートされるハードウェアのユーザマニュアルに移動して検索することもできます。

状況依存ヘルプ 現在アクティブなコンテキストのヘルプを表示するには、F1 キーを押すか、または dSPACE ソフトウェアの [Help] ボタンをクリックします。



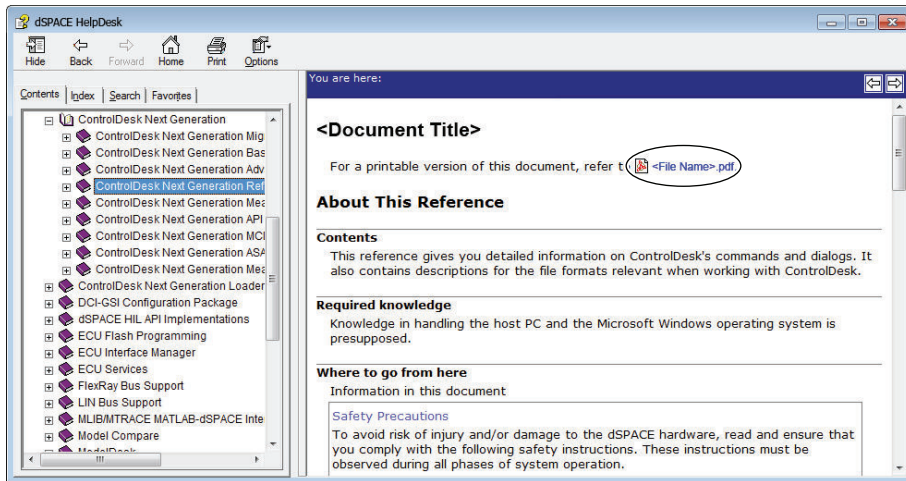
いくつかのソフトウェア製品では、文脈依存ヘルプは入手可能ではない。

dSPACE ソフトウェアの [Help]メニュー メニューバーから [Help] - [Contents] または [Help] - [Search] (すべてのソフトウェア製品で利用可能とはかぎりません) を選択して dSPACE HelpDesk を開きます。現在アクティブな製品のスタートページが表示されます。また、インストールされている他のソフトウェア製品およびそれにサポートされるハードウェアのユーザマニュアルに移動して検索することもできます。

PDF ファイル

PDF ファイルは、次の方法で利用することができます。

dSPACE HelpDesk ドキュメント名の先頭にある PDF リンクをクリックします。



dSPACE Release 2014-B の概要

目的 Release 2014-B の主な新機能について説明します。また、変更のない製品に関する情報についても紹介します。

項目の一覧 本章の内容

一般的な機能拡張および変更	15
64 ビットバージョンの RCP and HIL Software	20
製品バージョンの概要	21
各製品の主な新機能	25

一般的な機能拡張および変更

目的 複数の dSPACE 製品に関係する新機能と変更を下記に示します。

新しい dSPACE ハードウェアのサポート dSPACE Release 2014-B では、以下の新しい dSPACE ハードウェアが導入されています。

- MicroLabBox
ラボ用の新しいプロトタイピングハードウェアです。詳細については、「RTI/RTI-MP および RTLib の新機能」(121 ページ)を参照してください。
- DS4342 CAN FD Interface Module
CAN FD パスプロトコルをサポートする DS4504 Interface Board 用の新しいピギーバックモジュールです。

環境変数の削除

dSPACE RCP and HIL 2014-A から、このリリースのインストール先のパスを参照するグローバル環境変数が、インストール時に設定されなくなりました。dSPACE 製品は、これらの環境設定に関係なく動作します。

変数`%DSPSPACE_ROOT%`または`%DSPSPACE_CONFIG%`を読み取るカスタムコードを移行する必要があります。

一部のコマンドラインツール(ハンドコードされたリアルタイムアプリケーションをビルドするための `Down` ツールなど)を利用しやすくするために、Windows の[スタート]メニューに[*Command Prompt for dSPACE RCP and HIL <バージョン>*]という名前のコマンドプロンプトへのショートカットが含まれています。環境設定と検索パスは自動的に設定されます。

新しい Python のサポート

dSPACE Release 2013-B から、自動化インターフェースなどに Python を使用している dSPACE ソフトウェア製品は、Python 2.7 をサポートします。dSPACE Release 2014-B では、小規模な変更がいくつか行われました。詳細および必要な移行手順については、「Python 2.7 ディストリビューションの変更点」(35 ページ)を参照してください。

RCP and HIL Software の MATLAB のサポート(64 ビット)

次の RCP and HIL Software 製品が、MATLAB® 64 ビットバージョンをサポートする製品バージョンでも使用できるようになりました。

- RTI Bypass Blockset

使用可能な製品の詳細については、「64 ビットバージョンの RCP and HIL Software」(20 ページ)を参照してください。

32 ビットおよび 64 ビットソフトウェアの提供

dSPACE ソフトウェアは 2 つの DVD セットで提供されます。各 DVD の内容は同じですが、次の点が異なります。

- 2 枚の 32 ビット DVD には、MATLAB の 32 ビットバージョンをサポートするなどの dSPACE ソフトウェア製品の 32 ビットバージョンのみが含まれます。
- 2 枚の 64 ビット DVD には、次の内容が含まれます。
 - MATLAB の 64 ビットバージョンをサポートするよう移植された、MATLAB を使用するすべての dSPACE 製品
 - MATLAB の 64 ビットバージョンをサポートするすべての 32 ビットバージョンの dSPACE 製品
 - MATLAB に関連しない 32 ビットバージョンの dSPACE 製品すべて (ControlDesk Next Generation など)



したがって、インストール時には、64 ビット DVD セットを 32 ビット DVD セットに取り替えなくても dSPACE ソフトウェアをインストールすることができます。

dSPACE Release 2014-B の 64 ビットバージョンに含まれる dSPACE 製品の全一覧については、「64 ビット dSPACE DVD に含まれる製品についての制限事項」(231 ページ)を参照してください。

DVD セットの内容

dSPACE Release 2014-B から、dSPACE ソフトウェアは各 DVD セット (32 ビット DVD セットおよび 64 ビット DVD セット) の 2 枚の DVD で提供されます。DVD には、以下の dSPACE ソフトウェアパッケージとメインの製品が収録されています。

■ ディスク 1:

- AutomationDesk 4.1
- ControlDesk Next Generation (ControlDesk 5.3)
- TargetLink 4.0
- Model Compare 2.3



この製品は米国での使用が禁止されています

米国では Model Compare を使用することはできません。この製品を米国内で使用することも第三者に使用させることも米国の法律に違反します。

- SystemDesk 4.3 (AUTOSAR 4.x をサポート)
- VEOS 3.3
- dSPACE ソフトウェアのその他各種ツール

■ ディスク 2:

- RCP and HIL Software

「RCP and HIL Software」は、RTI、ConfigurationDesk、MotionDesk、ModelDesk などのさまざまな dSPACE ソフトウェア製品が含まれるソフトウェアパッケージを指す総称です。



ディスク 2 には、その他の dSPACE ソフトウェア製品は収録されていません。

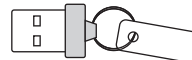
ドングルライセンスの新しいハードウェアドングル

dSPACE Release 2014-B では、ドングルライセンスのハードウェアドングルが WibuKey から CodeMeter に変更されています。両方とも WIBU-SYSTEMS 社の製品であり、外観は下図のとおりです。

WibuKeyドングル



CodeMeterドングル



新しい CodeMeter ハードウェアドングルは、dSPACE Release 2014-B で初めてご購入いただいた新規のお客様にご利用いただけます。

次の互換性情報にご注意ください。

- 通常、既存の WibuKey ドングルで dSPACE Release 2014-B をご利用いただけます。dSPACE Release 2014-B では、両バージョンのドングルドライバがホスト PC にインストールされます。ドライバソフトウェアがご使用のドングルを自動的に検出します。他の作業は必要ありません。
- 新しい CodeMeter ドングルで dSPACE Release 2014-A 以前のバージョンを使用する場合は、ご使用のホスト PC に dSPACE Installation Manager 3.8 をインストールする必要があります。このバージョンには、新しいドングルのドライバが含まれています。dSPACE Installation Manager は、
http://www.dspace.com/ja/jpn/home/support/patches/updtools/im_px.cfm からダウンロードすることができます。
- dSPACE Release 6.3 以前のバージョンでは、新しい CodeMeter ドングルのテストは行っていません。必要に応じて、dSPACE サポートにご連絡ください。

dSPACE HelpDesk 使用時の 制限事項

dSPACE HelpDesk は、C:\Program Files\Common Files\dSPACE(32 ビット版オペレーティングシステム)または C:\Program Files(x86)\Common Files\dSPACE(64 ビット版オペレーティングシステム)のリリースごとのフォルダにインストールされます。たとえば、dSPACE Release 2014-B の製品と dSPACE Release 2014-A の製品をインストールしている場合は、2 つの dSPACE HelpDesk を使用することができます。

以下の制限事項に注意してください。

ドキュメントへのリンクが機能せず、エラーメッセージ「*Selection is not associated with any topics.*」が返される場合があります。これは、次のいずれかの理由が考えられます。

- 製品がライセンスキーに含まれていないため、製品のドキュメントがインストールされていない。
- 製品のドキュメントが別の dSPACE HelpDesk にインストールされている。たとえば、現在の dSPACE Release にある製品が変更されていない場合、そのユーザマニュアルは製品セットアップが作成されたバージョンの dSPACE HelpDesk にインストールされています。

dSPACE Release 2014-B をインストールした場合、以下の製品のユーザマニュアルは dSPACE HelpDesk 201A に格納されます。

- AutomationDesk 4.1

製品のユーザマニュアルの場所が不明な場合は、Windows の[スタート]メニューから製品固有の[dSPACE HelpDesk]ショートカットを使用してオンラインヘルプを開いてください。

印刷版のユーザマニュアル

dSPACE Release 2014-B には、印刷版のユーザマニュアルは付属していません。必要な印刷版のマニュアルをユーザが指定できるようになっています。印刷版のユーザマニュアルについては、http://www.dspace.com/go/request_jp_documentation を参照してください。



印刷版のマニュアルを注文しない場合は、ご使用の製品に関する新機能、拡張機能、安全上の注意事項などの情報については、オンラインヘルプまたは PDF ファイルをご使用ください。

ソフトウェアサポートの廃止

- dSPACE Release 2014-B では、オペレーティングシステム Microsoft Windows XP がサポートされなくなりました。dSPACE ソフトウェアのインストールはブロックされます。
- dSPACE Release 2014-A までは以下の製品が提供されますが、dSPACE Release 2014-B からは提供されなくなります。
 - RTI AUTOSAR Package
dSPACE では現在、MicroAutoBox II 上の V-ECU のラピッドコントロールプロトタイピング (RCP) 用に RTI AUTOSAR Blockset 2.0 を提供しています。
ただし、RTI AUTOSAR Blockset 2.0 は、RTI AUTOSAR Package の直接の後継ではありません。そのため、RTI AUTOSAR Package で作成されたモデルは、RTI AUTOSAR Blockset 2.0 で使用するために移行することができません。RTI AUTOSAR Package は、Simulink モデルに単一の AUTOSAR ソフトウェアコンポーネントを統合するために開発されました。一方、RTI AUTOSAR Blockset 2.0 では V-ECU インプリメンテーションをモデルに統合することができます。アトミックソフトウェアコンポーネントに加え、V-ECU インプリメンテーションにはオペレーティングシステムの設定など ECU のベーシックソフトウェアの一部も含まれません。

RTI AUTOSAR Blockset 2.0 は dSPACE Release 2014-B には同梱されていません。ブロックセットを注文する場合は、dSPACE 営業部にお問い合わせください。

- dSPACE HIL API .NET
 - ASAM AE HIL API の.NET インプリメンテーションは、MAPort インプリメンテーションのみに縮小されました。EESPort インプリメンテーションは廃止されました。代わりに、新しい dSPACE XIL API の EESPort インプリメンテーションを使用することができます。詳細については、「dSPACE HIL API .NET 1.7 の新機能」(99 ページ)および「dSPACE XIL API 2.0 の新機能」(103 ページ)を参照してください。

64 ビットバージョンの RCP and HIL Software

目的	RCP and HIL Software 製品の大多数が、MATLAB 64 ビットバージョンをサポートするようになりました。
RCP and HIL(64 ビット) Software での製品サポート	<p>RCP and HIL(64 ビット)Software には通常、dSPACE Release 2014-B (32 ビット)DVD で入手可能な RCP and HIL Software と同じ製品が含まれています。ただし、MATLAB を使用する一部の RCP and HIL Software 製品は、現時点では MATLAB x64 に対応していません。したがって、これらの製品の一部は、RCP and HIL Software (64 ビット)に含まれていません。</p> <p>RCP and HIL およびその他の dSPACE ソフトウェア製品での 64 ビット MATLAB のサポートについては、「64 ビット dSPACE DVD に含まれる製品についての制限事項」(231 ページ)を参照してください。</p>
サポートされる MATLAB バージョン	<p>RCP and HIL(64 ビット)Software は以下をサポートします。</p> <ul style="list-style-type: none"> ■ MATLAB R2013a(64 ビット) ■ MATLAB R2013b(64 ビット) ■ MATLAB R2014a(64 ビット) ■ MATLAB R2014b(64 ビット) <p>「サポートしている MATLAB リリース」(226 ページ)も参照してください。</p>
サポートされている MEX コンパイラ	RCP and HIL(64 ビット)Software は、MEX 関数のビルドについては Microsoft Windows SDK 7.1 のみサポートします。

このコンパイラは、Microsoft 社 Web サイトから無料でダウンロードすることができます。このコンパイラを使用するには、.NET framework 4.0 が必要です。こちら Microsoft 社から無料で提供されています。コンパイラとフレームワークのダウンロード、またはその他の詳細については、<http://www.mathworks.com/support/compilers/R2012b/win64.html> を参照してください。

MEX コンパイラが必要な次のような RCP and HIL Software 製品を使用する場合は、このコンパイラをインストールして MEX コンパイラとして設定します。

- RTI CAN MultiMessage Blockset
- RTI LIN MultiMessage Blockset
- Automotive Simulation Model

システム要件

RCP and HIL (64 ビット) Software には、Windows 7 Enterprise Service Pack 1 (64 ビット版) が必要です。その他の 64 ビット版のオペレーティングシステム (Windows XP および Windows Vista) はサポートされません。

ホスト PC のメインメモリは、4 GB RAM 以上である必要があります。8 GB RAM 以上をお勧めします。

「オペレーティングシステム」(227 ページ)も参照してください。

製品バージョンの概要

目的

次の表に、各製品の最新のリリースおよび過去 3 回のリリースのバージョン履歴を示します。新機能が追加されている場合は、本書での参照先を示しています。

製品名	dSPACE Release			
	2013-A	2013-B	2014-A	2014-B
AutomationDesk	3.6p2	4.0	4.1	4.1 ¹⁾
Automotive Simulation Model	5.0	5.1	6.0	7.0 「Automotive Simulation Model (ASM)」(43 ページ)を参照してください。
ConfigurationDesk	4.4	5.0	5.1	5.2 「ConfigurationDesk」(75 ページ)を参照してください。
Container Manager	3.2	4.1	4.2	4.3

製品名	dSPACE Release			
	2013-A	2013-B	2014-A	2014-B
ControlDesk 3.x ²⁾	3.7.5	–	–	-
ControlDesk Next Generation ³⁾	5.0	5.1	5.2	5.3 「ControlDesk Next Generation」(85 ページ)を参照してください。
DCI Configuration Tool	3.0	3.1	3.2.2	3.3
dSPACE CAN API	2.6	2.7	2.7.1	2.7.1
dSPACE ECU Flash Programming Tool	2.2.3	2.2.4	2.2.5	2.2.5
dSPACE FlexRay Configuration Package	3.1	3.2	3.3	3.4 「dSPACE FlexRay Configuration Package」(109 ページ)を参照してください。
dSPACE HIL API .NET	1.4	1.5	1.6	1.7 「dSPACE HIL API .NET」(99 ページ)を参照してください。
dSPACE Python Extensions	1.4	1.5	1.6	1.7 「dSPACE Python Extensions」(101 ページ)を参照してください。
新製品 : dSPACE XIL API	–	–	–	2.0 「dSPACE XIL API」(103 ページ)を参照してください。
ECU Interface Manager	1.3	1.4	1.4.1	1.5 「ECU Interface Manager」(105 ページ)を参照してください。
Firmware Manager	–	1.0	1.1	1.2 「Firmware Manager」(107 ページ)を参照してください。
Model Compare	2.3	2.4	2.4	2.5 「Model Compare」(111 ページ)を参照してください。
ModelDesk	3.0	3.1	3.2	4.0 「ModelDesk」(115 ページ)を参照してください。
MotionDesk	3.2	3.3	3.4	3.5 「MotionDesk」(117 ページ)を参照してください。

製品名	dSPACE Release			
	2013-A	2013-B	2014-A	2014-B
MotionDesk Blockset	2.2.2	2.2.3	2.3	2.3.1 「MotionDesk」(117 ページ)を参照してください。
Real-Time Testing	2.1	2.2	2.3	2.4 「Real-Time Testing」(119 ページ)を参照してください。
RTI ⁴⁾	7.0	7.1	7.2	7.3 「RTI/RTI-MP および RTLib」(121 ページ)を参照してください。
RTI-MP ⁵⁾	7.0	7.1	7.2	7.3 「RTI/RTI-MP および RTLib」(121 ページ)を参照してください。
RTI AUTOSAR Package	1.3.1	1.3.1	1.3.1	-
RTI Bypass Blockset	3.0	3.1	3.2	3.3 「RTI Bypass Blockset」(129 ページ)を参照してください。
RTI CAN Blockset	3.0	3.1	3.2	3.3 「RTI CAN Blockset」(133 ページ)を参照してください。
RTI CAN MultiMessage Blockset	2.8	2.9	3.0	4.0 「RTI CAN MultiMessage Blockset」(135 ページ)を参照してください。
NEW: RTI Electric Motor Control Blockset	-	-	-	1.0 「RTI Electric Motor Control Blockset」(139 ページ)を参照してください。
RTI Ethernet Blockset	-	-	1.0	1.1 「RTI Ethernet Blockset」(141 ページ)を参照してください。
RTI Ethernet (UDP) Blockset	1.2.1	1.3	1.3	1.3
RTI FPGA Programming Blockset	2.5	2.6	2.7	2.8 「RTI FPGA Programming Blockset」(143 ページ)を参照してください。
RTI LIN MultiMessage Blockset	2.1	2.2	2.3	2.4 「RTI LIN MultiMessage Blockset」(147 ページ)を参照してください。
RTI RapidPro Control Unit Blockset	2.1.1	2.2	2.2	2.2

製品名	dSPACE Release			
	2013-A	2013-B	2014-A	2014-B
RTI USB Flight Recorder Blockset	–	1.0	1.1	1.2 「RTI USB Flight Recorder Blockset」 (149 ページ)を参照してください。
RTI Watchdog Blockset	–	1.0	1.0	1.0
SCALEXIO Firmware	2.2	2.3	3.0	3.1 「SCALEXIO Firmware」(151 ページ)を参照してください。
SYNECT Server	1.1	1.2	1.3.1	1.4
SystemDesk 3.x ⁶⁾	3.2	3.2	–	–
SystemDesk 4.x ⁷⁾	4.0	4.1	4.2	4.3 「SystemDesk」(153 ページ)を参照してください。
TargetLink/TargetLink Data Dictionary	3.4	3.5	3.5	4.0 「TargetLink」(159 ページ)を参照してください。
Variable Editor	1.8	1.8	1.8	1.8
VEOS	3.0p3	3.1	3.2	3.3 「VEOS」(223 ページ)を参照してください。

¹⁾ 「dSPACE Release 2014-B での AutomationDesk 4.1 の使用」(39 ページ)に記載される制限事項に注意してください。

²⁾ ControlDesk 3.x は dSPACE Release 2013-A に付属しているものが最後のリリースでした。ControlDesk の後継製品である ControlDesk Next Generation に移行してください。移行の詳細については、「ControlDesk 3.x から ControlDesk Next Generation への移行」(☞『ControlDesk Next Generation 移行ガイド』)()を参照してください。

³⁾ ControlDesk Next Generation は、ControlDesk 3.x および CalDesk の後継となる製品です。

⁴⁾ 標準の I/O ブロックセットを含みます。

⁵⁾ RTI Gigalink Blockset を含みます。

⁶⁾ AUTOSAR 3.x のサポート

⁷⁾ AUTOSAR 4.x のサポート

定期的に更新を行っていない場合は、新機能と必要な移行手順について、上記の各 dSPACE Release の『新機能と移行手順』マニュアルを参照してください。

制限事項の詳細については、「dSPACE Release 2014-B での AutomationDesk 4.1 の使用」(39 ページ)を参照してください。

**ConfigurationDesk
(Implementation Version)**

ConfigurationDesk の主な新機能は、次のとおりです。

- 非同期タスクをモデリングするための新手法
- ビヘイビアモデルを使用せずにアプリケーションプロセスを使用する手法
- 1 つまたは複数の LIN コントローラを含む V-ECU インプリメンテーションのサポート
- SENT In ファンクションブロックと Lambda DCR、Lambda NCCR ファンクションブロックが拡張されています。
- ユーザーインターフェースが改良され、ConfigurationDesk アプリケーションをより簡単に使用でき、共通の設定タスクを使用する方法が便利になりました。

新機能の詳細については、「ConfigurationDesk – Implementation」(76 ページ)を参照してください。

**ControlDesk Next
Generation**

ControlDesk Next Generation (ControlDesk 5.3) の主な新機能は、次のとおりです。

- プラットフォーム/デバイスの拡張:
 - MicroLabBox のサポート
 - ビデオキャプチャリングデバイス:ビデオストリームのデータの再生
 - ブートファームウェアが破損した dSPACE リアルタイムハードウェアへのアクセス
 - CAN 用の Kvaser Leaf Light HS インターフェースのサポート
- プロジェクト/エクスぺリメント管理の拡張:
 - さまざまな SYNECT Server バージョンのサポート
- 変数管理の拡張:
 - 変数記述ファイルの圧縮
- 計器およびビジュアル表示の向上:
 - 新しい時間プロッタ
 - 計器に Python スクリプトを追加するオプション

- 計測および記録の拡張:
 - MDF 4.x データをシグナル単位でロードして大容量計測ファイル
を処理
 - 時間ベーススラストの停止トリガとしてのサンプルカウントトリガの
使用
- データセット管理の拡張:
 - 書き込み可能な計測をパラメータとして処理
- Bus Navigator の拡張:
 - dSPACE プラットフォームの CAN FD のサポート
 - モニタリング／ロギングと計測との同期
- ECU 診断の拡張:
 - Variable Browser での診断変数の説明の拡張
 - フォールトメモリ計器: DTC の数とレベルの表示
- Signal Editor の拡張:
 - VEOS シミュレーションでの V-ECU 変数へのスティミュラス信号の
入力
- 自動化の拡張:
 - 変数記述ファイルのメタ情報へのアクセス
 - 自動化によるメッセージの処理

新機能の詳細については、「ControlDesk Next Generation の新機能
(ControlDesk 5.3)」(86 ページ)を参照してください。

dSPACE FlexRay Configuration Package

dSPACE FlexRay Configuration Tool の主な新機能は、次のとおりです。

- データベースファイルとしての FIBEX 4.1 ファイルのサポート

新機能の詳細については、「dSPACE FlexRay Configuration Package
3.3 の新機能」(109 ページ)を参照してください。

dSPACE HIL API .NET

dSPACE HIL API .NET の主な新機能は次のとおりです。

- MicroLabBox のサポート
- VEOS 3.3
- 複数の環境 VPU を持つオフラインシミュレーションアプリケーション
へのスティミュラス信号の入力

新機能の詳細については、「dSPACE HIL API .NET」(99 ページ)を参照
してください。

dSPACE XIL API

dSPACE XIL API は、ASAM AE XIL API 2.0.1 規格をサポートする新しい製品で、ジェネリックシミュレータインターフェースを提供します。ASAM AE HIL API 1.0.2 規格の後継で、いくつかの変更と拡張が行われています。

XIL API の dSPACE インプリメンテーションでは、テストベンチ、モデルアクセスポート(MAPort)、電氣的欠陥シミュレーションポート(EESPort)をサポートします。

機能の詳細については、「dSPACE XIL API 2.0 の新機能」(103 ページ)を参照してください。

HIL API から XIL API への移行については、「dSPACE XIL API 2.0 への移行」(103 ページ)を参照してください。

ECU Interface Manager

ECU Interface Manager の主な新機能は、次のとおりです。

- Renesas RH85x マイクロコントローラのサポート
- 関数を削除するオプション
- 関数および書き込みアクセスの実行を永久に無効化するオプション
- マイクロコントローラレジスタに変数値を再び書き込むオプション
- すべてのインスタンスに同じサービス ID を使用するオプション

ECU Interface Manager の詳細については、「ECU Interface Manager 1.5 の新機能」(105 ページ)を参照してください。

Firmware Manager

Firmware Manager の主な新機能は次のとおりです。

- MicroLabBox のサポート
- DS4505 Interface Board または MicroAutoBox II 1401/1511/1512 に取り付けられる DS4342 CAN FD Interface Module のサポート
- 破損したブートファームウェアの修復がユーザインターフェースで可能

新機能の詳細については、「Firmware Manager 1.2 の機能」(107 ページ)を参照してください。

Model Compare

Model Compare の主な新機能は、次のとおりです。

- 名前または名前の一部での階層アイテムの全文検索
- バスサポートブロックの圧縮結果の表示の改良
- LSB 値の表示の改良
- dSPACE アドオンによる HIL モデル比較のサポート(オプション)

新機能の詳細については、「Model Compare 2.5 の新機能」(111 ページ)を参照してください。

ModelDesk	<p>ModelDesk の新しい主要機能は次のとおりです。</p> <ul style="list-style-type: none">■ エンジンモデルなどのシミュレーションモデルのパラメータを計測に基づいて完全に設定 <p>新機能の詳細については、「ModelDesk 4.0 の新機能」(115 ページ)を参照してください。</p>
MotionDesk	<p>MotionDesk の主な新機能は、次のとおりです。</p> <ul style="list-style-type: none">■ 新しい Multistate LED 計器■ 新しい動画の 3D オブジェクト(キャラクターおよび動物) <p>新機能の詳細については、「MotionDesk 3.5 の新機能」(117 ページ)を参照してください。</p>
Python Extensions	<p>MAPort の dSPACE HIL API Python Implementation の主な新機能は次のとおりです。</p> <ul style="list-style-type: none">■ MicroLabBox のサポート■ VEOS 3.3■ 複数の環境 VPU を持つオフラインシミュレーションアプリケーションへのステイミュラス信号の入力 <p>新機能の詳細については、「dSPACE Python Extensions」(101 ページ)を参照してください。</p>
Real-Time Testing	<p>Real-Time Testing の主な新機能は次のとおりです。</p> <ul style="list-style-type: none">■ VEOS 3.3 のサポート <p>新機能の詳細については、「Real-Time Testing 2.4 の新機能」(119 ページ)を参照してください。</p>
RTI、RTI-MP、RTLib	<p>RTI、RTI-MP、RTLib の主な新機能は、次のとおりです。</p> <ul style="list-style-type: none">■ MicroLabBox のサポート■ MicroAutoBox への拡張■ アップデートされたコンパイラでの C++コードのサポート■ コード生成設定用の新しいデフォルト設定 <p>新機能の詳細については、「RTI/RTI-MP および RTLib の新機能」(121 ページ)を参照してください。</p>

RTI Bypass Blockset

RTI Bypass Blockset の主な新機能は次のとおりです。

- MATLAB x64 のサポート
- ECU アプリケーションのバイナリコンテンツのラベル表示
- FlexRay パツファへの FlexRay フレームの割り当ての改良
- XCP 1.2 のサポート
- RTI Bypass Blockset MATLAB API への拡張

新機能の詳細については、「RTI Bypass Blockset 3.3 の新機能」(129 ページ)を参照してください。

RTI CAN Blockset

RTI CAN Blockset の主な新機能は次のとおりです。

- MicroLabBox のサポート

新機能の詳細については、「RTI CAN Blockset 3.3 の新機能」(133 ページ)を参照してください。

RTI CAN MultiMessage Blockset

RTI CAN MultiMessage Blockset の主な新機能は次のとおりです。

- MicroLabBox のサポート
- CAN FD (CAN with Flexible Data Rate) プロトコルのサポート
- データベースファイルとしての FIBEX 4.1 ファイルのサポート

新機能の詳細については、「RTI CAN MultiMessage Blockset 4.0 の新機能」(135 ページ)を参照してください。

NEW:RTI Electric Motor Control Blockset

RTI Electric Motor Control Blockset は、最新式のモーターのコントローラを実装する特殊機能を提供する新しいブロックセットです。MicroLabBox をサポートしています。

新しいブロックセットの詳細については、「RTI Electric Motor Control Blockset 1.0 の機能」(139 ページ)を参照してください。

RTI Ethernet Blockset

RTI Ethernet Blockset の主な新機能は次のとおりです。

- MicroLabBox のサポート
- RTLlib サポートのマニュアル

新機能の詳細については、「RTI Ethernet Blockset 1.1 の新機能」(141 ページ)を参照してください。

RTI FPGA Programming Blockset	<p>RTI FPGA Programming Blockset の主な新機能は次のとおりです。</p> <ul style="list-style-type: none">■ Xilinx®ソフトウェアのサポートの拡張■ MicroLabBox のサポート■ I/O ブロックのユーザ固有のチャンネル名などの一般的な機能拡張 <p>新機能の詳細については、「RTI FPGA Programming Blockset 2.8 の新機能」(143 ページ)を参照してください。</p>
RTI LIN MultiMessage Blockset	<p>RTI LIN MultiMessage Blockset の主な新機能は次のとおりです。</p> <ul style="list-style-type: none">■ データベースファイルとしての FIBEX 4.1 ファイルのサポート <p>新機能の詳細については、「RTI LIN MultiMessage Blockset 2.4 の新機能」(147 ページ)を参照してください。</p>
RTI USB Flight Recorder Blockset	<p>RTI USB Flight Recorder Blockset の主な新機能は次のとおりです。</p> <ul style="list-style-type: none">■ MicroLabBox のサポート■ RTLib サポートの別冊マニュアル <p>新機能の詳細については、「RTI USB Flight Recorder Blockset 1.2 の新機能」(149 ページ)を参照してください。</p>
SCALEXIO Firmware	<p>SCALEXIO Firmware の主な新機能は次のとおりです。</p> <ul style="list-style-type: none">■ 新しいバージョンのリアルタイム PC のサポート <p>新機能の詳細については、「SCALEXIO Firmware 3.1 の新機能」(151 ページ)を参照してください。</p>
SystemDesk 4.x	<p>SystemDesk 4.3 の主な機能は次のとおりです。</p> <ul style="list-style-type: none">■ AUTOSAR 4.1.3、4.1.2、4.1.1、4.0.3、4.0.2 のサポート <p>機能の詳細については、「新しい一般機能」(154 ページ)を参照してください。</p>
TargetLink	<p>TargetLink の主な新機能は、次のとおりです。</p> <ul style="list-style-type: none">■ 2D 信号(行列のサポート)■ 動的ルックアップテーブル■ バス信号の処理の改良(C 言語の構造体) <p>すべての新機能の詳細については、「TargetLink 4.0 および TargetLink Data Dictionary 4.0 の新機能」(160 ページ)を参照してください。</p>

TargetLink の移行に関する(TargetLink、TargetLink AUTOSAR モジュール、TargetLink Data Dictionary) 詳細については、「TargetLink 4.0 および TargetLink Data Dictionary 4.0 への移行」(187 ページ)を参照してください。

VEOS

VEOS の主な新機能は、次のとおりです。

- LIN バスシミュレーションのサポート
- オフラインシミュレーションでの V-ECU 変数へのスティミュラス信号の入力

新機能の詳細については、「VEOS」(223 ページ)を参照してください。

以前のリリースからの移行について

目的	最新の dSPACE Release の製品をインストールした後、いくつかの追加手順が必要になる場合があります。前回の dSPACE Release から移行する場合に必要な手順は、本書の製品固有の移行トピックスに記載されています。それ以前の dSPACE Release から移行する場合は、関連する『 新機能と移行手順 』を参照してください。
----	--

dSPACE Release 2014-B への移行

目的	Release 2014-B をインストールした後、いくつかの追加手順が必要な場合があります。
----	---

dSPACE Release 2014-A からの移行	製品固有の移行手順 製品ごとに必要な移行手順は、通常、製品ごとに自動的に実行されます。例外については、製品ごとの移行に関する説明を参照してください。
-----------------------------	---

dSPACE Release 2013-B 以前のリリースからの移行	dSPACE Release 2013-B 以前のリリースから Release 2014-B への移行を行うには、その間の dSPACE Release バージョンへの移行手順も併せて実行する必要があります。Release 2014-B をインストールした状態で、移行に必要なすべての手順を実行することができます。 必要な移行手順については、各 dSPACE Release バージョンの『 新機能と移行手順 』ドキュメントを参照してください。
------------------------------------	--

以前のリリースのドキュメント	以前のリリースの PDF ファイルの名前は、NewFeaturesAndMigrationxx.pdf (xx はリリース番号) です。
----------------	--

以前のリリースの『**新機能と移行手順**』ファイルは、次の場所にありません。

- 最新の dSPACE HelpDesk インストールフォルダの、
C:\Program Files<(x86)>\Common Files\dSPACE\HelpDesk 2014-B\Print\PreviousReleases を参照してください。
- dSPACE DVD の場合は、\Doc\Print\PreviousReleases を参照してください。
- <http://www.dspace.jp/goto.cfm/supver.rcphil> からダウンロードしてください。ここには、かなり以前のリリースの『**新機能と移行手順**』もあります。

Python 2.7 ディストリビューションの変更点

目的

dSPACE が提供する Python ディストリビューションでの変更点について説明します。

Python の以前のバージョンから Python 2.7 に移行する場合は、dSPACE Release 2013-B の『**新機能と移行手順**』ドキュメントに記載される移行手順を参照してください。

この情報は、dSPACE の Web サイトでも入手することができます (<http://www.dspace.com/ja/jpn/home/support/kb/supapnot/tatan/py27mig.cfm> を参照)。

項目の一覧

本章の内容

Python 2.7 の主な変更点	36
dSPACE Python ディストリビューションの主な変更点	36
Python インストールの使用に関する一般情報	37
標準の Python 2.7 ディストリビューションの機能拡張	37

Python 2.7 の主な変更点

目的 Python 2.7 ディストリビューションの変更点について記載します。

Python Software Foundation 提供のドキュメント『What's New』 Python のアップデートバージョンに関する『What's New』ドキュメントは、Python Software Foundation から入手することができます。

- 『What's New for Python 2.7』
(<http://docs.python.org/2.7/whatsnew/2.7.html>)

dSPACE Python ディストリビューションの主な変更点

目的 dSPACE が提供する Python ディストリビューションには、dSPACE 固有の変更点がいくつか含まれています。

新しいモジュールバージョン Python Core のバージョンが 2.7.8 にアップデートされ、最新リリースが含まれると同時に、バグフィックスが行われました。

PyWin32 のバージョンは 219.10 にアップデートされ、最新リリース 219.0 が含まれると同時に、dSPACE 固有の変更が行われています。

dSPACE Python ディストリビューションのコンポーネント dSPACE DVD の Python 2.7 ディストリビューションは、次の Python コンポーネントを提供します。

Python コンポーネント	バージョン
Python Core	2.7.8
PyWin32	219.10
Numpy	1.7.1
Matplotlib	1.2.1
WxPython	2.9.4.0
Py2exe	0.6.9
Comtypes	0.6.2
PIL	1.1.7
Python for .NET	2.0p1

Python インストールの使用に関する一般情報

目的	コンピュータで Python の両バージョンを使用する場合には、次の内容が当てはまります。
Python 2.5 と Python 2.7 の並列使用	<p>Python の両バージョンはコンピュータで並列使用できますが、次の制限事項があります。</p> <ul style="list-style-type: none"> ■ PY および PYW ファイルのファイル関連付けは、1 つの Python バージョンにのみ設定することができます。通常は、インストールされている中で最も新しい Python バージョンです。 ■ 環境変数は、Python の両バージョンによって使用されます。 PYTHONHOME の値などの環境変数の値は、使用する Python インストールに設定する必要があります。Python が設定する環境変数の概要については、http://docs.python.org/2/using/cmdline.html を参照してください。
Python の両バージョンの並列動作による dSPACE テストオートメーションの使用	dSPACE Release 2013-A まで提供されていた dSPACE Python 2.5 セットアップまたは dSPACE Python Extensions セットアップにより配布された dSPACE Python モジュールをテストオートメーションスクリプトで使用し、スクリプトを移行しない場合は、Python の両バージョンを使用する必要があります。

標準の Python 2.7 ディストリビューションの機能拡張

目的	標準 Python 2.7 には dSPACE 固有の機能拡張が一部含まれています。これらは以前と同じ動作を確保するか、既知のバグが解決されています。次の機能拡張は dSPACE Release 2014-B で使用することができます。
Python の既知のバグを解決するための拡張	<p>Python 2.7 の既知のバグを解決するために、次の変更が行われました。</p> <ul style="list-style-type: none"> ■ 以前のバージョンからの PyWin32 パッケージへの変更が採用されています。 ■ Python for .NET パッケージは、.NET 4.5.2 で実行できるように修正されています。

Python 2.7 のバグとそれらの解決方法の最新情報については、
<http://bugs.python.org> を参照してください。



dSPACE が変更した PyWin32 ファイルを識別するために、
バージョン番号を 219.0 から 219.10 に変更しています。

AutomationDesk

dSPACE Release 2014-B での AutomationDesk 4.1 の使用

dSPACE Release 2014-B で AutomationDesk 4.1 を使用する際の制限事項

AutomationDesk 4.1 は、dSPACE Release 2014-A でリリースされました。最新リリースの DVD に AutomationDesk の新規バージョンは含まれていません。そのため、いくつかの制限事項が存在します。

アクティブ化されている製品の確認

AutomationDesk で作業を始める前に、dSPACE Installation Manager を開いて現在アクティブになっている製品を確認してください。AutomationDesk によるリモートアクセスは、アクティブ化されている製品バージョンを使用して実行されます。

次の製品は、AutomationDesk のシーケンスに影響を及ぼす可能性があります。

- Python Extensions (Platform Management API を含む)
- HIL API .NET
- Real-Time Testing
- ControlDesk Next Generation
- RCP and HIL (ModelDesk を含む)

プラットフォームへのアクセス

dSPACE プラットフォームへのアクセスに関して、以下の制限事項があります。

■ MicroLabBox

dSPACE の新しいプラットフォームである MicroLabBox は、サポートされていません。

■ SCALEXIO

SCALEXIO プラットフォームを使用する場合、AutomationDesk でのプラットフォーム管理にいくつかの制約があります。SCALEXIO プラットフォームを登録できますが、アプリケーションをロードし、Platform Access ライブラリを介してアクセスすることはできません。

- アプリケーションをロードする場合は、Platform Management ライブラリを使用します。
- アプリケーションを使用する場合は、HIL API Convenience ライブラリまたは HIL API ライブラリを使用します。

■ VEOS

AutomationDesk でのプラットフォーム管理では、新しい VEOS バージョンを使用することはできません。VEOS を登録できないため、アプリケーションをロードすることができません。

- VEOS の登録には、ControlDesk Next Generation を使用します。



登録された VEOS プラットフォームは、AutomationDesk の Platform Manager に表示されません。

- アプリケーションをロードする場合は、Platform Management ライブラリを使用します。

デモプロジェクトの使用

VEOS、SCALEXIO、および Real-Time Testing にアクセスする AutomationDesk デモプロジェクトは、プラットフォーム固有のアプリケーションを使用します。これらの AutomationDesk デモプロジェクトを開始するには、これらのアプリケーションを Real-Time Testing とともにインストールされたサンプル実験のアプリケーションに事前に置き換える必要があります。

- 必要な ZIP アーカイブを展開します (たとえば、`C:\Program Files <(x86)>\Common Files\dSPACE\RealTimeTesting\2.4\Demos\SampleExperiments\TurnSignal_VEOS` にある `TurnSignal_VEOS.zip`)。

- 展開されたフォルダ(たとえば、TurnSignal_VEOS)を AutomationDesk の TurnSignalTests\SampleExperiments デモフォルダにコピーします。
- 必要に応じて、フォルダ名を以前に使用していた名前に変更するか、これらのアプリケーションファイルを参照する AutomationDesk デモプロジェクトのパスを更新します。

Automotive Simulation Model (ASM)

項目の一覧

本章の内容

ASM Base InCylinder Blockset	44
ASM Diesel Engine Blockset	45
ASM Diesel Exhaust Blockset	50
ASM Diesel InCylinder Blockset	51
ASM Drivetrain Basic Blockset	52
ASM Electric Components Blockset	54
ASM Engine Gasoline Basic Blockset	55
ASM Engine Gasoline Blockset	59
ASM Gasoline InCylinder Blockset	64
ASM Parameterization Tool	65
ASM Pneumatics Blockset	67
ASM Traffic Blockset	68
ASM Turbocharger Blockset	69
ASM Vehicle Dynamics Blockset	70

他章の参照情報

Migrating ASM Models ([📄](#) 『ASM User Guide』)

Provides general information on the migration process of ASM models.

ASM Base InCylinder Blockset

ASM Base InCylinder Blockset 1.9.1 への移行

EGR_COOLER

ブロック出力の名前が、Hdot_Out_ERG Cooler[J/s]から
Hdot_Out_EGR Cooler[J/s]に修正されています。

ASM Diesel Engine Blockset

項目の一覧

本章の内容

ASM Diesel Engine Blockset 2.0 の新機能	45
ASM Diesel Engine デモモデルの変更点	46
ASM Diesel Engine Blockset 2.0 への移行	47

ASM Diesel Engine Blockset 2.0 の新機能

HPP_CRANKBASED

HPP_CRANKBASED ブロックに、燃料システムの高圧ポンプ用のモデルが追加されました。高圧ポンプを通過する体積流量をクランク角の関数として計算します。

このブロックには、次の 2 つのモデル方式が含まれます。

- 平均値モデル
- パルス化モデル

どちらの方式も、流量計算のソースとして燃料供給時間を使用します。このポンプの制御信号は、クランク角の角度での燃料計測ユニットの通電の開始です。

RAIL

このブロックに、パラメータ Const_T_Rail が追加されました。

PUMP_TORQUE

圧縮の計算とピストン圧力差の実装が変更されました。

THROTTLE_MECHANICAL

これは、制御信号に従ってスロットルバルブ位置の計算を行う新しく追加されたブロックです。

Pos_Throttle[%]のスイッチが MDL_In ブロックに追加されました。

- SoftECU をオンにした場合、スロットル位置は THROTTLE_MECHANICAL ブロックによって計算されます。
- RealECU をオンにした場合、スロットル位置はモデルの外部(たとえば、実際のスロットルバルブや外部モデルなど)で定義する必要があります。

THROTTLE_VALVE	このブロックは変更されました。スロットル位置の計算が削除されました。この計算は、新しい THROTTLE_MECHANICAL ブロックで実行されます。
ENGINE_SETUP	ENGINE_SETUP は、新しく追加されたブロックですこのブロックには、エンジンの基本的な機械的パラメータが含まれます。また、エンジンの制御や EngineDiesel モデル内で使用する信号のサイズの設定に使用するパラメータも含まれます。
COMMON_ENGINE_PARAMETERS	これは、新しく追加されたブロックです。このブロックには、空気、排気、燃料の物理定数が含まれます。COMMON_DIESEL_PARAMETERS ブロックは廃止され、COMMON_ENGINE_PARAMETERS ブロックと ENGINE_SETUP ブロックに分割されました。
RAIL_CONTROL_CRANKBASED	これは、新しく追加されたブロックです。このブロックは、HPP_CRANKBASED ブロックのコントローラとして機能します。 このブロックは、燃料計測ユニット (FMU) の作動設定値を計算してレール圧を制御します。レール圧の設定値は、エンジンの運転ポイントに依存します。テストには、定数または外部設定値を使用できます。 このコントローラは、角度変換後の線形制御に対応した PI コントローラです。トリガモードを有効にすると、作動サイクルの前に FMU 作動の開始を計算し、ポンプの次の作動サイクルまでその開始点を維持できます。
LP_INTAKE_MANIFOLD	実装で吸排気通路での逆流が考慮されるようになりました。
EXHAUSTTHROTTLE	ルックアップテーブルパラメータ Map_p_diff_ExhThrottle の mdot_Exh および Pos_ExhThrottle 入力ポートが、T_Out_DPF と Vdot_In_ExhThrottle に置き換えられました。これは、元の入力ポートがこのパラメータの軸に適合していなかったためです。

ASM Diesel Engine デモモデルの変更点

FuelSystem	FuelSystem モデルに、電流ベースモデルとクランク角度ベースモデルの 2 つの高圧ポンプモデルが含まれるようになりました。
------------	---

High-Pressure Pump ライブラリを開くには、モデル内で HighPressurePump サブシステムの横にある Open High Pressure Pump Models をクリックします。ライブラリからブロックをドラッグアンドドロップして、HighPressurePump モデルを置き換えることができます。

/MDLUserInterface/ Environment/MDL_PAR	信号ルーティングが再構成されました。
/MDL/Environment	信号ルーティングが再構成されました。
/MDL/Environment/Driver	SignalSelection ブロックが ASM ライブラリにリンクされなくなりました。Simulink サブシステムになりました。
冷却システム	信号ルーティングが再構成されました。
スロットルバルブ位置	Pos_Throttle[%]のスイッチが MDL_In ブロックに追加されています。 <ul style="list-style-type: none"> ■ SoftECU をオンにすると、THROTTLE_MECHANICAL ブロックでスロットル位置が計算されます。 ■ RealECU をオンにする場合は、実際のスロットルバルブや外部モデルなどによって、モデルの外部からスロットル位置を定義する必要があります。
LP_EGR	排気スロットルを通る流量の計算が変更されています。最新バージョンでは、エンジンから出る排気ガス流量から高圧 EGR の流量と低圧 EGR クーラーを通る流量を引いて計算されます。

ASM Diesel Engine Blockset 2.0 への移行

COMMON_DIESEL_PARAMETERS_11_0	COMMON_DIESEL_PARAMETERS ブロックは廃止されました。このブロックは、新しいブロックである COMMON_ENGINE_PARAMETERS と ENGINE_SETUP に置き換えられています。 元のブロックのパラメータは新しいブロックに割り当てられます。COMMON_DIESEL_PARAMETERS ブロックは、新しいブロックを含む通常の Common_Diesel_Parameters Simulink サブシステムに変換されません。
-------------------------------	---

THROTTLE_VALVE_2_0	<p>THROTTLE_VALVE ブロックは廃止されました。このブロックは、新しいブロックである THROTTLE_MECHANICAL と THROTTLE_VALVE に置き換えられています。</p> <p>元のブロックのパラメータは新しいブロックに割り当てられます。THROTTLE_VALVE ブロックは、新しいブロックを含む通常の Simulink サブシステムに変換されます。</p>
SWITCHES_FUEL_METER_UNIT_1_0	<p>このブロックは廃止されました。このブロックのパラメータは、HIGH_PRESSURE_PUMP ブロックに移動されています。</p>
HIGH_PRESSURE_PUMP	<p>廃止された SWITCHES_FUEL_METER_UNIT_1_0 ブロックのパラメータは、HIGH_PRESSURE_PUMP ブロックに移動されています。以前のバージョンで、SWITCHES_FUEL_METER_UNIT のパラメータは HIGH_PRESSURE_PUMP モデルへの入力ポートでした。そのため、HIGH_PRESSURE_PUMP の移行後のバリエーションでは、新旧パラメータ構造体の設定が行われ、モデル内の以前の入力ポートが解消されます。移行前に Map_V_dead および Const_p_Low パラメータが追加されます。</p>
SWITCHES_THROTTLE_1_0	<p>このブロックは廃止されました。</p>
RAIL	<p>移行前のバリエーションに Const_T_Rail パラメータが追加されます。</p>
ENGINE_SETUP	<p>移行前のバリエーションに、Const_max_num_Inj_Direct、Const_max_num_Inj_Port、Const_m_Air_ref の各パラメータが追加されます。</p> <p>MDL.EngineDiesel.Const.Const_max_num_Inj パラメータの値は、MDL.EngineDiesel.Setup.Const_max_num_Inj_Direct パラメータに割り当てられます。</p> <p>MDL.EngineDiesel.Const.Const_max_num_Cyl パラメータの値は、移行後に MDL.EngineDiesel.Setup.Const_max_num_Cyl パラメータに割り当てられます。</p> <p>Const_max_num_Inj_Direct、Const_max_num_Inj_Port、Const_max_num_Cyl の各パラメータは、MDL_In システムでエンジンモデルの信号サイズを外部モデルや実際の ECU に合わせるのに使用されます。また、Operator ライブラリの ASM モデルは、これらの値を使用してそれぞれのマスクの下にある S-function へのサイズの入力を定義します。</p> <p>ModelDesk でこれらの値を書き込む機能は、新しく追加された機能です。そのため、ModelDesk プロジェクトを移行した後に、ModelDesk でこ</p>

れらを手動で指定する必要があります。MATLAB/Simulink モデルでは、これは ASM の移行によって書き込まれる移行前および移行後の IniFiles で自動的に実行されます。MATLAB で 2 度目に go を実行 (移行が完了) した後に、モデル構造でこれらの値を確認し、ModelDesk の対応するディーゼルのエンジンセットアップページに入力します。モデル構造:

MDL.EngineDiesel.Setup.Const_max_num_Cyl

MDL.EngineDiesel.Setup.Const_max_num_Inj_Port

MDL.EngineDiesel.Setup.Const_max_num_Inj_Direct

ASM Diesel Exhaust Blockset

ASM Diesel Exhaust Blockset 2.0

DIESEL_OXIDATION_
CATALYST

DOC を通じた負の逆流に対して値を 99 に設定することで、ラムダ値がマイナスになるのを回避することができます。

ASM Diesel InCylinder Blockset

ASM Diesel InCylinder デモモデルの変更

/MDLUserInterface/ Environment/MDL_PAR	信号ルーティングが再構成されました。
/MDL/Environment	信号ルーティングが再構成されました。
/MDL/Environment/Driver	SignalSelection ブロックが ASM ライブラリにリンクされなくなりました。 Simulink サブシステムになりました。

ASM Drivetrain Basic Blockset

項目の一覧

本章の内容

ASM Drivetrain Basic Blockset 4.0 の新機能	52
ASM Drivetrain Basic Blockset 4.0 への移行	52

ASM Drivetrain Basic Blockset 4.0 の新機能

TORQUE_CONTROLLER

負のトルク設定が存在する場合に、TORQUE_CONTROLLER ブロックを有効化することができます。

SIGNAL_SELECTION

SIGNAL_SELECTION ブロックがライブラリから削除されました。このブロックは通常の Simulink サブシステムとしてデモ内に残っています。また、このブロックの以前のバージョンは、次の Former Versions サブライブラリに含まれています。

ASM_DrivetrainBasic_lib/Driver/FormerVersions/SIGNAL_SELECTION_4_0

CYCLES

提供された信号に対して統一されたステップサイズを実現するため、CYCLES ブロック内の積分器が連続から離散に変更されています。

LUT1D

LUT1D は、新しく追加されたブロックです。このブロックを使用すると、モデル内に計測データを含めることができます。このブロック用に、新しいサブシステムが ASM_DrivetrainBasic_lib/Driver/Measurement に作成されています。

ASM Drivetrain Basic Blockset 4.0 への移行

TORQUE_CONTROLLER

移行時には、一連のブロックが Sw_TrqController_Mode[0Off|1On] 入力ポートに追加されるため、負のトルク設定が存在する場合、このブロックはアクティブでなくなります。これにより、このブロックの従来の機能が確保されます。

SIGNAL_SELECTION

移行時に、このブロックへのリンクは次の場所にある Former Versions サブライブラリに変更されます。

ASM_DrivetrainBasic_lib/Driver/FormerVersions/
SIGNAL_SELECTION_4_0

ASM Electric Components Blockset

項目の一覧

本章の内容

ASM Electric Components Blockset 2.7 の新機能	54
ASM Electric Components デモモデルの変更	54

ASM Electric Components Blockset 2.7 の新機能

ELECTRIC_MACHINE_BASIC	ELECTRIC_MACHINE_BASIC は、新しく追加されたブロックですこのブロックは、電流コントローラとパワーエレクトロニクスに対応したモーター (PMSM や SCIM など) の基本的な機能を表します。モーターは 1 次伝達関数 (PT1) としてシミュレートされます。
THREE_PHASE_DCM_INVERTER	THREE_PHASE_DCM_INVERTER は、新しく追加されたブロックですこのブロックは、ブリッジ構成で接続された最大 6 個の電源スイッチから成る、3 相不連続モード電力コンバータを表します。
SQUIRREL_CAGE_ASYNCHRONOUS_MACHINE_D_Q	逆起電圧出力ポートが計算されるようになりました。 新しい信号 (q 軸逆起電圧など) が ASMSignalBus に追加されました。
SCIM_CONTROLLER	弱め界磁コントローラが追加されました。ベース回転数範囲を超えてかご型誘導モーターを動かすことができます。

ASM Electric Components デモモデルの変更

Vehicle Electrical System デモ	ELECTRIC_MACHINE_BASIC ブロックが ASM_VehicleElectricalSystem デモモデルに追加の負荷として追加されています。
SQUIRREL_CAGE_ASYNCHRONOUS_MACHINE_D_Q	これはかご形誘導電動機の新しいデモモデルです。

ASM Engine Gasoline Basic Blockset

項目の一覧

本章の内容

ASM Engine Gasoline Basic Blockset 2.0 の新機能	55
ASM Engine Gasoline Basic デモモデルの変更	56
ASM Engine Gasoline Basic Blockset 2.0 への移行	56

ASM Engine Gasoline Basic Blockset 2.0 の新機能

THROTTLE_MECHANICAL

THROTTLE_MECHANICAL は新しく追加されたブロックで、制御信号に従ってスロットルバルブ位置の計算を行います。

Pos_Throttle[%]のスイッチが MDL_In ブロックに追加されています。

- SoftECU をオンにした場合、スロットル位置は THROTTLE_MECHANICAL ブロックによって計算されます。
- RealECU をオンにした場合、スロットル位置はモデルの外部(たとえば、実際のスロットルバルブや外部モデルなど)で定義する必要があります。

THROTTLE_VALVE

THROTTLE_VALVE は、THROTTLE ブロックに代わるブロックです。変更点の 1 つとして、THROTTLE_VALVE ブロックの Map_A_Red パラメータが 1 にスケールリングされています。

ENGINE_SETUP

ENGINE_SETUP は、新しく追加されたブロックです。このブロックには、エンジンの基本的な機械的パラメータが含まれます。また、エンジンの制御や EngineGasoline Basic モデル内で使用する信号のサイズの設定に使用するパラメータも含まれます。

COMMON_ENGINE_PARAMETERS

COMMON_ENGINE_PARAMETERS は、新しく追加されたブロックです。このブロックには、空気、排気、燃料の物理定数が含まれます。

COMMON_GASOLINE_PARAMETERS ブロックは廃止されました。COMMON_ENGINE_PARAMETERS ブロックと ENGINE_SETUP ブロックに分割されています。

FRICITION_TORQUE	FRICITION_TORQUE ブロックのルックアップテーブルが、摩擦圧力ではなく、摩擦トルクに基づいて計算されるようになりました。
------------------	---

ASM Engine Gasoline Basic デモモデルの変更

/MDLUserInterface/ Environment/MDL_PAR	信号ルーティングが再構成されました。
---	--------------------

/MDL/Environment	信号ルーティングが再構成されました。
------------------	--------------------

/MDL/Environment/Driver	SignalSelection ブロックが ASM ライブラリにリンクされなくなりました。Simulink サブシステムになりました。
-------------------------	---

冷却システム	信号ルーティングが再構成されました。
--------	--------------------

スロットルバルブ位置	<p>Pos_Throttle[%]のスイッチが MDL_In ブロックに追加されています。</p> <ul style="list-style-type: none"> ■ SoftECU をオンにすると、THROTTLE_MECHANICAL ブロックによってスロットル位置が計算されます。 ■ RealECU をオンにする場合は、実際のスロットルバルブや外部モデルなどによって、モデルの外部からスロットル位置を定義する必要があります。
------------	--

ASM Engine Gasoline Basic Blockset 2.0 への移行

COMMON_GASOLINE_PARAMETERS_8_0	<p>COMMON_GASOLINE_PARAMETERS ブロックは廃止されました。このブロックは、新しいブロックである COMMON_ENGINE_PARAMETERS と ENGINE_SETUP に置き換えられています。元のブロックのパラメータは新しいブロックに割り当てられます。</p>
--------------------------------	--

COMMON_GASOLINE_PARAMETERS ブロックは、新しいブロックを含む通常の Simulink サブシステム Common_Gasoline_Parameters に変換されています。

THROTTLE_2_0

THROTTLE ブロックは廃止されました。このブロックは、新しいブロックである THROTTLE_MECHANICAL と THROTTLE_VALVE に置き換えられています。元のブロックのパラメータは新しいブロックに割り当てられます。THROTTLE ブロックは、新しいブロックを含む通常の Simulink サブシステムに変換されています。

COMMON_ENGINE_PARAMETERS

移行前のバリエントに次のパラメータが追加されます。

- Const_Rm
- Const_Q_LHV
- Const_BulkModulus_Fuel
- Const_Fuel_Coeff
- Const_Air_Coeff
- Const_Exhaust_Coeff
- Const_R_Fuel
- Const_R_Exhaust

ENGINE_SETUP

移行前のバリエントに次のパラメータが追加されます。

- Const_max_num_Inj_Direct
- Const_max_num_Inj_Port
- Const_Inj_Matrix
- Const_Ratio_Comp
- Const_A_Piston
- Const_FiringOrder
- Const_num_Inj_Direct

MDL.EngineGasolineBasic.Const.Const_max_num_Cyl パラメータの値は、移行後に MDL.EngineGasolineBasic.Setup.Const_max_num_Cyl パラメータに割り当てられます。

MDL.EngineGasolineBasic.Const.Const_max_num_Inj.v パラメータの値は、移行後に

MDL.EngineGasolineBasic.Setup.Const_max_num_Inj_Direct に割り当てられます。

Const_max_num_Inj_Direct、Const_max_num_Inj_Port、Const_max_num_Cyl の各パラメータは、MDL_In システムでエンジンモデルの信号サイズを外部モデルや実際の ECU に合わせるのに使用されます。また、Operator ライブラリの ASM モデルは、これらの値を使用してそれぞれのマスクの下にある S-function へのサイズの入力を定義します。

ModelDesk でこれらの値を書き込む機能は、新しく追加された機能です。そのため、ModelDesk プロジェクトを移行した後に、ModelDesk でこれらを手動で指定する必要があります。MATLAB/Simulink モデルでは、これは ASM の移行によって書き込まれる移行前および移行後の IniFiles で自動的に実行されます。MATLAB で 2 度目に go を実行(移行が完了)した後に、モデル構造でこれらの値を確認し、ModelDesk の対応するガソリンのエンジンセットアップページにこれらを入力します。モデル構造:

```
MDL.EngineDiesel.Setup.Const_max_num_Cyl
```

```
MDL.EngineDiesel.Setup.Const_max_num_Inj_Port
```

```
MDL.EngineDiesel.Setup.Const_max_num_Inj_Direct
```

THROTTLE_VALVE

移行前のバリエントに Const_A_max パラメータが追加されます。

ASM Engine Gasoline Blockset

項目の一覧

本章の内容

ASM Engine Gasoline Blockset 3.0 の新機能	59
ASM Engine Gasoline デモモデルの変更	60
ASM Engine Gasoline Blockset 3.0 への移行	61

ASM Engine Gasoline Blockset 3.0 の新機能

HPP_CRANKBASED

HPP_CRANKBASED ブロックに、燃料システムの高圧ポンプ用のモデルが追加されました。高圧ポンプを通過する体積流量をクランク角の関数として計算します。

このブロックには、次の 2 つのモデル方式が含まれます。

- 平均値モデル
- パルス化モデル

どちらの方式も、流量計算のソースとして燃料供給時間を使用します。このポンプの制御信号は、クランク角の角度での燃料計測ユニットの通電の開始です。

RAIL

このブロックに、パラメータ Const_T_Rail が追加されました。

PUMP_TORQUE

圧縮の計算とピストン圧力差の実装が変更されました。

THROTTLE_MECHANICAL

これは新しく追加されたブロックで、制御信号に従ってスロットルバルブ位置の計算を行います。

Pos_Throttle[%]のスイッチが MDL_In ブロックに追加されています。

- SoftECU をオンにした場合、スロットル位置は THROTTLE_MECHANICAL ブロックによって計算されます。
- RealECU をオンにした場合、スロットル位置はモデルの外部(たとえば、実際のスロットルバルブや外部モデルなど)で定義する必要があります。

THROTTLE_VALVE	このブロックは、THROTTLE ブロックに代わるブロックです。変更点の 1 つとして、THROTTLE_VALVE ブロックの Map_A_Red パラメータが 1 にスケーリングされています。
ENGINE_SETUP	ENGINE_SETUP は、新しく追加されたブロックですこのブロックには、エンジンの基本的な機械的パラメータが含まれます。また、エンジンの制御や EngineGasoline モデル内で使用する信号のサイズの設定に使用するパラメータも含まれます。
COMMON_ENGINE_PARAMETERS	これは、新しく追加されたブロックです。このブロックには、空気、排気、燃料の物理定数が含まれます。COMMON_GASOLINE_PARAMETERS ブロックは廃止されました。COMMON_ENGINE_PARAMETERS ブロックと ENGINE_SETUP ブロックに分割されています。
RAIL_CONTROL_CRANKBASED	これは、新しく追加されたブロックです。このブロックは、HPP_CRANKBASED ブロックのコントローラとして機能します。 このブロックは、燃料計測ユニット (FMU) の作動設定値を計算してレール圧を制御します。レール圧の設定値は、エンジンの運転ポイントに依存します。テストには、定数または外部設定値を使用することができます。 このコントローラは、角度変換後の線形制御に対応した PI コントローラです。トリガモードを有効にすると、作動サイクルの前に FMU 作動の開始を計算し、ポンプの次の作動サイクルまでその開始点を維持することができます。
FRICTION_TORQUE	FRICTION_TORQUE ブロックのルックアップテーブルが、摩擦圧力ではなく、摩擦トルクのマップとして計算されるようになりました。

ASM Engine Gasoline デモモデルの変更

FuelSystem	FuelSystem モデルに、電流ベースモデルとクランク角度ベースモデルの 2 つの高圧ポンプモデルが含まれるようになりました。 High-Pressure Pump ライブラリを開くには、モデル内で HighPressurePump サブシステムの横にある Open High Pressure Pump Models をクリックします。ライブラリからブロックをドラッグアンドドロップして HighPressurePump モデルを置き換えることができます。
------------	--

/MDLUserInterface/ Environment/MDL_PAR	信号ルーティングが再構成されました。
/MDL/Environment	信号ルーティングが再構成されました。
/MDL/Environment/Driver	SignalSelection ブロックが ASM ライブラリにリンクされなくなりました。 Simulink サブシステムになりました。
冷却システム	信号ルーティングが再構成されました。
スロットルバルブ位置	Pos_Throttle[%]のスイッチが MDL_In ブロックに追加されました。 <ul style="list-style-type: none"> ■ SoftECU をオンにすると、THROTTLE_MECHANICAL によってスロットル位置が計算されます。 ■ RealECU をオンにする場合は、実際のスロットルバルブや外部モデルなどによって、モデルの外部からスロットル位置を定義する必要があります。
HighPressurePump	デフォルトの高圧ポンプモデルが、HIGH_PRESSURE_PUMP (電流ベースの高圧ポンプ)ではなく HPP_CRANKBASED (クランク角ベースの高圧ポンプ)になりました。

ASM Engine Gasoline Blockset 3.0 への移行

COMMON_GASOLINE_PARAMETERS_8_0	COMMON_GASOLINE_PARAMETERS ブロックは廃止されました。このブロックは、新しいブロックである COMMON_ENGINE_PARAMETERS と ENGINE_SETUP に置き換えられています。元のブロックのパラメータは新しいブロックに割り当てられます。 COMMON_GASOLINE_PARAMETERS ブロックは、新しいブロックを含む通常の Common_Gasoline_Parameters Simulink サブシステムに変換されています。
THROTTLE_4_0	THROTTLE ブロックは廃止されました。このブロックは、新しいブロックである THROTTLE_MECHANICAL と THROTTLE_VALVE に置き換えられています。元のブロックのパラメータは新しいブロックに割り当てられます。 THROTTLE ブロックは、新しいブロックを含む通常の Simulink サブシステムに変換されています。

SWITCHES_FUEL_METER_UNIT_1_0	このブロックは廃止されました。このブロックのパラメータは、HIGH_PRESSURE_PUMP ブロックに移動されています。
SWITCHES_THROTTLE_1_0	このブロックは廃止されました。
HIGH_PRESSURE_PUMP	<p>廃止された SWITCHES_FUEL_METER_UNIT_1_0 ブロックのパラメータは、HIGH_PRESSURE_PUMP ブロックに移動されています。以前のバージョンで、SWITCHES_FUEL_METER_UNIT のパラメータは HIGH_PRESSURE_PUMP モデルへの入力ポートでした。そのため、HIGH_PRESSURE_PUMP の移行後のバリエーションでは、新旧パラメータ構造体の設定が行われ、モデル内の以前の入力ポートが解消されます。移行前に Map_V_dead および Const_p_Low パラメータが追加されます。</p>
ENGINE_SETUP	<p>移行前のバリエーションに、Const_max_num_Inj_Direct および Const_max_num_Inj_Port が追加されます。</p> <p>MDL.EngineGasoline.Const.Const_max_num_Cyl のパラメータ値は、移行後のバリエーションで MDL.EngineGasoline.Setup.Const_max_num_Cyl パラメータに割り当てられます。</p> <p>MDL.EngineGasoline.Const.Const_max_num_Inj.v のパラメータ値は、移行後のバリエーションで MDL.EngineGasoline.Setup.Const_max_num_Inj_Direct に割り当てられます。</p> <p>Const_max_num_Inj_Direct、Const_max_num_Inj_Port、Const_max_num_Cyl の各パラメータは、MDL_In システムでエンジンモデルの信号サイズを外部モデルや実際の ECU に合わせるのに使用されます。また、Operator ライブラリの ASM モデルは、これらの値を使用してそれぞれのマスクの下にある S-function へのサイズの入力を定義します。</p> <p>ModelDesk でこれらの値を書き込む機能は、新しく追加された機能です。そのため、ModelDesk プロジェクトを移行した後に、ModelDesk でこれらを手動で指定する必要があります。MATLAB/Simulink モデルでは、これは ASM の移行によって書き込まれる移行前および移行後の IniFiles で自動的に実行されます。MATLAB で 2 度目に go を実行(移行が完了)した後に、モデル構造でこれらの値を確認し、ModelDesk の対応するガソリンのエンジンセットアップページにこれらを入力します。モデル構造:</p> <pre>MDL.EngineDiesel.Setup.Const_max_num_Cyl MDL.EngineDiesel.Setup.Const_max_num_Inj_Port MDL.EngineDiesel.Setup.Const_max_num_Inj_Direct</pre>

RAIL	移行前のバリエントに Const_T_Rail パラメータが追加されます。
COMMON_ENGINE_PARAMETERS	移行前のバリエントに、Const_Rm および Const_Q_LHV パラメータが追加されます。
THROTTLE_VALVE	移行前のバリエントに Const_A_max が追加されます。

関連トピック

基礎

- 「Migrating ASM Models」 (📖 『ASM User Guide』)

ASM Gasoline InCylinder Blockset

項目の一覧

本章の内容

ASM Gasoline InCylinder デモモデルの変更	64
ASM Gasoline InCylinder Blockset 1.9 への移行	64

ASM Gasoline InCylinder デモモデルの変更

/MDLUserInterface/
Environment/MDL_PAR

信号ルーティングが再構成されました。

/MDL/Environment

信号ルーティングが再構成されました。

/MDL/Environment/Driver

SignalSelection ブロックが ASM ライブラリにリンクされなくなりました。
Simulink サブシステムになりました。

ASM Gasoline InCylinder Blockset 1.9 への移行

HEAT_RELEASE_VIBE

Const_num_Cyl マスクパラメータは Const_max_num_Cyl に修正されています。

複数のメモリブロックの初期化で、このマスクパラメータを使用するように変更されています。

ones(MDL.InCylinderGasoline.Setup.Const_max_num_Cyl,v,1)が
ones(Const_max_num_Cyl,v,1)に置き換えられています。

CATALYST

PT2 伝達関数が更新されました。これは、T1 パラメータのリアルタイムパスに影響します。

ASM Parameterization Tool

項目の一覧

本章の内容

ASM Parameterization Tool 1.6.5 の新機能	65
ASM Parameterization Tool 1.6.5 への移行	65

ASM Parameterization Tool 1.6.5 の新機能

ModelDesk のプロセッシング

ModelDesk で、*プロセッシング*機能を利用することができます。この機能では、計測情報の管理とともに、計算関数の実行がサポートされません。

この新機能により、ModelDesk で ASMPParameterization Tool の機能がサポートされます。このリリースでは、平均有効圧エンジンモデルの新規ユーザは、ModelDesk でエンジンパラメータ設定プロジェクトを開始する必要があります。これは、平均有効圧エンジンモデルに対するデモのエンジンパラメータ設定が提供されていないためです。ただし、ASMPParameterization Tool は、dSPACE インストレーションとして提供されているため、以前のリリースからの移行が可能です。このように、ユーザは最新のリリースで既存のパラメータ設定プロジェクトを使用することもできます。

ASM Parameterization Tool 1.6.5 への移行

ModelDesk へのエクスポート

[Parameter Export to ModelDesk] ページが表示されなくなりました。ユーザは ModelDesk の新しい処理機能で提供されるインポート機能を利用する必要があります。

ModelDesk のインポートパラメータは、次の内容を含む GeneralSettings ファイルでサポートされます。

```
function Settings = MySettings()
```

```
Settings.PreFcn = '<Filepath with filename and extension of  
ImportMDLfromWorkspace.m>';
```

インストールされたエンジンパラメータ設定デモの zip ファイルから、ImportMDLfromWorkspace.m ファイルを事前にコピーしておいてください。

```
<dSPACE_Root>\Demos\ASM\<EnginePackage>\Parameterization\  
<ModelDesk-Project.zip>\Pool\Processing\Function\  
PreFcns\ImportMDLfromWorkspace.m
```

ModelDesk で処理構成を評価する場合、コマンドは現在の MATLAB ワークスペース構造を読み取り、既存のパラメータを ModelDesk エクスポートで使用するシミュレーションモデルの ModelDesk パラメータセットにインポートします。処理構成の評価は、ModelDesk の [Processing Configuration] ページの対応するボタンを使用して実行します。

ASM Pneumatics Blockset

項目の一覧

本章の内容

ASM Pneumatics Blockset 2.0 の新機能	67
ASM Pneumatics Blockset 2.0 への移行	67

ASM Pneumatics Blockset 2.0 の新機能

エアサスペンション

新しい Air Suspension サブライブラリには、次のトラックまたはトレーラ
のエアサスペンションシステムの実装用ブロックが含まれています。

- エアサスペンションバルブ
- 配管
- フロントおよびリアアクスルのエアスプリングとリフトペローズ

さらに、幅広いエアサスペンション構成を可能にする Setup ブロックも含まれています。

BrakePneumatics_ AirSuspension

このデモモデルは、新しく追加されたものです。これは、3本のアクスルを備えた車両用の空気供給、ブレーキ空気圧、エアサスペンションで構成されています。

BrakePneumatics_ AirSuspension_Trailer

このデモモデルは、新しく追加されたものです。これは、3本のアクスルを備えたトレーラ用の空気供給、ブレーキ空気圧、エアサスペンションで構成されています。

ASM Pneumatics Blockset 2.0 への移行

COMMON_PARAMETERS

このブロックには、空気の定積熱容量と等エントロピー指数に対応する新しい Const_cv_Air および Const_kappa_Air パラメータが追加されています。

このブロックには、空気の定積熱容量と等エントロピー指数に対する出力ポートが追加されています。

ASM Traffic Blockset

項目の一覧

本章の内容

ASM Traffic Blockset 3.1 の新機能	68
ASM Traffic Blockset 3.1 への移行	68

ASM Traffic Blockset 3.1 の新機能

SOFT_ECU_ACC

速度コントローラが、事前制御機能を含む新しいPIコントローラに置き換えられています。加速および減速の速さと最小／最大値が動的に設定され、車速に依存するようになりました。また、AEB(自動緊急ブレーキ)およびFCW(前方衝突警報)システムも追加されています。

ASM Traffic Blockset 3.1 への移行

SOFT_ECU ブロック

速度コントローラが、新しい機能を含むコントローラに置き換えられています。AEB(自動緊急ブレーキ)およびFCW(前方衝突警報)システムが追加されたため、パラメータを自動的に移行することはできません。

古いASMモデルの移行時には、SOFT_ECU サブライブラリへのリンクが以前のインプリメンテーション(FormerVersion サブシステムを参照)に変更されます。このため、シミュレーションの挙動は変更されません。新しいSOFT_ECU インプリメンテーションを使用する場合は、ブロックをASM_Traffic_lib ライブラリからモデルにドラッグして、必要に応じてパラメータを調整します。

この変更は、以下のブロックに反映されています。

- SOFT_ECU_ACC
- TARGET_SELECTION_ACC
- USER_INTERFACE_ACC
- OUTPUT_INTERFACE_ACC

ASM Turbocharger Blockset

ASM Turbocharger Blockset 3.0 の新機能

POSTTURBHPMAN

ブロックの出力ポートで比エンタルピーが使用できるようになりました。

ASM Vehicle Dynamics Blockset

項目の一覧

本章の内容

ASM Vehicle Dynamics Blockset 3.0 の新機能	70
ASM Vehicle Dynamics デモモデルの変更	72
ASM Vehicle Dynamics Blockset 3.0 への移行	72

ASM Vehicle Dynamics Blockset 3.0 の新機能

SUSCOMP_2D_FRONT

この新しいサスペンションコンプライアンスブロックは、左車輪と右車輪での力とトルクに依存するコンプライアンスをパラメータ化するために使用することができます。

SUSCOMP_2D_REAR

この新しいサスペンションコンプライアンスブロックは、左車輪と右車輪での力とトルクに依存するコンプライアンスをパラメータ化するために使用することができます。

TIRE_VEL_SAMPLING_POINTS

Magic Formula タイヤモデルは、前後方向のタイヤ速度に依存するさまざまなタイヤスリップ曲線のパラメータ化に使用することができます。このブロックでは、4つの速度サンプリングポイントを定義することができます。

STEERING_3DOF_VARIABLE_RATIO

STEERING_3DOF_VARIABLE_RATIO ブロックは、3つの慣性質量、対応する力の要素、および新しい摩擦の要素を使用して既存の STEERING ブロックを改良したものです。この 3DOF モデルでは、1DOF モデルよりも、車両の中心上のハンドリング動作の表現が向上しています。また、このブロックは可変ステアリングレシオを用いてパラメータ化することもできます。

SOFT_ECU_POWERSTEERING

補助パワーステアリングトルクをステアリングモデルに提供するため、SOFT_ECU_POWERSTEERING ブロックが Soft ECU ブロックセットに追加されました。

ドライブトレイン

Drivetrain モデルが拡張され、新しいモデリング方式を利用することができます。弾性シャフトを含む柔軟なドライブトレインの他に、リジッドなドライブトレインのシミュレートも可能になりました。これらの 2 つの方式の主な違いは、次のとおりです。

柔軟なドライブトレイン: 非弾性ドライブシャフトを使用したドライブトレインの動作をシミュレートし、ドライブトレインコンポーネントの弾性と慣性を考慮します。

非弾性ドライブシャフトを含めることで、ドライブトレインのダイナミクスと振動をシミュレートすることができます。ただし、各種コンポーネントの剛性が比較的高く、慣性が小さいことにより、硬い方程式系になる可能性があります。これにより、数値積分に困難が生じ、不安定なシステムにつながる可能性があります。そのため、この場合は、ドライブトレインモデル全体を安定化します。

リジッドなドライブトレイン: 非弾性ドライブシャフトを使用せずにドライブトレインの動作をシミュレートします。ドライブトレインコンポーネントの弾性や慣性は考慮しません。

このモデル方式を使用すると、シンプルなモデルと少ないパラメータでドライブトレインコンポーネントをシミュレートすることができます。また、比較的少ない労力で、標準のドライブトレインモデルを他の構成用に拡張することができます。

リジッドなドライブトレイン向けに、次のブロックが導入されています。

- CENTRAL_DIFFERENTIAL_RIGID
- REAR_DIFFERENTIAL_RIGID
- FRONT_DIFFERENTIAL_RIGID
- CLUTCH_RIGID
- CLUTCH_4WD_RIGID
- TORQUE_CONVERTER_RIGID
- GEARBOX_MT_RIGID
- GEARBOX_AT_RIGID
- TRANSFER_GEARBOX_RIGID
- CRANK_SHAFT_RIGID
- LOCKUP_CLUTCH_RIGID

柔軟なドライブトレインとリジッドなドライブトレインの両方に対応した標準デモとして、さまざまなドライブ構成がサポートおよび提供されています。これらの構成は、個別のサブシステムを交換し、関連する設定やスイッチを調整することで切り替えることができます。

ASM Vehicle Dynamics デモモデルの変更

サスペンションコンプライアンス	<p>新しい SUSCOMP_2D_FRONT ブロックがフロントアクスルのデフォルトのサスペンションコンプライアンスとして使用されます。</p> <p>新しい SUSCOMP_2D_REAR ブロックがリアアクスルのデフォルトのサスペンションコンプライアンスとして使用されます。</p>
ステアリング	<p>新しい STEERING_3DOF_VARIABLE_RATIO モデルがデフォルトのステアリングモデルとして使用されます。</p> <p>SOFT_ECU_POWERSTEERING ブロックが SoftECU サブシステムに追加されています。</p> <p>MDL_IN ブロックが拡張され、ステアリングモデルに提供される支援トルクを制御します。</p>
ドライブトレインサブシステム	<p>新しいリジッドドライブトレインアプローチにより、新しいショートカットが Drivetrain サブシステム内に追加されました。</p> <p>これらのショートカットで、フレキシブルドライブトレインとリジッドドライブトレインの両方に対応する以下のドライブトレイン関連のデモを開くことができます。</p> <ul style="list-style-type: none"> ■ Crankshaft デモ ■ Transmission System デモ ■ Final Drive Assembly デモ

ASM Vehicle Dynamics Blockset 3.0 への移行

TIRE_MODEL_MAGIC_FORMULA_xxx	<p>前後方向の速度に依存するタイヤパラメータ用の 2 つの入力ポートが新たに追加されています。</p> <p>XML ファイル (ModelDesk 向け) 内の ScalingParameter.LFZO パラメータの名前が ScalingParameters.LFZO に変更されました。</p>
CRANK_SHAFT	<p>CRANK_SHAFT ブロックに Sw_Trq_Mass_Mode パラメータが新しく追加されました。このパラメータは、信号ルーティングと入力ポートを介して提供されていました。移行を行うと対応する入力ポートの削除が補正されます。</p>

**DRIVETRAIN_VARIANT_
SWITCHES**

Sw_Trq_Mass_Mode パラメータは削除されました。このパラメータは、信号ルーティングと入力ポートを介して CRANK_SHAFT ブロックに提供されていました。このパラメータは CRANK_SHAFT ブロックに含まれており、入力として取得されることはありません。

AERODYNAMICS

風圧力の計算で、風速鉛直成分(v_z)が考慮されるようになりました。これにより、ピッチ時の風ベクトルの量が正しく考慮されます。

ConfigurationDesk

目的

ConfigurationDesk は、さまざまな使用例に適用できるツールです。リアルタイムアプリケーションの実装や、RapidPro ハードウェアの設定を行うことができます。

ConfigurationDesk – Implementation

項目の一覧

本章の内容

ConfigurationDesk 5.2 (Implementation Version) の新機能	76
ConfigurationDesk 5.2 への移行	81

ConfigurationDesk 5.2 (Implementation Version) の新機能

非同期タスクをモデリングするための新手法

ConfigurationDesk では、タスクおよびランナブルファンクションに I/O イベントを接続する方法が、新しくより便利になりました。

- ConfigurationDesk アプリケーションに既にタスクを使用するアプリケーションプロセスが含まれている場合は、ファンクションブロックの Properties Browser でそのタスクに I/O イベントを簡単に割り当てることができます。
- ConfigurationDesk アプリケーションに適切な Runnable Function ブロックを使用しないモデルのアプリケーションプロセスが含まれる場合は、ファンクションブロックのコンテキストメニューからアクセスできる [Generate Model Interface] - [Runnable Function Blocks and Tasks] コマンドを使用することができます。これにより、適切な Runnable Function ブロックが Simulink インターフェイスモデル内に作成されるだけでなく、対応するランナブルファンクションと I/O イベントを含むタスクも自動的に作成されます。

詳細については、「Basics on Modeling Asynchronous Tasks」(PDF『ConfigurationDesk Real-Time Implementation Guide』)を参照してください。

ビヘイビアモデルを追加するときに設定済みのアプリケーションプロセスを直接作成

ビヘイビアモデルを実行アプリケーションに追加するときに、ConfigurationDesk でそのビヘイビアモデルに設定済みのアプリケーションプロセスを直接作成することができます。その場合、[Add Model] または [Add/Replace Model Topology] ダイアログで [Create preconfigured application process] チェックボックスをオンにする必要があります。実行アプリケーションにプロセッサユニットアプリケーションが含まれていない場合は、設定済みのアプリケーションプロセスは新しいプロセッサユニットアプリケーションに作成されます。実行アプリケーションにプロセッサユニットアプリケーションが 1 つのみ含まれている場合

は、設定済みのアプリケーションプロセスはその既存のプロセッサユニットアプリケーションに作成されます。[Create preconfigured application process]チェックボックスはデフォルトでオンになっています。詳細については、「Add Model」(🔗『ConfigurationDesk Real-Time Implementation Reference』)を参照してください。

ビヘイビアモデルを使用せずにアプリケーションプロセスを使用

I/O ファンクションをビヘイビアモデルのポートに割り当てる必要がない、特定の使用事例があります。たとえば、シミュレータのサービスへの統合や、ケーブルのテストなどがあります。これらの用途の場合、ConfigurationDesk で、デフォルトタスクを提供する、あるいはモデルインプリメンテーションを割り当てる必要のない、特定のアプリケーションを作成することができます。詳細については、「Introduction to Application Processes Without Behavior Models」(🔗『ConfigurationDesk Real-Time Implementation Guide』)を参照してください。

ソースファイルを使用しない FMU のサポート

ConfigurationDesk では、ソースファイルを使用する FMU を、ソースファイルを使用せずに SCALEXIO 互換ライブラリファイルを使用する FMU に変換する手法を提供します。変換後の FMU は実行アプリケーションで使用することができます。詳細については、「Creating Precompiled FMUs」(🔗『ConfigurationDesk Real-Time Implementation Guide』)を参照してください。

V-ECU インプリメンテーションの新機能

サポートされる V-ECU インプリメンテーションコンテナのバージョン 次の表に、V-ECU インプリメンテーションコンテナをエクスポートするツールのバージョンと、関連するコンテナのバージョンを示します。

V-ECU インプリメンテーションコンテナのエクスポート元	コンテナのバージョン
SystemDesk 3.x	1.0
SystemDesk 4.2	2.0
SystemDesk 4.3	2.1
TargetLink 3.5	1.0
TargetLink 4.0	2.1

SystemDesk でエクスポートされた V-ECU の詳細については、「ECU コンフィギュレーション」(156 ページ)を参照してください。

LIN コントローラを含む V-ECU インプリメンテーションのサポート

ConfigurationDesk では、1 つまたは複数の LIN コントローラを含む V-ECU インプリメンテーションを実行アプリケーションに追加することができます。詳細については、「Special Aspects of V-ECU Implementations Containing LIN Controllers」(🔗『ConfigurationDesk Real-Time Implementation Guide』)を参照してください。



ConfigurationDesk の LIN 通信サポートには、以下の制限事項があります。

- LIN コンフィギュレーションでは、LIN マスターのみ定義することができ、LIN スレーブは定義することができません。
- LIN トランスポートプロトコルはサポートされません。
- LIN ノード設定サービスはサポートされません。

V-ECU インプリメンテーションのスタートアップの遅延

ConfigurationDesk では、リアルタイムアプリケーションをビルドして、含まれる V-ECU インプリメンテーションの開始を遅延させることができません。詳細については、「Delaying the Start of V-ECU Implementations」(📖『ConfigurationDesk Real-Time Implementation Guide』)を参照してください。

RTE インターベンションのサポート 以前の制限事項は該当しなくなりました。SCALEXIO も含む RTE インターベンションを使用する V-ECU インプリメンテーションを使用することができます。

既存の Simulink モデルへのモデルインターフェースの生成

ConfigurationDesk では、未解決または選択されたすべてのモデルポートブロック用のモデルインターフェースを、ConfigurationDesk アプリケーションに追加した Simulink モデルに直接生成することができます。詳細については、「Handling the Model Interface」(📖『ConfigurationDesk Real-Time Implementation Guide』)を参照してください。

MATLAB R2014a のサポート対象外の新機能

MATLAB R2014a で導入された以下の新機能はサポートされていません。

■ データディクショナリ

Simulink は、モデルが使用する設計データを永久リポジトリとしてのデータディクショナリに格納することができます。

MATLAB R2014b のサポート対象外の新機能

MATLAB R2014b で導入された以下の新しい機能はサポートされていません。

■ Simulink Function サブシステム

Simulink Function サブシステムでは、dSPACE Model Port Block Library のモデルポートブロックなどの dSPACE ブロックを使用することはできません。Simulink Function サブシステムの内容は、TRC ファイルには生成されません。

■ モデルテンプレート

モデルテンプレートでは、dSPACE Model Port Block Library のモデルポートブロックなどの dSPACE ブロックを使用することはできません。

拡張されたファンクションブロックタイプ

SENT In ファンクションブロック SENT In ファンクションブロックは、SAE J2716 JAN2010 SENT 規格に従って次の新機能を提供します。

- 定められた SENT メッセージ数による(短縮および拡張シリアルメッセージ形式の)シリアルメッセージの受信
- データ受信のためのあらかじめ設定されたデータプロトコルを使用して特定のセンサアプリケーションをサポート

基本情報については、「SENT In」(『ConfigurationDesk I/O Function Implementation Guide』)を参照してください。

Lambda DCR ファンクションブロック、Lambda NCCR ファンクションブロック Lambda DCR および Lambda NCCR ファンクションブロックを使用して、センサシミュレーションを GND ベースで動作させるか、dSPACE ハードウェアの GND から電氣的に絶縁させるかを指定できるようになりました。

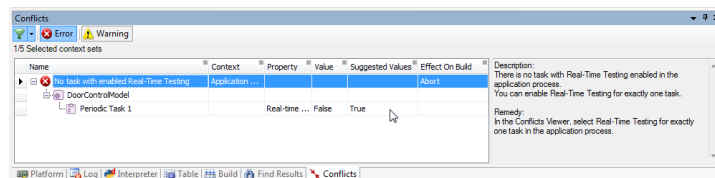
Isolated コンフィギュレーションを使用すると、両方の信号端子(信号とリファレンス)に同じ電流フローを使用して、二端子回路生成と同様のセンサシミュレーションを必要とする最新の ECU をサポートすることができます。

すべてのファンクションブロック 各ファンクションブロックのファンクションポートで、Model access パラメータを使用できるようになりました。このパラメータでは、モデルポートマッピングにより、ビヘイビアモデルへのアクセスの有効/無効を切り替えることができます。

ビヘイビアモデルのポート値にアクセスしない場合や、このようなファンクションポートに対して必要なモデルポートのみ作成されるようにする場合は、[Model access]を[Disabled]に設定することができます。

Conflicts Viewer の改良

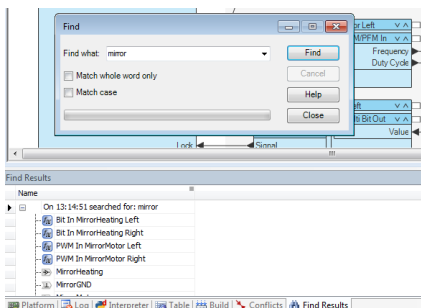
Conflicts Viewer が再設計され、競合するソースの検出と競合の解決がより簡単になりました。新しいフィルタ機能では、特定のコンテキストセットの競合や、任意の重要度の競合に焦点を当てることができます。



詳細については、「Resolving Conflicts」(『ConfigurationDesk Real-Time Implementation Guide』)を参照してください。

グローバル検索

新しい[Find]コマンドでは、名前で ConfigurationDesk アプリケーションの要素を検索することができます。Properties Browser で、アクセス可能なプロパティを持つあらゆる要素を検索することができます。検索結果は Find Results Viewer に整理されて表示されます。



詳細については、「How to Find Elements of a ConfigurationDesk Application」(『ConfigurationDesk Real-Time Implementation Guide』)を参照してください。

ブロックの折り畳みと展開

ポートとマッピングを処理し易くするために、グラフィカルなウィンドウで折り畳み矢印および展開矢印を使用してブロックの折り畳みと展開を行うことができます。

詳細については、「Collapsing and Expanding Blocks」(『ConfigurationDesk Real-Time Implementation Guide』)を参照してください。

ハードウェアリソースの割り当てで使用可能な新しい手法

ハードウェアリソース(チャンネルセット/チャンネル)は、使用可能なハードウェアトポロジから特定のファンクションブロックにドラッグアンドドロップして割り当てることができます。詳細については、「Methods for Assigning Hardware Resources」(『ConfigurationDesk Real-Time Implementation Guide』)を参照してください。

ConfigurationDesk 5.2 への移行

4.3 より前のバージョンのプロジェクトおよびコンポーネントファイルのサポート終了

ConfigurationDesk 4.2 以前の ConfigurationDesk プロジェクトを、ConfigurationDesk 5.2 に移行することはできません。4.3 より前のバージョンのアプリケーションコンポーネントファイル (DTF、ECH、HTF、MTF) も、移行することはできません。

TRC ファイル生成に関する変更

[Configuration Parameters]ダイアログの[Code Generation] - [DSRT variable description file options]ページの以下の設定が変更されています。

- [Apply subsystem read/write permissions]の設定は使用することができません。
- 次の設定は、デフォルトでオフになります。
 - [Include signal labels]
 - [Include virtual blocks]
 これらの新しいデフォルト値は、新規モデルや、モデルを別のプラットフォームに切り替える場合にのみ適用されます。
- [Include only Simulink.Parameter and Simulink.Signal objects with global storage class]の設定で、RTI CAN MultiMessage Blockset および RTI LIN MultiMessage Blockset の BusSystems グループが考慮されます。

Bus Selector の取り扱いの変更

MATLAB R2014a で、Simulink の Bus Selector ブロックが仮想化されています。dSPACE Release 2014-B と組み合わせて使用する際に、Bus Selector ブロックの入力ポートが非仮想バスに接続されている場合、このブロックやこのブロックに接続された他の仮想ブロックの出力ポートは、変数記述ファイルに生成されません。

バスの信号にアクセスするには、以下を実行します。

- バスから信号に直接アクセスします。
または
- 非仮想ブロックを信号に接続します (たとえば、係数 1 の Gain ブロック)。このブロックの出力ポートは、変数記述ファイルで利用することができます。

Bus Selector ブロックの入力ポートが仮想バスを介して接続されている場合、特別な方法を利用しなくても、出力ポートは変数記述ファイルに生成されます。

Simulink Coder で生成される コードの変更

MATLAB R2014a の使用 MATLAB/Simulink R2014a では、Simulink Coder によるコード生成が変更されています。そのため、ConfigurationDesk での変数記述ファイル (TRC ファイル) の生成も変更されています。

dSPACE Release 2014-B と MATLAB R2014a を組み合わせて使用する場合は、ビルドプロセスを開始する前に

`revertInlineParametersOffToR2013b` コマンドを使用する必要があります。このコマンドを使用すると、MATLAB R2014a での Simulink Coder の動作や dSPACE TRC ファイルの生成を、MATLAB R2013b 以前と同じにすることができます。`revertInlineParametersOffToR2013b` コマンドは、Simulink Coder 製品の一部として MATLAB R2014a に含まれています。

詳細については、次の URL にある Simulink Coder のリリースノートを参照してください。<http://www.mathworks.de/de/help/rtw/release-notes.html> (Mathworks のアカウントが必要です)

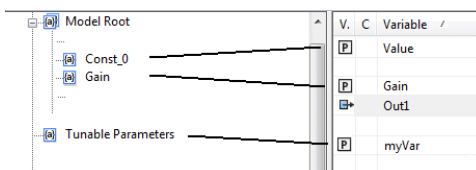
MATLAB R2014b の使用 [Inline Parameters]最適化オプションをオフにして作業した場合にかぎり、Simulink Coder の変更が生成される TRC ファイル内のエントリに影響を及ぼします。1 つまたは複数のブロックパラメータによって参照されるワークスペース変数は、モデル内でグローバルパラメータとして扱われ、ブロックのローカルパラメータとしては扱われません。これらは、生成される TRC ファイルの Tunable Parameters グループに格納されます。これらのグローバルパラメータの値を変更すると、変更したグローバルパラメータを参照するブロックパラメータも変更されます。シミュレーション結果が変わります。

例:

次の場合、ワークスペース変数はローカルブロックパラメータとして生成されます。

- パラメータが構造体または構造体項目である場合 (例:
`myStruct.Value`)
- パラメータが式または関数内で使用される場合 (例:`myValue + 1` または `sin(myValue)`)

マスクパラメータへの参照を使用する場合、ローカルブロックパラメータも生成されます。このマスクパラメータそのものは、変数記述ファイルには生成されません。



このモデルでは、Const_0 ブロックパラメータと Gain ブロックパラメータをワークスペース変数 `myVar` を使用して構成しています。

[Optimization]ページの[Inline parameters]オプションに関係なく、このワークスペース変数は変数記述ファイルに生成されます。これらの変数は、Const_0/Value と Gain/Gain を計器に接続することで、ControlDesk Next Generation で以前と同様に使用できますが、内部的にこれらのパラメータ値は `myVar` の値に依存しています。これらのパラメータのいずれかを変更すると、他のすべてのパラメータも変更されます。

カスタムファンクションブロックの移行

プロジェクトをロード後、一部またはすべてのカスタムファンクションブロックタイプが無効になっている場合は、次のいずれかの手順を実行します。

- [ConfigurationDesk Options]ダイアログの[Configuration]ページで、カスタムファンクションブロックのグローバル検索パスを不足しているカスタムファンクション XML ファイルを含むフォルダに変更します。関連するカスタムファンクションブロックタイプが自動的に有効になります。

または

- 各カスタムファンクションブロックタイプの XML ファイルを、プロジェクト固有の検索パスまたはグローバル検索パスのいずれかにコピーします。可能な場合、次のファイルも併せてコピーします。
 - ヘッダーファイル:<Function_block_type_name/CModule_name>.h
 - C++ソースコードファイル:
<Function_block_type_name/CModule_name>.cpp
 - タイプ定義ファイル:
<Function_block_type_name/CModule_name>_TypeDef.h

プロジェクト固有のカスタムファンクションディレクトリは、次のとおりです。

```
<DocumentsFolder>\<Project>\CustomFunctions
```

カスタムファンクションのデフォルトのグローバル検索パスは、次のとおりです。

```
<DocumentsFolder>\UserFiles
```

その後、Function Browser の[Reload Custom Function Definitions]コマンドを使用して、ファンクションブロックタイプを有効にすることができます。

既存のヘッダーまたはソースファイルが見つからない場合は、適切なエディタでカスタムファンクションブロックタイプの対応する XML ファイルを開き、<Module Name="XXX">タグの XXX が

<Function_block_type_name/CModule_name>と同じであることを確認します。

XML ファイルのみ所有している場合は、[Create Custom Function Code]で C++ソースコードファイルおよびヘッダーファイルを作成し、カスタムファンクションブロックタイプのコンテキストメニューの[Create Custom Function Type Definition]でタイプ定義ファイルを作成します。

変数記述ファイル内のカスタムファンクションパスの変更

変数記述ファイル内のカスタムファンクションの変数へのパスが変更されています。たとえば、ControlDesk Next Generation で ConfigurationDesk 5.2 から変数記述ファイルを再ロードする場合、これらの変数へのレイアウトおよびシグナジェネレータの接続が失われます。これらの変数は再接続する必要があります。

ControlDesk Next Generation

項目の一覧

本章の内容

ControlDesk Next Generation の新機能 (ControlDesk 5.3)	86
ControlDesk Next Generation への移行 (ControlDesk 5.3)	95

他章の参照情報

 ControlDesk Next Generation 移行ガイド ControlDesk 3.x、CalDesk、および ControlDesk Next Generation の以前のバージョンから ControlDesk 5.3 への移行について説明します。
 ControlDesk Next Generation Migration of ControlDesk 3.x Automation Explains migration from ControlDesk 3.x automation to ControlDesk Next Generation automation.

ControlDesk Next Generation の新機能 (ControlDesk 5.3)

項目の一覧

本章の内容

新しい一般機能 (ControlDesk 5.3)	86
プロジェクトおよび実験の新機能 (ControlDesk 5.3)	87
プラットフォーム管理およびプラットフォーム/デバイスの 新機能 (ControlDesk 5.3)	87
変数管理の新機能 (ControlDesk 5.3)	88
新しいビジュアル表示および計器機能 (ControlDesk 5.3)	89
新しい計測機能および記録機能 (ControlDesk 5.3)	91
Bus Navigator の新機能 (ControlDesk 5.3)	92
新しいデータセット管理機能 (ControlDesk 5.3)	92
ECU 診断の新機能 (ControlDesk 5.3)	93
Signal Editor の新機能 (ControlDesk 5.3)	93
新しい自動化機能 (ControlDesk 5.3)	94

新しい一般機能 (ControlDesk 5.3)

リボンのカスタマイズ

カスタムリボン制御を追加して、ControlDesk のリボンをカスタマイズすることができます。各カスタムリボン制御は Python *拡張スクリプト* にリンクさせる必要があります。これにより、ControlDesk の自動化インターフェースに基づくカスタムファンクションをユーザインターフェースに追加することができます。

「Customizing the Ribbon」(📖『ControlDesk Next Generation Basic Practices Guide』)を参照してください。

プロジェクトおよび実験の新機能 (ControlDesk 5.3)

さまざまな SYNECT Server バージョンのサポート

SYNECT Server で ControlDesk を使用する場合、ControlDesk でさまざまな dSPACE Release の SYNECT Server バージョンを使用できるようになりました (dSPACE Release 2014-B の SYNECT Server バージョン以降)。

プラットフォーム管理およびプラットフォーム/デバイスの新機能 (ControlDesk 5.3)

MicroLabBox のサポート

ControlDesk では新しい MicroLabBox をサポートします。「RTI/RTI-MP および RTLib の新機能」(121 ページ)も参照してください。

CAN 用の Kvaser Leaf Light HS インターフェースのサポート

ControlDesk で、Kvaser 社製の Leaf Light HS インターフェースが新たにサポートされています。

「Supported CAN Interfaces」(☞『ControlDesk Next Generation Basic Practices Guide』)を参照してください。

自動化によって ControlDesk を起動する場合のプラット フォーム検索

自動化によって ControlDesk を起動する場合も、プラットフォーム検索オプションを指定することができます。プラットフォーム検索とは、ControlDesk 起動時に登録プラットフォームを検索することです。

詳細については、「Platform Management Page」(☞『ControlDesk Next Generation Reference』)を参照してください。

ブートファームウェアが破損し た dSPACE リアルタイムハード ウェアへのアクセス

ControlDesk 5.3 以降、ブートファームウェアが破損した dSPACE リアルタイムハードウェアは、ControlDesk の Platform/Device Manager に ⚠ 記号付きで表示されるようになりました。

Log Viewer エントリをチェックして、必要に応じてファームウェアアップデートを実行します。「Update Firmware」(☞『ControlDesk Next Generation Reference』)を参照してください。

メモリセグメント管理の機能拡張

- ControlDesk で A2L ファイルのすべてのメモリセグメントタイプをサポートします。
- オンライン適合の起動時に、個々のメモリセグメントの内容をデバイスのミラーメモリのセグメントの内容と比較するかどうかを指定することができます。
- 各メモリセグメントに対して個別に、ECU イメージファイルの生成時にエクスポートするかどうかを指定できるようになりました。

詳細については、「Memory Segments」(📖『ControlDesk Next Generation Reference』)を参照してください。

ビデオキャプチャリングデバイス:ビデオストリームのデータの再生

計測時に他のプラットフォームの信号と同期してビデオストリームを再生して、実際のカメラをシミュレーションすることもできます。

「How to Configure Video Capturing Devices」(📖『ControlDesk Next Generation Basic Practices Guide』)を参照してください。

VEOS プラットフォーム:シミュレーション一時停止時の新しい自動化イベント

VEOS プラットフォームで、実行中のオフラインシミュレーションが一時停止のときのイベントが提供されます。

DS1005 ベースおよび DS1006 ベースのマルチプロセッサシステム:プロセッサ名の編集

DS1005 ベースまたは DS1006 ベースのマルチプロセッサシステムに含まれるボードに、プロセッサ名を指定することができます。

「Edit Processor Names」(📖『ControlDesk Next Generation Reference』)を参照してください。

変数管理の新機能(ControlDesk 5.3)

長時間のプロジェクトのロードの回避

プロジェクトのロード時間を短縮するために、プロジェクトの変数記述ファイルを含むデータベースの内容を圧縮することができます。これにより、データベースの不要なデータが削除されます。

「Problem with Long Project Loading Times」(📖『ControlDesk Next Generation Basic Practices Guide』)を参照してください。

新しいビジュアル表示および計器機能(ControlDesk 5.3)

本章の内容

時間プロッタ	89
計器への Python スクリプトの追加	90
ストップウォッチ計器	90
Instrument Selector の自動化	90
計器選択の向上	91
サブグループのすべての変数をレイアウト上に配置	91

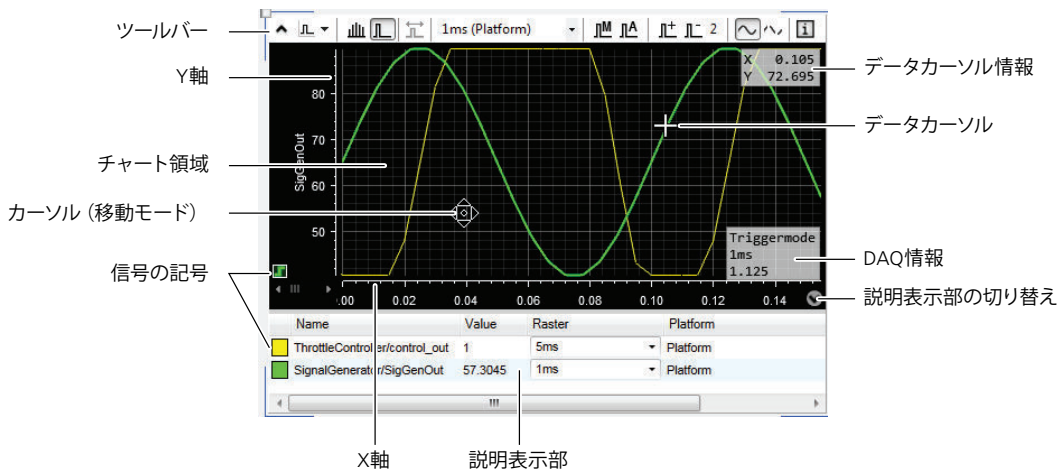
時間プロッタ

ControlDesk の新しい *時間プロッタ* では、時間ベースラスタ(時間プロット)で計測される信号を表示することができます。

標準のプロッタとの比較:

- 新しい時間プロッタはパフォーマンスが向上しています。
- 時間プロッタの取り扱いは、インデックスプロッタ(ControlDesk 5.2 で導入)および XY プロッタと同じです。
- 時間プロッタ、XY プロッタ、インデックスプロッタの適応分野は明確に分かれています。そのため、各プロッタタイプの取り扱いの複雑さが軽減されます。

下図に計測の例を示します。



詳細については、「Time Plotter」(『ControlDesk Next Generation Reference』)を参照してください。

標準のプロッタと時間プロッタの違いの概要については、「Differences Between Plotter, Time Plotter, Index Plotter, and XY Plotter」(☞『ControlDesk Next Generation Basic Practices Guide』)を参照してください。

計器への Python スクリプトの追加

各計器に Python スクリプトを追加して、選択した計器のイベントに Python コードを割り当てることができます。これにより、自動化を使用してより柔軟に計器の機能を拡張することができます。

含まれるレイアウトを開くときなど、計器を初期化するたびに、スクリプトは自動的に実行されます。

スクリプトは計器とともに保存されます。そのため、計器をコピー／ペーストすると、コピーされた計器にはコピー元と同じ機能拡張が含まれます。

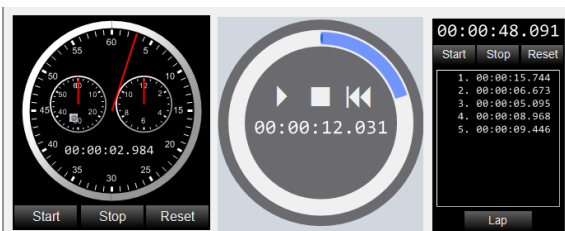
拡張機能を含むスクリプトは、以下の場合でも計器に維持されます。

- 計器をコピー／ペーストする場合。
- 計器を ControlDesk の Instrument Selector で [Custom Instruments] リストに追加する場合。これにより、計器のカスタマイズと個別化を行うことができます。

「Adding a Python Script to an Instrument」(☞『ControlDesk Next Generation Basic Practices Guide』)を参照してください。

ストップウォッチ計器

Instrument Selector に、さまざまなストップウォッチが追加されました。



ラップタイムの開始、停止、表示などのストップウォッチの機能は、計器スクリプトを使用してストップウォッチに追加されます。「Example of Adding a Python Script to an Instrument」(☞『ControlDesk Next Generation Basic Practices Guide』)を参照してください。

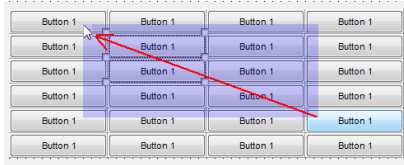
Instrument Selector の自動化

Instrument Selector にツール自動化によってアクセスすることができます。

計器選択の向上

レイアウト上での計器の選択が向上しました。

たとえば、計器グループ内の計器を選択する場合、**Ctrl+Shift** を押しながら計器の上にポインタを移動します。正方形の記号で、選択した計器がビジュアル表示されます。

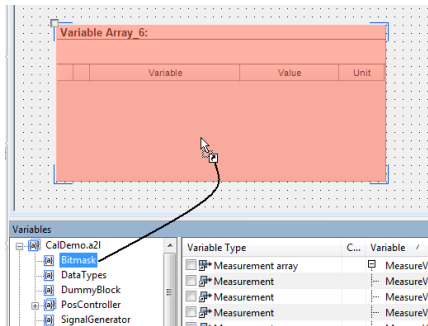


「Selecting and Positioning Instruments on a Layout」

(☞『ControlDesk Next Generation Basic Practices Guide』)を参照してください。

サブグループのすべての変数をレイアウト上に配置

計器がサブグループのすべての変数(ブロックやサブシステムのすべての変数など)を表示できる場合は、変数ツリーから計器にグループノードをドラッグすることができます。



新しい計測機能および記録機能(ControlDesk 5.3)

MDF 4.x データをシグナル単位でロードして大容量計測ファイルを処理

Measurement Data Pool で MDF 4.x ファイルを開くと、含まれる信号をレイアウトにビジュアル表示するとき、信号を個別にロードすることができます。これにより、大容量 MDF 4.x ファイルの後処理を行うことができます。

「Measurement Files Page」(☞『ControlDesk Next Generation Reference』)を参照してください。

サンプルカウントトリガを時間ベーススラストの停止トリガとして使用

サンプルカウントトリガでは、データ取得でのサンプル数を指定できるほか、時間ベーススラストの停止トリガとしても使用できるようになりました。「Basics on Triggers」(📖『ControlDesk Next Generation Basic Practices Guide』)を参照してください。

Bus Navigator の新機能 (ControlDesk 5.3)

dSPACE プラットフォームの CAN FD のサポート

Bus Navigator で、以下の dSPACE プラットフォームの CAN FD (CAN with Flexible Data Rate) がサポートされます。

- DS1005
- DS1006
- DS1007
- MicroAutoBox

「Features of the Bus Navigator Specific for CAN」(📖『ControlDesk Next Generation Advanced Practices Guide』)を参照してください。

モニタリングおよびロギングと計測との同期

測定データをさまざまなプラットフォーム/デバイスから取得する場合でも、データのモニタリングおよびロギングを計測データと同期することができます。

「Bus Navigator Page」(📖『ControlDesk Next Generation Reference』)を参照してください。

新しいデータセット管理機能 (ControlDesk 5.3)

書き込み可能な計測をパラメータとして処理

パラメータのみでなく、書き込み可能な計測変数を含むデータセットを作成することができます。

「Adding Writable Measurement Variables to Data Sets」(📖『ControlDesk Next Generation Basic Practices Guide』)を参照してください。

変数記述ファイルの再ロード時の自動データセット処理

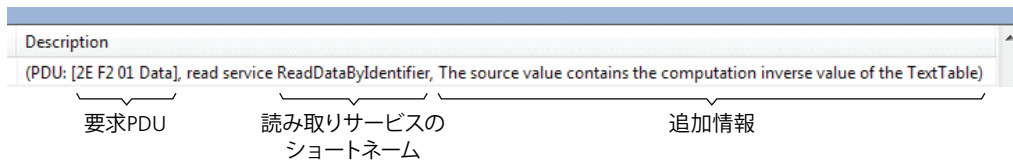
変数記述ファイルの再ロードまたは置き換え後に、自動的にデータセットを削除するか復元するか指定することができます。

「Data Set Manager Page」(📖『ControlDesk Next Generation Reference』)を参照してください。

ECU 診断の新機能(ControlDesk 5.3)

Variable Browser での診断変数の説明の拡張

Variable Browser で、可能な場合には、診断変数の説明テキストに関連する要求 PDU の情報、読み取りサービスの名前(パラメータの場合のみ)、詳細な追加の説明情報が表示されるようになりました。この情報は、括弧内に表示されます。この情報を使用して変数リストをフィルタすることができます。



計器列のカスタマイズ

フォールトメモリ計器と診断計器の列をカスタマイズできるようになりました。つまり、列の追加や削除、列の順序の変更を行うことができます。

下記を参照してください。

- 「Columns (Fault Memory Instrument)」(📖『ControlDesk Next Generation Reference』)
- 「Columns (Diagnostics Instrument)」(📖『ControlDesk Next Generation Reference』)

フォールトメモリ計器:DTCの数とレベルの表示

フォールトメモリ計器で DTC の数とレベルも表示されます。

「Fault Memory Instrument」(📖『ControlDesk Next Generation Reference』)を参照してください。

Signal Editor の新機能(ControlDesk 5.3)

VEOS シミュレーションでの V-ECU A2L 変数へのステイミュラス信号の入力

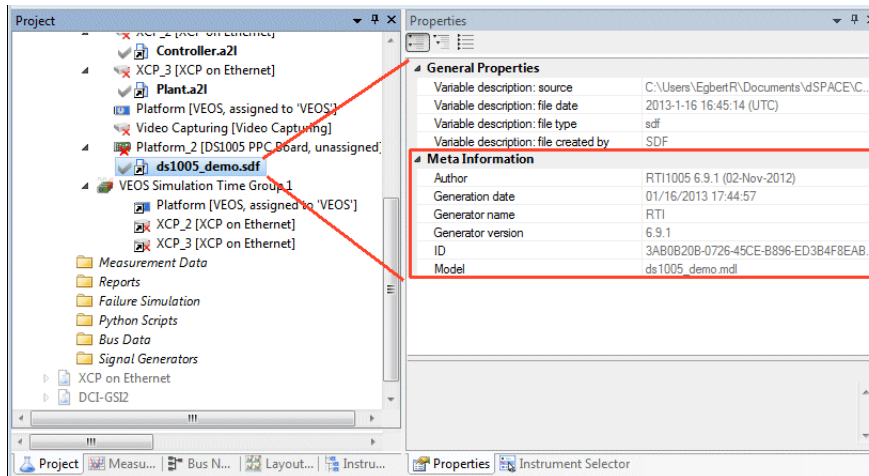
ControlDesk の Signal Editor で、VEOS でのオフラインシミュレーションの際に V-ECU の A2L 変数へのステイミュラス信号の入力が可能になりました。

詳細については、「How to Stimulate Variables of a Simulation Application」(📖『ControlDesk Next Generation Advanced Practices Guide』)を参照してください。

新しい自動化機能(ControlDesk 5.3)

変数記述ファイルのメタ情報へのアクセス

ツール自動化で変数記述ファイルのメタ情報にアクセスできるようになりました。



メッセージの使用

ControlDesk に、自動化でメッセージを処理するインターフェースが追加されています。このインターフェースでは、ログファイルに追加される情報、警告、エラーなどあらゆる種類のメッセージを監視することができます。また、これらのメッセージをメッセージの重要度などによってフィルタすることや、自動化によって特定のメッセージの発生に対して反応することもできます。

詳細については、「Automating Message Handling」(『ControlDesk Next Generation Advanced Practices Guide』)を参照してください。

ControlDesk Next Generation への移行 (ControlDesk 5.3)

項目の一覧

本章の内容

ControlDesk Next Generation への移行 (ControlDesk 5.3)	95
---	----

他章の参照情報

 ControlDesk Next Generation 移行ガイド ControlDesk 3.x、CalDesk、および ControlDesk Next Generation の以前のバージョンから ControlDesk 5.3 への移行について説明します。	
 ControlDesk Next Generation Migration of ControlDesk 3.x Automation Explains migration from ControlDesk 3.x automation to ControlDesk Next Generation automation.	

ControlDesk Next Generation への移行(ControlDesk 5.3)

ControlDesk 5.2 から ControlDesk ControlDesk 5.3 に移行して既存のエクスペリメントを再利用するには、次の移行手順が必要になる場合があります。

本章の内容

ControlDesk 5.3 の廃止	96
CANcardX サポートの廃止	96
フォールト読み取り関数に対する ODX セマンティクス準拠のデフォルトサービスの廃止	96
ControlDesk 5.3 への移行	96
MultiCaptureHistoryEnabled プロパティの動作の変更	96
Remove メソッドの動作の変更 (IXaMainRecorder インターフェース)	96
TextTable パラメータタイプのサポートによる変更	96
CalDesk、ControlDesk 3.x、または ControlDesk Next Generation 4.3 以前のバージョンからの移行	97

ControlDesk 5.3 の廃止

CANcardX サポートの廃止 Vector Informatik 社製の CANcardX インターフェースのサポートは終了しました。

フォールト読み取り関数に対する ODX セマンティクス準拠のデフォルトサービスの廃止 ControlDesk では、次の ECU 診断関数に対して、ODX セマンティクスに準拠したデフォルトの診断サービスは提供されなくなりました。

- Reading fault memory entries
- Reading environment data
- Clearing single or all fault memory entries

ただし、上記の関数に対する ControlDesk でのプロトコル固有のサービス識別情報に準拠したデフォルトの診断サービスは、引き続き提供されます。ODX セマンティクス準拠のデフォルトサービスからプロトコル固有のサービス識別情報準拠のデフォルトサービスの使用に変更するには、ご使用の自動化スクリプトでサービスの名前を調整する必要があります。

XML 設定ファイルを使用すると、デフォルトの診断サービスと異なるサービスをセットアップすることができます。「Basics of the XML Configuration File」(📄『ControlDesk Next Generation Advanced Practices Guide』)を参照してください。

ControlDesk 5.3 への移行**MultiCaptureHistoryEnabled プロパティの動作の変更**

ControlDesk 5.2 以前では、「MeasurementTriggeredRaster / IXaMeasurementTriggeredRaster <<Interface>>」インターフェースの MultiCaptureHistoryEnabled プロパティの動作に誤りがありました。これは ControlDesk 5.3 以降では修正されています。

このプロパティ設定は、計測の実行中に変更できなくなりました。ControlDesk 5.3 以降でこのプロパティ設定を計測の実行中に変更しようとする、例外が発生します。必要に応じて、スクリプトを調整する必要があります。

Remove メソッドの動作の変更 (IXaMainRecorder インターフェース)

ControlDesk 5.2 以前では、「MainRecorder / IXaMainRecorder <<Interface>>」インターフェースの Remove メソッドの動作に誤りがありました。これは ControlDesk 5.3 以降では修正されています。

レコーダは実行中に削除できなくなりました。ControlDesk 5.3 以降で実行中のレコーダを削除しようとする、例外が発生します。必要に応じて、スクリプトを調整する必要があります。

TextTable パラメータタイプのサポートによる変更 ControlDesk 5.2 以前では、診断サービスなどのパラメータに対して TextTable タイプはサポートされていませんでした。代わりに、TextTable タイプのパラメータに対して文字列タイプが使用されていました。

ControlDesk 5.3 では、パラメータに対して TextTable タイプがサポートされます。

そのため、このパラメータタイプを自動化スクリプトで使用している場合は、スクリプトを調整する必要があります。

CalDesk、ControlDesk 3.x、 または ControlDesk Next Generation 4.3 以前の バージョンからの移行

CalDesk、ControlDesk 3.x、または ControlDesk Next Generation の以前のバージョンから移行して既存のエクスペリメントを再利用するには、追加の移行手順が必要な場合があります。移行手順については、「ControlDesk Next Generation への移行」(☞『ControlDesk Next Generation 移行ガイド』)を参照してください。

dSPACE HIL API .NET

dSPACE HIL API .NET 1.7 の新機能

プラットフォームサポートの強化

MAPort を使用する dSPACE HIL API .NET インプリメンテーションは、MicroLabBox をサポートします。

MAPort を使用する dSPACE HIL API .NET インプリメンテーションは、以下をサポートします

- MicroLabBox
- VEOS 3.3
- 複数の環境 VPU を持つオフラインシミュレーションアプリケーションへのスティミュラス信号の入力

dSPACE HIL API .NET インプリメンテーションの詳細については、 『dSPACE HIL API .NET Implementation Document』を参照してください。

EESPort サポートの廃止

dSPACE HIL API .NET インプリメンテーションは、モデルアクセス用の MAPort のみサポートする dSPACE HIL API Python インプリメンテーションに適合されています。そのため、欠陥シミュレーション用の EESPort は廃止されました。欠陥シミュレーションを自動化するには、後継の HIL API に移行します（「dSPACE XIL API 2.0 への移行」(103 ページ)を参照）。

ユーザマニュアルの場所の変更

dSPACE HIL API インプリメンテーション(.NET および Python)のユーザマニュアルは、dSPACE HelpDesk の[Software]から [Software] - [Test Automation]に移動しました。

dSPACE Python Extensions

dSPACE Python Extensions 1.7 の新機能

dSPACE Platform Management API

dSPACE Platform Management API は MicroLabBox をサポートします。プラットフォーム ID は *DS1202* です。

詳細については、[📖](#) 『dSPACE Platform Management API Reference』を参照してください。

プラットフォームサポートの強化

MAPort の HIL API Python Implementation を使用する dSPACE Python Extensions 1.7 では、以下をサポートします。

- MicroLabBox
- VEOS 3.3
- 複数の環境 VPU を持つオフラインシミュレーションアプリケーションへのステイミュラス信号の入力

dSPACE HIL API Python Implementation の詳細については、[📖](#) 『dSPACE HIL API Python Implementation Document』を参照してください。

ユーザマニュアルの場所の変更

dSPACE HIL API インプリメンテーション(.NET および Python)のユーザマニュアルは、dSPACE HelpDesk の[Software]から [Software] - [Test Automation]に移動されました。

dSPACE XIL API

項目の一覧


本章の内容

dSPACE XIL API 2.0 の新機能	103
dSPACE XIL API 2.0 への移行	103

dSPACE XIL API 2.0 の新機能


dSPACE XIL API の基礎

dSPACE XIL API は、ASAM AE HIL API の後継である ASAM AE XIL API 2.0.1 規格の、dSPACE 固有のインプリメンテーションです。dSPACE インプリメンテーションは、C#/ .NET でプログラミングされ、モデルアクセスポート (MAPort) と電氣的欠陥シミュレーションポート (EESPort) を提供するテストベンチをサポートします。

dSPACE XIL API インプリメンテーションの詳細については、 『dSPACE XIL API Implementation Guide』を参照してください。

dSPACE XIL API 2.0 への移行

dSPACE HIL API .NET から dSPACE XIL API .NET へのア プリケーションの移行

必要な移行手順については、「Migrating HIL API Scripts to XIL API Scripts」( 『dSPACE XIL API Implementation Guide』)を参照してください。

ECU Interface Manager

項目の一覧

本章の内容

ECU Interface Manager 1.5 の新機能	105
ECU Interface Manager 1.5 への移行	106

ECU Interface Manager 1.5 の新機能

Renesas RH85x マイコン ローラのサポート

ECU Interface Manager では、Renesas RH85x マイコンローラをサポートします。

関数の削除

ECU Interface Manager では、メモリスペースの解放などのためにすべての関数を永久に削除することができます。

削除した関数は復元することができます。

「Delete」(🔗『ECU Interface Manager Reference』)を参照してください。

関数および書き込みアクセス の実行を永久に無効化

ECU Interface Manager では、関数および書き込みアクセスを永久に無効することができます。これにより、メモリスペースが節約されます。

「Insert Control Logic - Disable Execution」(🔗『ECU Interface Manager Reference』)を参照してください。

レジスタへの再書き込み

制御される書き込みアクセスにおいて、ECU Interface Manager で変数の新しい値をマイクロコントローラレジスタに再び書き込み、データの整合性を保証することができます。

「Insert Control Logic - Register Write Back」(☞『ECU Interface Manager Reference』)を参照してください。

すべてのインスタンスに同じサービス ID を使用

ECU Interface Manager で、関数呼び出しや書き込みアクセスのすべてのインスタンスに同じサービス ID を使用することができます。

「Functions and Variables Configuration Pane」(☞『ECU Interface Manager Reference』)を参照してください。

ECU Interface Manager 1.5 への移行

前のバージョンの ECU Interface Manager で最後に保存したプロジェクトの移行

ECU Interface Manager 1.5 では、前のバージョンの ECU Interface Manager で最後に保存したプロジェクトを再利用することができます。ただし、ECU Interface Manager 1.5 でプロジェクトを保存した後は、前のバージョンの ECU Interface Manager でそのプロジェクトを使用できなくなります。

新しい名前でのプロジェクトの保存 前のバージョンの ECU Interface Manager で保存したプロジェクトを ECU Interface Manager 1.5 で開いて保存しようとする、新しい名前で保存することを求めるプロンプトが表示されます。これにより、前のバージョンの ECU Interface Manager で使用できるプロジェクトのバージョンを維持することができます。

Firmware Manager

Firmware Manager 1.2 の機能

プラットフォームサポートの強化

Firmware Manager では、以下の新しい dSPACE ハードウェアをサポートします。

- MicroLabBox

ファームウェアアーカイブ DS1202FwArchive.arc は、ボードのファームウェアコンポーネントを提供します。

- DS4342 CAN FD Interface Module

モジュールのファームウェアは以下のファームウェアアーカイブで使用することができます。

- DS1005
- DS1006
- DS1007
- MicroAutoBox

ユーザインターフェースを使用したブートファームウェアの復元

dSPACE プラットフォームは DS1104 R&D Controller Board を除き、ブートファームウェアが破損している場合も Platform Manager で認識することができます。ブートファームウェアの復元に、コマンドラインユーティリティを使用する必要はなくなりました。Firmware Manager のユーザインターフェースを使用して、ブートファームウェアを復元することができます。

DS1104 R&D Controller Board については、引き続きコマンドラインユーティリティを使用する必要があります。

dSPACE FlexRay Configuration Package

dSPACE FlexRay Configuration Package 3.3 の新機能

FlexRay Configuration Tool

FIBEX 4.1 のサポート FlexRay Configuration Tool で、FlexRay ネットワーク記述用の FIBEX 4.1 ファイルもサポートされます。

Model Compare



この製品は米国での使用が禁止されています

米国では Model Compare を使用することはできません。この製品を米国内で使用することも第三者に使用させることも米国の法律に違反します。

項目の一覧

本章の内容

Model Compare 2.5 の新機能	111
Model Compare 2.5 への移行	112

Model Compare 2.5 の新機能

階層アイテムの全文検索

[Reference]および[Comparison Hierarchies]にリストされるアイテムの名前を、Model Navigator で全文検索することができます。

関連ドキュメント

- 「Model Navigator」([📖](#) 『Model Compare Reference』)

バスの比較結果の表示

バス信号の内部階層が Property Inspector に表示されるようになりました。

関連ドキュメント

- 「Property Inspector」([📖](#) 『Model Compare Reference』)

LSB 値の表示の改良	可能な場合、任意の LSB を 2 のべき乗表記でも表示できるようになりました。2 のべき乗値は、次のように丸括弧内に表示されます。0.125 (2 ⁻³)。
お気に入りの設定へのアクセスの改良	お気に入りの比較設定にメニューバーからもアクセスすることができます。 関連ドキュメント ■ 「Favorites」 (📖 『Model Compare Reference』)
HIL モデル比較の任意サポート	Model Compare では、dSPACE アドオンによって HIL モデルへのサポートが向上しています。dSPACE にお問い合わせください。

Model Compare 2.5 への移行

Save as default (ボタン)	[Comparison Settings] ダイアログの [Save as default] ボタンのラベルが変更されています。新しいラベルは [Save as user default settings] です。お気に入りリストで、[Use factory default settings] に加えて [Use user default settings] を選択できるようになりました。 関連ドキュメント ■ 「Save as User Default Settings」 (📖 『Model Compare Reference』)
線照合	4 つの線照合アルゴリズムから選択することができます。 工場出荷時のデフォルト値が、[Identical source and destination nodes and port numbers] (Model Compare バージョン 2.4 では最初のオプションで、デフォルト値) から [Identical destination node] (Model Compare バージョン 2.3 以前では最後のオプションで、デフォルト値) に変更されています。 関連ドキュメント ■ 「General Page」 (📖 『Model Compare Reference』)
XML ダンプ	Model Compare で TargetLink モデルから XML ダンプファイルを作成できるのは、MIL シミュレーションモードの場合のみです。



このバージョンの Model Compare で作成された XML ダンプファイルのみを使用することをお勧めします。それ以外の場合、Model Compare の新機能の一部が使用できなくなります。

関連ドキュメント

- 「MATLAB で XML ダンプファイルを作成する方法」(📖『Model Compare ガイド』)

ModelDesk

ModelDesk 4.0 の新機能

Processing

ModelDesk の Processing コンポーネントが新しくなりました。このコンポーネントでは、ASMPParameterization ツールより便利な方法で、計測に基づいてシミュレーションモデルのパラメータを設定することができます。これは特に、テストベンチで計測されたデータを使用してエンジンモデルのパラメータを設定する場合に便利です。

Processing コンポーネントで計測データを準備することができます。計測からの生データを、パラメータ計算に使用される計測データに適合させることができます。パラメータ値は、MATLAB で M 関数を使用して計算されます。算出された値は、ModelDesk でモデルのパラメータページに書き込むことができます。

処理に必要なすべてのデータ(計測データ、M 関数など)は、ModelDesk プロジェクトに統合されます。

ツール自動化

ModelDesk のツール自動化で、以下の機能を使用することができます。

- 道路網のトラフィックオブジェクト、ジャンクション、経路へのアクセス
- 処理の実行の開始(上記参照)

Road Generator

新しい[Road Network]パネルには、交通網の道路エレメントとジャンクションエレメントのすべてがリストされます。[Road]ペインと[Junction]ペインは省略されます。[Road Network]ペインでは、複数の道路エレメントとジャンクションエレメントで構成される道路のセクションをコピー／ペーストすることができます。

定義した道路を MATLAB MAT ファイルにインポートできるようになりました。

パラメータ設定

Automotive Simulation Models このリリースでは、Automotive Simulation Models のパラメータを設定することができます。Automotive Simulation Models の詳細については、「Automotive Simulation Model (ASM)」(43 ページ)を参照してください。

パラメータのプロパティ パラメータページでモデルパラメータを選択すると、[Properties]ペインにそのプロパティが表示されます。

MotionDesk

項目の一覧

本章の内容

MotionDesk 3.5 の新機能	117
MotionDesk 3.5 への移行	118

MotionDesk 3.5 の新機能

計器

MotionDesk に、さまざまな色で変数の値を表示できる Multistate LED 計器が追加されました。Multistate LED 計器には、複数の数値範囲とそれらの適切な表示形式を指定することができます。

3D オブジェクトライブラリ

MotionDesk の 3D オブジェクトライブラリには、動画のキャラクターおよび動物が含まれます。3D オブジェクトはシミュレーションに依存せず動画化されますが、プロパティで動画の種類(歩くや走るなど)および外観を指定することができます。他のどの移動可能 3D オブジェクトについても、それらの位置はアプリケーションからのデータストリームをベースにすることができます。

新しい 3D オブジェクトは追加ライセンスで保護されます。

ライブラリには、男性、女性、子供が含まれます。また牛、犬、馬などの動物も含まれています。次の図に、走っている女性の動画を示します。



ツール自動化

アクティブなプロジェクトの名前とパスを取得することができます。

MotionDesk 3.5 への移行

MotionDesk 2.1.6 以前からの移行

MotionDesk 2.2 以降では、実験データを管理するためのプロジェクト／実験の構造が変更されています。したがって、それより以前の MotionDesk プロジェクトは移行する必要があります。詳細については、「Migrating from MotionDesk 2.1.6 and Lower」(📖『MotionDesk Guide』)を参照してください。

MotionDesk 2.2.1 以前からの移行

MotionDesk では 3D オブジェクトに新しい形式を使用しているため、シーンとカスタム 3D オブジェクトを移行する必要があります。詳細については、「Migrating from MotionDesk 2.2.1 and Lower」(📖『MotionDesk Guide』)を参照してください。

MotionDesk 3.4 からの移行

LED 計器は自動的に Multistate LED 計器に移行されます。

Real-Time Testing

Real-Time Testing 2.4 の新機能

サポートされるプラットフォーム

- Real-Time Testing 2.4 は VEOS 3.3 をサポートします。現在、Real-Time Testing は常に有効で、複数の環境モデルで使用することができます。

RTI/RTI-MP および RTLlib

項目の一覧

本章の内容

RTI/RTI-MP および RTLlib の新機能	121
RTI/RTI-MP および RTLlib の移行上の注意点	125

RTI/RTI-MP および RTLlib の新機能

MicroLabBox

MicroLabBox は、ラボ用の新しいシングルボードハードウェアです。使用事例によって、MicroLabBox のご注文の際に次の 2 つの形式のいずれかを選択することができます。アナログ入出力チャンネルとしてボード前面に 2 つの D-Sub コネクタを搭載する *前面コネクタパネル* と、ボードの上部にある別個の BNC コネクタをアナログ入出力チャンネルとする *上部コネクタパネル* です。

ボードは、Ethernet インターフェースを使用してホスト PC に容易に接続することができます。

MicroLabBox には、モデルの計算を行うデュアルコアプロセッサ (Freescale P5020) が 1 つと、ホスト通信を行うシングルコアプロセッサ (Freescale P1011) が 1 つ搭載されています。これにより、高い処理能力、入出力の低レイテンシ、データ取得時の高いデータスループットを実現しています。デュアルコアプロセッサでは、RTI-MP で MicroLabBox をマルチコアプラットフォームとして使用することができます。

MicroLabBox には次の I/O チャンネルがあります。

- アナログ入力チャンネル
 - サンプリングレートが同じ 1 MSPS の 24 個の差動チャンネル
 - サンプリングレートが同じ 10 MSPS の 8 個の差動チャンネル
- アナログ出力チャンネル
 - 16 個のシングルエンドチャンネル
- デジタル双方向チャンネル
 - 48 個のシングルエンドチャンネル
 - 12 個の差動チャンネル

MicroLabBox では、CAN Type 1 ライブラリなどのボードライブラリに加えて、シミュレーションデータを USB 大容量記憶装置に記録できる RTI USB Flight Recorder Blockset を使用することができます。RTI Ethernet Blockset によって、ボードの Ethernet 機能を使用して Simulink モデルに Ethernet 通信を実装することができます。

ボード機能の詳細については、[図 1](#)『MicroLabBox Features』を参照してください。

MicroLabBox は、RTI FPGA Programming Blockset でプログラミングできる FPGA モジュールを提供します。詳細については、「RTI FPGA Programming Blockset 2.8 の新機能」(143 ページ)を参照してください。

RTI および RTI-MP を使用する場合の MicroLabBox 固有の機能 RTI および RTI-MP で MicroLabBox を使用する場合は、以下の問題を考慮する必要があります。

- [RTI General Build Options]ダイアログの[Data set storage]オプションは常に設定され、変更することはできません。
- リアルタイムアプリケーションのダウンロードに使用するプラットフォームを識別するために、プラットフォーム名またはネットワーククライアントアドレスのいずれかを使用することができます。

- リアルタイムアプリケーションをダウンロードすると、CmdLoader のパラメータを通してシミュレーションの初期状態の値が設定されます。シミュレーションの初期状態を変更するためにアプリケーションを再ビルドする必要はありません。たとえば、ダウンロードを開始する前に[Multiprocessor Setup]ダイアログの値を設定するのみで十分です。
- シミュレーション状態に関連するオプションは、RUN または STOP のいずれかのみを設定されます。PAUSE というシミュレーション状態はありません。

制限事項 RTI および RTI-MP で MicroLabBox を使用する場合は、下記の制限事項があります。

- *As fast as possible* 実行モードはサポートされません。
- RTI-MP で MicroLabBox を使用できますが、RTI Gigalink ブロックセットは MicroLabBox をサポートしません。
- MicroLabBox は RTI Bypass Blockset をサポートしません。
- MicroLabBox は Real-Time Testing をサポートしません。
- MicroLabBox は MotionDesk Blockset をサポートしません。

MicroAutoBox

MicroAutoBox の一部のハードウェアおよびソフトウェアが機能強化されました。

SCALEXIO サポートの拡張 MicroAutoBox を DS1507 または DS1512 とともに使用すると、DS4342 CAN FD Interface Module を使用して CAN サポートを拡張することができます。これにより、CAN FD プロトコルに基づいて CAN バスを実装することができます。

詳細については、[📖 『MicroAutoBox Features』](#)を参照してください。

MicroAutoBox 設定ツール MicroAutoBox 設定ツール DS1401ConfigGUI.exe のユーザビリティが一部改良されています。

MATLAB R2014a のサポート 対象外の新機能

MATLAB R2014a で導入された以下の新しい機能は、dSPACE ブロックセットでサポートされていません。

- データディクショナリ
Simulink は、モデルが使用する設計データを永続リポジトリとしてのデータディクショナリに格納することができます。

MATLAB R2014b のサポート 対象外の新機能

MATLAB R2014b で導入された以下の新しい機能は、dSPACE ブロックセットでサポートされていません。

- Simulink Function Subsystem

Simulink Function Subsystem では、RTI および RTI-MP ブロック (Default CPU ブロックを除く) などの dSPACE ブロックや、ボード固有の RTI ブロックセットのブロックを使用することはできません。Simulink Function Subsystem の内容は、TRC ファイルには生成されません。

- モデルテンプレート

モデルテンプレートでは、RTI および RTI-MP ブロックなどの dSPACE ブロックや、ボード固有の RTI ブロックセットのブロックを使用することはできません。

アップデートされた DS1006 GNU C/C++コンパイラ

DS1006 Processor Board のコンパイラツールチェーンがアップデートされ、パフォーマンス、安定性、互換性が改良されました。

次の新機能があります。

- GNU コンパイラバージョン 4.8.3

- ツールチェーンには、標準の C ライブラリ機能を提供する組み込みシステム用の、Newlib C ライブラリのポートが含まれています。

- このツールチェーンで、カスタムコード用の C++ がサポートされるようになりました。別個の C++ Integration Kit のインストールは必要なくなります。



RTI によってモデルから生成されるコードは C コードです。
[Code Generation] ページの [Language] オプションの
[C++] 設定がサポートされなくなりました。

- ツールチェーンでは、C++ 例外処理および RTTI がサポートされます。ただし、パフォーマンスおよびタイミング上の理由から、これらの機能は使用しないことをお勧めします。

例外処理は、重大なエラーにのみ使用してください。リアルタイムアプリケーションは、例外が発生すると終了します。

アップデートされた Microtec PowerPC C/C++ Compiler

PowerPC ボード (DS1005、DS1103、DS1104、MicroAutoBox) 用のコンパイラツールチェーンがアップデートされ、カスタムコード用の C++ サポートが提供されるようになりました。別個の C++ Integration Kit のインストールは必要なくなります。



RTI によってモデルから生成されるコードは C コードです。
[Code Generation] ページの [Language] オプションの [C++] 設定がサポートされなくなりました。

RTI/RTI-MP および RTLib の移行上の注意点

TRC ファイルの生成に関する変更

[Code Generation] ダイアログの [RTI variable description file options] ページの以下の設定が変更されました。

- [Apply subsystem read/write permissions] 設定を使用することはできません。
- 以下の設定は、デフォルトではクリアされています。
 - [Include signal labels]
 - [Include virtual blocks]

新しいデフォルト値は、新しいモデルに対してのみ、またはモデルを別のプラットフォームに切り替えた場合にのみ関連があります。
- [Include only Simulink.Parameter and Simulink.Signal objects with global storage class] 設定で、RTI CAN MultiMessage Blockset および RTI LIN MultiMessage Blockset の BusSystems グループも考慮されます。

Bus Selector の取り扱いの変更

MATLAB R2014a で、Simulink Bus Selector ブロックが仮想化されています。dSPACE Release 2014-B と組み合わせて使用する際に、Bus Selector ブロックの入力ポートが *非仮想バス* に接続されている場合、このブロックやこのブロックに接続された他の仮想ブロックの出力ポートは、変数記述ファイルに生成されません。

バスの信号にアクセスするには、以下を実行します。

- バスから信号に直接アクセスします。
または
- 非仮想ブロックを信号に接続します (たとえば、係数 1 の Gain ブロック)。このブロックの出力ポートは、変数記述ファイルで利用することができます。

Bus Selector ブロックの入力ポートが仮想バスを介して接続されている場合、特別な方法を利用しなくても、出力ポートは変数記述ファイルに生成されます。

Simulink Coder で生成される コードの変更

MATLAB R2014a の使用 MATLAB/Simulink R2014a では、Simulink Coder によるコード生成が変更されています。そのため、RTI での変数記述ファイル (TRC ファイル) の生成も変更されています。

ビルドプロセスの開始前に、`revertInlineParametersOffToR2013b` コマンドを使用する必要があります。このコマンドを使用すると、MATLAB R2014a での Simulink Coder の動作や dSPACE TRC ファイルの生成を、MATLAB R2013b 以前と同じにすることができます。

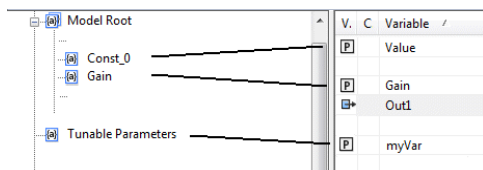
`revertInlineParametersOffToR2013b` コマンドは、Simulink Coder 製品の一部として MATLAB R2014a に含まれています。

詳細については、Simulink Coder R2014a のリリースノート ([*Simpler behavior for tuning all parameters and support for referenced models*]) を参照してください。

<http://www.mathworks.de/de/help/rtw/release-notes.html>
(MathWorks アカウントが必要です)。

MATLAB R2014b の使用 [Inline Parameters]最適化オプションをオフにした場合にかぎり、Simulink Coder の変更が生成される TRC ファイル内のエントリに影響を及ぼします。1 つまたは複数のブロックパラメータによって参照されるワークスペース変数は、モデル内でグローバルパラメータとして扱われ、ブロックのローカルパラメータとしては扱われません。これらは、生成される TRC ファイルの Tunable Parameters グループに格納されます。これらのグローバルパラメータの値を変更すると、変更したグローバルパラメータを参照するブロックパラメータも変更されます。シミュレーション結果が変わります。

例：



このモデルでは、Const_0 ブロックパラメータと Gain ブロックパラメータをワークスペース変数 `myVar` を使用して構成しています。

[Optimization]ページの[Inline parameters]オプションに関係なく、このワークスペース変数は変数記述ファイルに生成されます。これらの変数は、Const_0/Value と Gain/Gain を計器に接続することで、ControlDesk Next Generation で以前と同様に使用できますが、内部的にこれらのパ

ラメータ値は `myVar` の値に依存しています。パラメータのいずれかを変更すると、他のすべてのパラメータも変更されます。

次の場合、ワークスペース変数はローカルブロックパラメータとして生成されます。

- パラメータが構造体または構造体項目である場合 (例:
`myStruct.Value`)
- パラメータが式または関数内で使用される場合 (例:`myValue + 1` または `sin(myValue)`)

マスクパラメータへの参照を使用する場合、ローカルブロックパラメータも生成されます。このマスクパラメータそのものは、変数記述ファイルには生成されません。

DS1006 リアルタイムアプリケーションの移行

DS1006 GNU C/C++ Compiler がアップデートされたため、リアルタイムアプリケーションを再ビルドすることをお勧めします。RTLib 機能を参照するオブジェクトおよびライブラリに互換性がある一方で、標準の C/C++ 機能を参照するオブジェクトおよびライブラリは、アップデートされた C および C++ ライブラリと競合する可能性があります。

RTI Bypass Blockset

項目の一覧

本章の内容

RTI Bypass Blockset 3.3 の新機能	129
RTI Bypass Blockset 3.3 への移行	130

RTI Bypass Blockset 3.3 の新機能

RTI Bypass Blockset

MATLAB x64 のサポート RTI Bypass Blockset に、MATLAB 64 ビットバージョンのサポートが追加されています。

FlexRay フレームの FlexRay バッファへの割り当ての改善 RTI Bypass Blockset で、FlexRay フレームの FlexRay バッファへのマッピングに関して以下の改善が加えられました。

- Setup ブロックの[Buffers Configuration]ページに、FlexRay バッファのマッピング情報が表示されます。ダイアログを開くと、各 LPDU に割り当て可能なバッファが表示されます。バッファの割り当てができない場合は、その理由に関する情報が表示されます。

このマッピング情報の表示は、特定の LPDU に割り当てることができない原因を調べるのに役立ちます。

- XCP の異なる LPDU に同じフレーム名を使用することができます。

「Buffers Configuration Page (RTIBYPASS_SETUP_BLx for XCP on FlexRay)」( 『RTI Bypass Blockset Reference』)を参照してください。

ECU アプリケーションのバイナリコンテンツのラベル設定 内部バイパスコードをビルドする場合、拡張データベースファイル(出力 A2L ファイル)とマージされた ECU アプリケーション(出力 ECU アプリケーション)

が生成されます。RTI Bypass Blockset では、ECU アプリケーションの内容に関するラベルを指定し、エクスポートされるバイナリコードに統合することができます。あらかじめ定義されたマクロを使用して、ラベルを動的に定義できます。たとえば、製品、ユーザ、およびファイルの作成日時を追加することができます。「Build Page (RTIBYPASS_SETUP_BLx for INTERNAL)」([🔗『RTI Bypass Blockset Reference』](#))を参照してください。

メモリアドレスと使用するラベルを A2L ファイルで指定することができます。これに関連して、IF_DATA dSPACE_INTERNAL_BYPASS エントリに関連する AML ファイル dSPACE_INTERNAL_Bypass_v1_3_0.aml が新たに提供されています。「内部バイパス処理用のインターフェース記述データ」([🔗『Interface Description Data Reference』](#))を参照してください。

XCP 1.2 のサポート RTI Bypass Blockset は、XCP 1.2 規格に基づいた XCP 固有の IF_DATA エントリを含む A2L ファイルをサポートしています。サポートされる機能は、XCP 1.1 の場合と同じです。

RTI Bypass Blockset の MATLAB API

RTI Bypass Blockset の機能拡張のサポート RTI Bypass Blockset の MATLAB API は、RTI Bypass Blockset の機能拡張をサポートしています。

[🔗『RTI Bypass Blockset MATLAB API Reference』](#)を参照してください。

RTI Bypass Blockset 3.3 への移行

以前の RTI Bypass Blockset バージョン 3.x および 2.x のモデルの使用

最新のリリースには、以前のブロックセットバージョン 3.x および 2.x と互換性のある RTI Bypass Blockset 3.3 が含まれています。ただし、いくつかの注意事項があります。

- **RTI Bypass Blockset 2.5 以前のモデルの使用:** 以前の RTI Bypass Blockset バージョンと比較すると、データ管理が変更されています。RTI Bypass Blockset 2.5 以前でビルドした Simulink モデルを RTI Bypass Blockset 3.3 で開くと、古い Data Dictionary ファイル(ファイル名の拡張子 .dd)が自動的に削除されます。その後、[OK]をクリックして Setup ブロックダイアログを開いて閉じるか、または Read/Write/Upload/Download ブロックダイアログを開いて [Variables]ページの [Fill Variable Selector] ボタンをクリックするとすぐに、Setup ブロック内に格納されている情報を使用して新しい Data Dictionary ファイル (.vdb) が作成されます。

RTI Bypass Blockset 3.3 で保存したモデルを RTI Bypass Blockset 2.5 以前で使用する場合は、Setup ブロックで A2L ファイルを更新するか、または Read/Write/Upload/Download ブロックを開いて [Variables]ページの [Fill Variable Selector] ボタンをクリックするとすぐ

に、以前の RTI Bypass Blockset ブロックセットバージョンに必要なモデルの Data Dictionary ファイル(ファイル名の拡張子.dd)が作成されます。RTI Bypass Blockset 3.3 で作成された Data Dictionary ファイル(*.vdb)は、ディスク上にそのまま残ります。

RTI Bypass Blockset で Data Dictionary を再作成するには、Setup ブロックで指定されたデータベースファイルが指定された場所からアクセス可能で、これらのファイルが変更されていないことが必須条件となります。

- *RTI Bypass Blockset 2.6 から RTI Bypass Blockset 3.2 までのモデルの使用* RTI Bypass Blockset 2.6 から RTI Bypass Blockset 3.2 まででビルドされた Simulink モデルを RTI Bypass Blockset 3.3 で開くと、古い Data Dictionary ファイルが新しい Data Dictionary ファイルに置き換えられます。ただし、新しい Data Dictionary ファイルは以前のバージョンの RTI Bypass Blockset では使用することができません。RTI Bypass Blockset 2.6 から RTI Bypass Blockset 3.2 まででビルドされたモデルを使用するには、Setup ブロックで指定されているデータベースファイル(A2L ファイル)を再インポートして、以前のバージョンの RTI Bypass Blockset で適切なデータベースを作成する必要があります。

RTI CAN Blockset

RTI CAN Blockset 3.3 の新機能

新しくサポートされるプラットフォーム

RTI CAN Blockset は、2 つの CAN チャンネルを利用できる MicroLabBox をサポートしています。

このブロックセットは、ISO11898-6 トランシーバタイプを使用する場合に、MicroLabBox に対して CAN パーシャルネットワーキングをサポートします。

「Partial Networking Page (RTICAN CONTROLLER SETUP)」([📄](#))『RTI CAN Reference』を参照してください。

RTI CAN MultiMessage Blockset

項目の一覧

本章の内容

RTI CAN MultiMessage Blockset 4.0 の新機能	135
RTI CAN MultiMessage Blockset 4.0 への移行	136

RTI CAN MultiMessage Blockset 4.0 の新機能

新しくサポートされるプラットフォーム

RTI CAN MultiMessage Blockset は、MicroLabBox をサポートしていません。MicroLabBox では、2 つの CAN チャンネルを利用することができません。

CAN パーシャルネットワーキング このブロックセットは、ISO11898-6 トランシーバタイプを使用する場合に、MicroLabBox に対して CAN パーシャルネットワーキングをサポートします。「Partial Networking Page (RTICANMM ControllerSetup)」([📄](#)『RTI CAN MultiMessage Blockset Reference』)を参照してください。

FIBEX 4.1 のサポート

RTI CAN MultiMessage Blockset では、データベースファイルとして FIBEX 4.1 ファイルをサポートしています。

「General Settings Page (RTICANMM MainBlock)」([📄](#)『RTI CAN MultiMessage Blockset Reference』)を参照してください。

CAN FD プロトコルのサポート

RTI CAN MultiMessage Blockset では、CAN FD (*CAN with Flexible Data Rate*) プロトコルをサポートしています。従来の CAN プロトコルと比較して、CAN FD はシリアル通信の帯域幅が拡張されています。この改良は、次の 2 点に基づいています。

- CAN メッセージのデータフィールドの拡張 (最大 64 バイト)
- ビットレートの向上 (オプション)

アプリケーションでは、これらの 1 つまたは両方を利用することができます。

「Basics on Working with CAN FD」([📄 『RTI CAN MultiMessage Blockset Reference』](#))を参照してください。

『RTI CAN MultiMessage Blockset チュートリアル』

『RTI CAN MultiMessage Blockset チュートリアル』に、チェックサムアルゴリズムでのエンドツーエンドな通信保護 (E2E 保護) パラメータの使用に関するレッスンが追加されています。データベースファイルから E2E 保護パラメータを使用し、リアルタイムアプリケーションで E2E 保護パラメータに基づいてチェックサムアルゴリズムを実装する方法について説明します。

「Lesson 15 (Advanced): Using Checksums Based on E2E Protection Parameters」([📄 『RTI CAN MultiMessage Blockset Tutorial』](#))を参照してください。

RTI CAN MultiMessage Blockset 4.0 への移行

RTI CAN MultiMessage Blockset の以前のバージョンで作成したモデルの使用

RTI CAN MultiMessage Blockset の以前のバージョンで作成されたモデルを再利用するには、CAN の設定に変更を加える前に、すべての RTICANMM ブロックの S-function を更新して保存する必要があります。

モデル内のすべての RTICANMM ブロックに対して新しい S-function を一度に作成するには、モデルを開いた後で次のいずれかを実行します。

- MATLAB コマンドウィンドウに `rtimmsu_update('System', gcs)` と入力します。

このコマンドおよびオプションの詳細を確認するには、MATLAB コマンドウィンドウに `help rtimmsu_update` と入力します。

- 「RTICANMM GeneralSetup」ブロックの [Options] メニューから [Create S-Function for all CAN Blocks] コマンドを選択します。

詳細については、「Limitations with RTICANMM」([📄 『RTI CAN MultiMessage Blockset Reference』](#))を参照してください。

**バージョン 4.0 より前の RTI
CAN MultiMessage Blockset
で生成されたコードを使用した
場合のコンパイラメッセージ**

バージョン 4.0 より前の RTI CAN MultiMessage Blockset で生成されたコードを使用すると、データタイプの変更によるシミュレーションモデルのビルドプロセス中に、<<argument of type "can_tp1_canChannel *" is incompatible with parameter of type "DsTCanCh">>というフレーズを含む複数のコンパイラ警告メッセージが表示されます。これらの警告は無視してかまいません。最新バージョンのブロックセットを使用して RTICANMM コードを再生成すると、表示されなくなります。

**既存のチェックサムアルゴリズム
の使用**

CAN メッセージを含むアプリケーション用に本来開発されたチェックサムアルゴリズムは、CAN FD メッセージを含むアプリケーションで再利用することはできません。これは、CAN FD に新しいメッセージタイプが含まれているか、データフィールドが長いからです。既存のチェックサムアルゴリズムは、標準的な CAN メッセージのみ含むアプリケーションでは引き続き使用することができます。CAN FD アプリケーションの場合は、チェックサムアルゴリズムを適合させる必要があります。

RTI Electric Motor Control Blockset

RTI Electric Motor Control Blockset 1.0 の機能

新しいブロックセット

RTI Electric Motor Control Blockset は、最新式のモーターのコントローラを実装する特殊機能を提供する新しいブロックセットです。

以下のためのブロックを提供します。

- マルチチャンネルの PWM 信号生成
- ブロック通信の PWM 信号生成
- エンコーダ
- ホールセンサ

このブロックセットは MicroLabBox をサポートします。

詳細については、[📖](#) 『RTI Electric Motor Control Blockset Reference』を参照してください。

RTI Ethernet Blockset

RTI Ethernet Blockset 1.1 の新機能

新しいプラットフォームのサポート

RTI Ethernet Blockset は MicroLabBox をサポートします。

設定可能なスイッチ動作

DS1007 PPC Processor Board および MicroLabBox には 3 つの Ethernet コネクタがあり、内部で Ethernet スイッチに接続されます。これらのボードのブラウザベースの設定／管理ツールに新しい[Switch Configuration]ページが追加され、スイッチの動作を指定することができます。これにより、Ethernet コネクタを I/O デバイスに使用するか、ホスト PC の通信に使用するかを選択することができます。

詳細については、「Basics on Browser-Based Configuration and Management Tool」([🔗](#)『DS1007 Hardware Installation and Configuration Guide』)を参照してください。

入手可能な RTLib 関数のマニュアル

RTI Ethernet Blockset で使用される RTLib 関数が、『DSIOETH RTLib Reference』に文書化されました。これにより、DS1007 PPC Processor Board または MicroLabBox の I/O Ethernet インターフェースを使用して通信をハンドコーディングすることができます。詳細については、[🔗](#)『DSIOETH RTLib Reference』を参照してください。

RTI FPGA Programming Blockset

項目の一覧

本章の内容

RTI FPGA Programming Blockset 2.8 の新機能	143
RTI FPGA Programming Blockset 2.8 への移行	145

RTI FPGA Programming Blockset 2.8 の新機能

Xilinx®のサポートの拡張

RTI FPGA Programming Blockset で、Xilinx 設計ツールの以下の製品とバージョンがサポートされるようになりかした。

Xilinx 設計ツールのバージョン	オペレーティングシステム	MATLAB バージョン ¹⁾
13.4 ¹⁾	<ul style="list-style-type: none">■ Windows XP Professional SP3 (32 ビット版)■ Windows 7 Business、Ultimate および Enterprise SP1 (32 ビット版および 64 ビット版)	<ul style="list-style-type: none">■ MATLAB R2010bSP2²⁾■ MATLAB R2011a■ MATLAB R2011b
14.1 ¹⁾	<ul style="list-style-type: none">■ Windows XP Professional SP3 (32 ビット版)■ Windows 7 Business、Ultimate および Enterprise SP1 (32 ビット版および 64 ビット版)	<ul style="list-style-type: none">■ MATLAB R2011a■ MATLAB R2011b

¹⁾ 32 ビットバージョンのみ

²⁾ dSPACE にてテスト済みですが、Xilinx で正式にはサポートされていません。

一般的な機能強化

新しいプラットフォームのサポート RTI FPGA Programming Blockset で、MicroLabBox のフレームワークが提供されます。

MicroLabBox のプラットフォーム ID は DS1202 です。このボードは、シグナルコンディショニング機能を持つ必要な I/O コネクタも搭載しています。そのため、MicroLabBox のフレームワークは *DS1202 with onboard I/O* という名前が付けられています。

このフレームワークには次の機能があります。

- A/D 変換
- D/A 変換
- デジタル入出力
- シリアルインターフェース (RS232 および RS422/485)
- FPGA アプリケーションおよびプロセッサアプリケーションの状態情報
- ボードのプログラム可能 LED へのアクセス

チャンネル名の設定の拡張 FPGA_XDATA ブロックのチャンネル名の設定を FPGA_IO_READ_BLx および FPGA_IO_WRITE_BLx ブロックでも使用することができます。この設定を使用すると、生成されたチャンネル名の代わりにユーザ固有のチャンネル名を入力することができます。より分かりやすい名前や短い名前を指定することができます。

詳細については、 『RTI FPGA Programming Blockset - FPGA Interface Reference』を参照してください。

ボード名の変更 DS2655 FPGA BaseModule という名称が DS2655 FPGA Base Board に変更されています。

ConfigurationDesk カスタム関数処理の単純化 DS2655 FPGA Base Board および 1 つ以上の DS2655 Multi-I/O Module を搭載する SCALEXIO システム用にビルドした FPGA アプリケーションを、以前より簡単に ConfigurationDesk にインポートできるようになりました。詳細については、「ConfigurationDesk – Implementation」(76 ページ)を参照してください。

関連トピック

基礎

- 「RTI FPGA Programming Blockset 2.8 への移行」(145 ページ)

RTI FPGA Programming Blockset 2.8 への移行

目的 既存のモデルの移行方法は、使用するブロックセットのバージョンによって異なります。

RTI FPGA Programming Blockset 1.0 から 2.8 への移行

RTI FPGA Programming Blockset 1.0 (dSPACE Release 6.4 で提供) は完全に実装されたものではなかったため、これを使用して実装したモデルは手動で移行する必要があります。最新の dSPACE RTI 環境に準拠したモデルをモデル化、ビルド、および実行するために、RTI FPGA Programming Blockset の各ブロックを新しいブロックに置き換える必要があります。



スクリプトインターフェースの更新機能は、RTI FPGA Programming Blockset 1.0 をサポートしていません。

RTI FPGA Programming Blockset 1.1 以降から 2.8 への移行

バージョン 1.1 以降の RTI FPGA Programming Blockset を使用して FPGA アプリケーションを実装した場合、これを RTI FPGA Programming Blockset 2.8 で使用するには、FPGA フレームワークを更新する必要があります。この場合に、スクリプトインターフェースを使用することができます(「スクリプトインターフェースを使用した FPGA フレームワークの更新」(145 ページ)を参照)。

MATLAB R2008b 以前のバージョンから MATLAB R2011b 以降のバージョンに更新した場合は、フレームワークも更新する必要があります。

スクリプトインターフェースを使用した FPGA フレームワークの更新



移行を開始する前に、モデルのバックアップを作成することをお勧めします。

スクリプトインターフェースには、フレームワークを更新するための `FPGAFrameworkUpdate` メソッドが用意されています。ブロックパラメータをその初期値に設定するか、変更せずにそのまま使用するかを指定することができます。

ブロックパラメータの値を変更せずに FPGA フレームワークを更新する場合

```
rtifpga_scriptinterface('FPGAFrameworkUpdate',  
<SimulinkHandle>)
```

このスクリプトでは、Simulink Handle で指定されているモデル/サブシステム内のすべてのサブシステムが処理されます。最新のフレームワークバージョンにアップデートしても、ブロックのパラメータは変更されません。

例: 次のスクリプトは、*MyProcModel* という名前のプロセッサモデルの中にあるすべての FPGA サブシステムに対して FPGA フレームワークを更新します。ブロックパラメータの指定された値は変更されません。

```
ProcModelHandle = get_param('MyProcModel','handle')
rtifpga_scriptinterface('FPGAFrameworkUpdate',
    ProcModelHandle)
```

FPGA フレームワークを更新して、ブロックパラメータの値をその初期値にリセットする場合

```
rtifpga_scriptinterface('FPGAFrameworkUpdate',
    <SimulinkHandle>, 'ReInit')
```

このスクリプトでは、Simulink Handle で指定されているモデル/サブシステム内のすべてのサブシステムが処理されます。最新のフレームワークバージョンに更新すると、ブロックのパラメータはそれぞれの初期値にリセットされます。

```
ProcModelHandle = get_param('MyProcModel','handle')
rtifpga_scriptinterface('FPGAFrameworkUpdate',
    ProcModelHandle, 'ReInit')
```

dSPACE Release 2014-B と互換性のない ConfigurationDesk カスタムファンクション



DS2655 FPGA Base Board および DS2655M1 I/O Module を搭載した SCALEXIO システムに関連

dSPACE Release 2013-A の RTI FPGA Programming Blockset 2.5 を使用して生成されたカスタムファンクションと、カスタムファンクションを含むリアルタイムアプリケーション (.RTA) は、dSPACE Release 2014-B と互換性がありません。dSPACE Release 2014-B の RTI FPGA Programming Blockset 2.8 を使用して FPGA モデルを再ビルドして、使用可能なカスタムファンクションを作成する必要があります。

RTI LIN MultiMessage Blockset

項目の一覧

本章の内容

RTI LIN MultiMessage Blockset 2.4 の新機能	147
RTI LIN MultiMessage Blockset 2.4 への移行	147

RTI LIN MultiMessage Blockset 2.4 の新機能

FIBEX 4.1 のサポート

RTI LIN MultiMessage Blockset では、データベースファイルとして FIBEX 4.1 ファイルをサポートしています。

「General Settings Page (RTILINMM MainSetup)」( 『RTI LIN MultiMessage Blockset Reference』)を参照してください。

RTI LIN MultiMessage Blockset 2.4 への移行

RTI LIN MultiMessage Blockset の以前のバージョンで作成したモデルの使用

RTI LIN MultiMessage Blockset の以前のバージョンで作成されたモデルを再利用するには、LIN の設定に変更を加える前に、すべての RTILINMM ブロックの S-function を更新して保存する必要があります。

モデル内のすべての RTILINMM ブロックに対して新しい S-function を一度に作成するには、モデルを開いた後で次のいずれかを実行します。

- MATLAB コマンドウィンドウに `rtimmsu_update('System', gcs)` と入力します。

このコマンドおよびオプションの詳細を確認するには、MATLAB コマンドウィンドウに `help rtimmsu_update` と入力します。

- 「RTILINMM GeneralSetup」ブロックの [Options] メニューから [Create S-Function for all LIN Blocks] コマンドを選択します。

詳細については、「Limitations of RTI LIN MultiMessage Blockset」
( 『RTI LIN MultiMessage Blockset Reference』) を参照してください。

RTI USB Flight Recorder Blockset

RTI USB Flight Recorder Blockset 1.2 の新機能

プラットフォームサポートの強化

RTI USB Flight Recorder Blockset は MicroLabBox をサポートします。
詳細については、📖 『RTI USB Flight Recorder Blockset Reference』を参照してください。

RTLib 関数の別冊マニュアル

USB フライトレコーディング機能のハンドコーディングに使用する RTLib 関数が、『*USB Flight Recorder RTLib Reference*』に文書化されました。これまででは、サポートされるプラットフォームの『RTLib References』に統合されていました。
詳細については、📖 『USB Flight Recorder RTLib Reference』を参照してください。

SCALEXIO Firmware

SCALEXIO Firmware 3.1 の新機能

SCALEXIO プロセッサユニット

SCALEXIO Firmware は、新しいリアルタイム PC をサポートしています。詳細については、「SCALEXIO Real-Time PC Data Sheet」([PDF](#)『SCALEXIO Hardware Installation and Configuration』)を参照してください。

I/O ファンクション

SENT In、Lambda DCR、Lambda NCCR のファンクションが改善されています。「ConfigurationDesk 5.2 (Implementation Version) の新機能」(76 ページ)を参照してください。

SystemDesk

項目の一覧

本章の内容

SystemDesk 4.3 の新機能	154
SystemDesk 4.3 への移行	158

SystemDesk 4.3 の新機能

項目の一覧

本章の内容

新しい一般機能	154
ソフトウェアアーキテクチャのモデリング	155
システムのモデリング	156
ECU コンフィギュレーション	156
エレメントの妥当性確認	157

新しい一般機能

SystemDesk 4.3 には、次の一般機能が新たに追加されています。

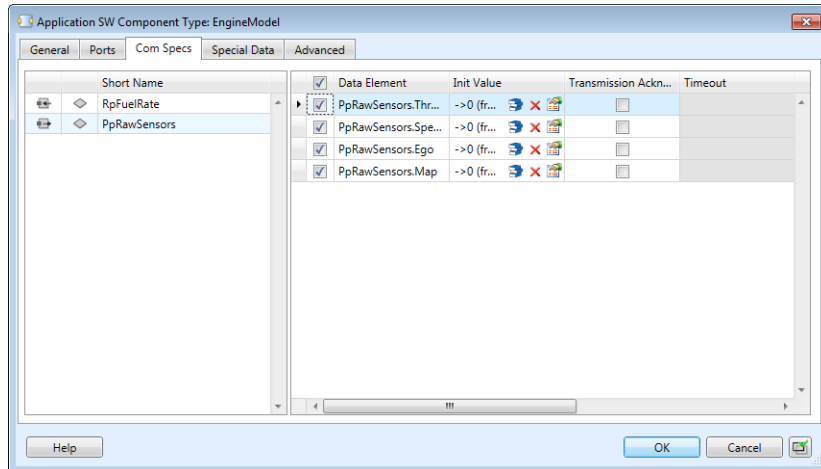
SystemDesk 4.3 でサポートされる AUTOSAR リリース

SystemDesk 4.3 では、AUTOSAR 4.1.3、4.1.2、4.1.1、4.0.3、4.0.2 に準拠したソフトウェアおよびシステムアーキテクチャのモデリングをサポートします。

ソフトウェアアーキテクチャのモデリング

通信仕様の処理の改良

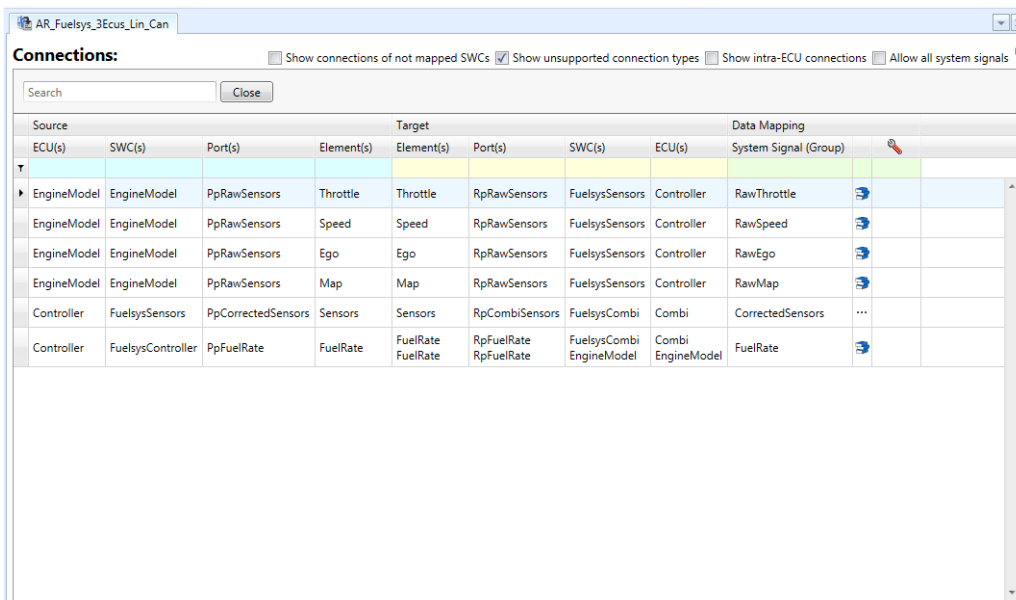
SystemDesk では、ソフトウェアコンポーネントのすべてのポートを即座に表示して編集することができます。そのため、複数のデータエレメントとポートの初期値を同時に編集することができます。



システムのモデリング

ECU 間通信とネットワーク通信 のマッピングの改良

ネットワーク通信を表すシステム信号に ECU 間通信を割り当てることができる、便利な Data Mapping Editor が SystemDesk に新しく追加されました。下図に、デモプロジェクトのエディタを示します。



ECU コンフィギュレーション

LIN ネットワーククラスタを使用 するシステムの V-ECU イン プリメンテーションと V-ECU の生 成

SystemDesk で、LIN ネットワーククラスタを使用するシステムの V-ECU インプリメンテーションと V-ECU の生成がサポートされるようになりました。

LIN 通信エレメントを含む LDF ファイルまたは AUTOSAR ARXML ファイルをインポートすることができます。

V-ECU インプリメンテーション SystemDesk では、*LinIf*、*PduR*、*COM* モジュールに、ベーシックソフトウェアモジュールのコンフィギュレーションを生成することができます。また、SystemDesk では、*PduR* および *COM* に C コードを生成することができます。

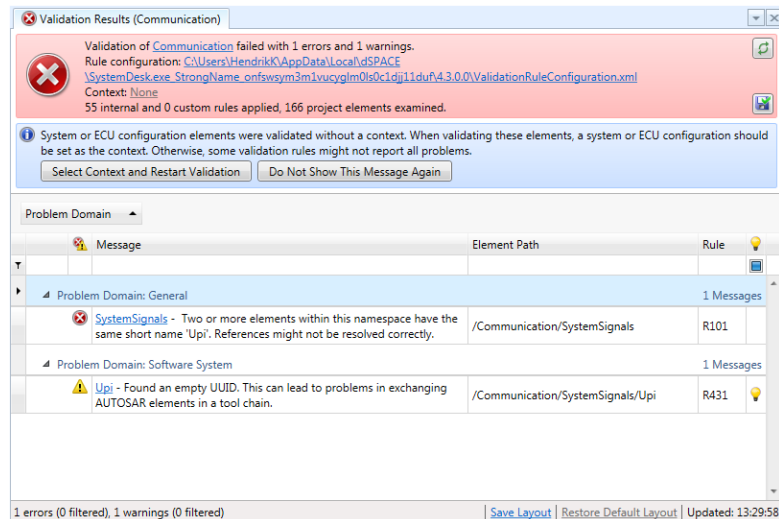
V-ECU SystemDesk では、VEOS シミュレーション用の V-ECU を生成することができます。

エレメントの妥当性確認

妥当性確認結果

SystemDesk では、妥当性確認結果が下図のように表示されるようになりました。

妥当性確認結果は明確に可視化されます。妥当性確認の繰り返しや、妥当性確認結果のファイルへの保存を行うことができます。これにより、妥当性確認結果のやり取りや、プロジェクトの資料としての保存が可能です。



Validation Results (Communication)

Validation of **Communication** failed with 1 errors and 1 warnings.
 Rule configuration: C:\Users\HendrikK\AppData\Local\dspace\SystemDesk.exe_StrongName_onfswym3m1vucyglm0ls0c1djl1duf4.3.0.0\ValidationRuleConfiguration.xml
 Context: None
 55 internal and 0 custom rules applied. 166 project elements examined.

System or ECU configuration elements were validated without a context. When validating these elements, a system or ECU configuration should be set as the context. Otherwise, some validation rules might not report all problems.

Select Context and Restart Validation | Do Not Show This Message Again

Problem Domain	Message	Element Path	Rule
Problem Domain: General			1 Messages
	SystemSignals - Two or more elements within this namespace have the same short name 'Upi'. References might not be resolved correctly.	/Communication/SystemSignals	R101
Problem Domain: Software System			1 Messages
	Upi - Found an empty UUID. This can lead to problems in exchanging AUTOSAR elements in a tool chain.	/Communication/SystemSignals/Upi	R431

1 errors (0 filtered), 1 warnings (0 filtered) | Save Layout | Restore Default Layout | Updated: 13:29:58

SystemDesk 4.3 への移行

SystemDesk 4.3 への移行

SystemDesk 4.3 では、SystemDesk 4.1 および 4.2 の SDP プロジェクトファイルはロード時に自動的に移行されます。



SystemDesk 4.1 または 4.2 の最新のパッチをインストールすることをお勧めします。その後、移行する SDP プロジェクトファイルを保存してから、SystemDesk 4.3 で開きます。

TargetLink

項目の一覧

本章の内容

TargetLink 4.0 および TargetLink Data Dictionary 4.0 の新機能	160
TargetLink 4.0 および TargetLink Data Dictionary 4.0 への移行	187
TargetLink の今後のバージョンでの変更予定	222

TargetLink 4.0 および TargetLink Data Dictionary 4.0 の新機能

項目の一覧

本章の内容

Simulink または Stateflow でのモデリング	160
コード生成のコア機能	167
Data Dictionary とデータ管理	169
AUTOSAR	175
テストのサポート	177
Code Generator オプション	180
ツールチェーンの統合	182
その他	183
API コマンド	185

Simulink または Stateflow でのモデリング

項目の一覧

本章の内容

行列信号のサポート	161
新しくサポートされる Simulink ブロック	161
バスサポートの向上	162
動的ルックアップテーブル	162
TargetLink のシミュレーションフレームの改善	163
Scaling-Invariant システムの改良	163
ブロックのプロパティの追加サポート	164
ファンクションサブシステムのシグネチャの一元的な指定	165

行列信号のサポート

2-D 行列信号

TargetLink では、2-D 信号とパラメータのコード生成をサポートし(→「Matrix signal」)、MIL/SIL/PIL シミュレーションモードでシミュレーションを実行することができます。



RTOS コード生成モードの場合は、2-D 信号を使用してタスクの境界を通過することはできません。タスク内では、2-D 信号はサポートされます。

関連ドキュメント

- 「Introduction to Working With Matrix Signals」(📖『TargetLink Preparation and Simulation Guide』)
- 「Code Pattern for Vectors and Matrices」(📖『TargetLink Preparation and Simulation Guide』)
- 「Examples of Working With Matrix Signals」(📖『TargetLink Preparation and Simulation Guide』)
- 「Blocks not supporting matrix signals」(📖『TargetLink Orientation and Overview Guide』)

新しくサポートされる Simulink ブロック

TargetLink で、次の Simulink ブロックのサポートが追加されています。

- Matrix Concatenate
- Permute Dimensions
- Reshape

関連ドキュメント

- 「Supported Simulink Blocks」(📖『TargetLink Block and Object Reference』)
- 「Working With Matrix Signals」(📖『TargetLink Preparation and Simulation Guide』)

バスサポートの向上

あらかじめ設定された構造体へのバス全体のマッピング

このバージョンの TargetLink では、TargetLink Data Dictionary (DD) にあらかじめ設定された構造体タイプを作成し、TargetLink の BusInport および BusOutport ブロックで、バス全体をこの DD 構造体タイプに割り当てることができます。これは、バスが多数のバスエレメントで構成されている場合や、1 つまたは複数のモデル内でタイプや変数が複数回使用される場合に特に役立ちます。

また、Switch、Multiport Switch、Merge、Unit Delay ブロックなどのバス対応ブロックで、DD 構造体タイプや構造体変数を参照することもできます。

関連ドキュメント

- 「Basics on the Representation of Buses in the Production Code」
([📖](#) 『TargetLink Customization and Optimization Guide』)
- 「Mapping Entire Buses to Explicit Structure Variables and Type Definitions in the Code」 ([📖](#) 『TargetLink Customization and Optimization Guide』)

動的ルックアップテーブル

シミュレーション時のテーブルデータの変更

以下の 3 つの TargetLink ルックアップテーブルブロックでは、ブロックのダイアログでテーブルデータを指定できるのみでなく、ブロックのオプションのテーブルデータ入力ポートを使用してブロックにテーブルデータ (1 次元または 2 次元) を供給することができます。後者の場合は、シミュレーション時や生成コードのランタイム中に、テーブルデータを変更することができます。

- Direct Look-Up Table (n-D)
- Prelookup
- Interpolation Using Prelookup

関連ドキュメント

- 「Principles on Look-up Tables」 ([📖](#) 『TargetLink Preparation and Simulation Guide』)
- 「How to Prepare Dynamic Look-Up Table Specification」
([📖](#) 『TargetLink Preparation and Simulation Guide』)

TargetLink のシミュレーションフレームの改善

SIL または PIL シミュレーションモードに切り替える際に、TargetLink で Commented ブロックパラメータを on に設定して、MIL サブシステムを無効することができます。

これには、次のような利点があります。

- SIL または PIL シミュレーションモードでのモデル初期化速度の改善
- SIL または PIL シミュレーションモードへの切り替え時の速度の改善

互換性に関する考慮事項

- Simulink では、コメント付きブロックの整合性チェックが実行されません。
- デフォルトでは、Simulink の `find_system` API 関数の検索対象にコメント付きブロックは含まれません。
- Simulink では、Commented ブロックプロパティを直接または間接的に変更する OpenFcn コールバックの使用は許可されていません。詳細については、「移行に関するその他の注意点」(212 ページ)を参照してください。

解決策 考えられる解決策として、次の 2 つがあります。

- シミュレーションモードが最初に MIL に切り替わるようにユーザスクリプトを調整する。
- TargetLink API 関数を使用するようにユーザスクリプトを調整する。
 - 「`t1_get_blocks`」
 - 「`t1_get_sfojects`」
 - 「`t1_find`」

Simulink モデルウインドウの TargetLink メニューの新しい「Activate MIL」エントリを使用すると、MIL シミュレーションモードへの切り替えを簡単に行うことができます。

Scaling-Invariant システムの改良

スケーリングの継承の改良

Scaling-Invariant サブシステムでは、出力ポートのスケーリングに影響を及ぼす入力ポートを任意で指定することができます。ポートマッピングはスケーリング伝搬関数に指定することができます。コードジェネレータでは、(新しい)

`UtilizeExplicitDependenciesForScalingInvariantSystems` コードジェ

ネレータオプションが有効な場合に、コード生成時にこのマッピングを考慮します。この機能により、Scaling-Invariant システムを含むループの場合にスケールリング伝搬を向上させることができます。個々の入力に関する出力スケールリングの依存関係が考慮され、スケールリング伝搬時にループを解決できるようになりました。

関連ドキュメント

- 「Details on the Scaling Propagation Function」 ([📖 『TargetLink Customization and Optimization Guide』](#))

システムハンドルへのアクセス

スケールリング伝搬回数により、Scaling-Invariant システムのインスタンスのハンドルへのアクセスも可能になりました。これは特に、スケールリングが入力だけでなく、マスク変数で指定されているような、インスタンス固有のブロックデータにも依存する場合に役に立ちます。

ブロックのプロパティの追加サポート

Assignment ブロックの追加オプション

TargetLink の Assignment ブロックで、[Starting index (dialog)]オプションと[Starting index (port)]オプションがサポートされます。

関連ドキュメント

- 「Output Page (Assignment Block)」 ([📖 『TargetLink Block and Object Reference』](#))

Selector ブロックの追加オプション

TargetLink で、Simulink Selector ブロックの[Starting index (dialog)]オプションと[Starting index (port)]オプションがサポートされます。

Sink ブロックのプロットの改善

TargetLink の Sink ブロックで、すべてのプロットチャンネルにまとめて対応する汎用スイッチの代わりに、個別のプロットチャンネルの指定がサポートされるようになりました。

関連ドキュメント

- 「Logging Page (Sink Block)」 ([📖 『TargetLink Block and Object Reference』](#))

Stateflow データの Date プロパティ

TargetLink で、TargetLink ブロックと同様に Stateflow データの[Date]プロパティを利用することができます。このプロパティは、TargetLink Property Manager を使用して Stateflow オブジェクトを検査する際に、最新の変更を識別するのに役立ちます。

ファンクションサブシステムのシグネチャの一元的な指定

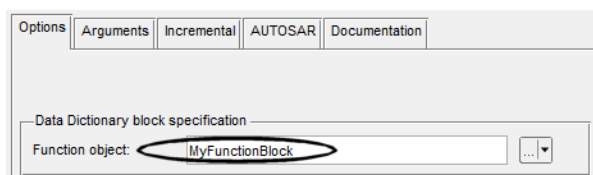
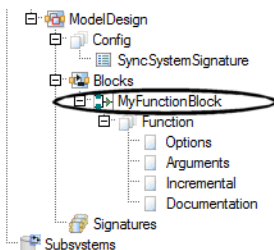
Data Dictionary でファンクションサブシステムのシグネチャを一元的に指定できるようになりました。また、これらの一元的な指定を元に、ファンクションサブシステムを生成することもできます。

本章の内容

DD Function Block オブジェクトでのファンクションブロックデータの指定	165
DD Signature オブジェクトでのサブシステムのシグネチャの指定	166

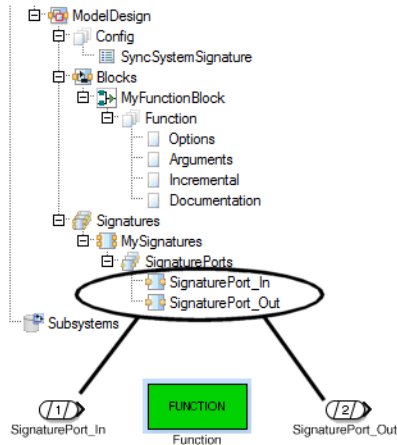
DD Function Block オブジェクトでのファンクションブロックデータの指定

Function Block オブジェクトを使用して、TargetLink Data Dictionary 内でファンクションブロックデータを指定できるようになりました。これにより、以下に示すように、TargetLink のファンクションブロックでこのオブジェクトを参照することができます。



DD Signature オブジェクトでのサブシステムのシグネチャの指定

Signature オブジェクトを使用して、サブシステムのシグネチャを指定することもできます。このためには、[Create Signature]コンテキストコマンドを使用し、サブシステムのポートを指定します。



最後に、[Synchronize System Signature]コンテキストコマンドを使用し、Data Dictionary Manager でモデル内にサブシステムとシグネチャを作成します。また、tlSyncSystemSignature API コマンドを使用することもできます。

詳細については、以下を参照してください。

- 「Basics on Centrally Specifying Function Subsystem Signatures」 (『TargetLink Customization and Optimization Guide』)
- 「How to Specify Function Block Data from Within the Data Dictionary」 (『TargetLink Customization and Optimization Guide』)
- 「How to Specify a Subsystem's Signature from within the Data Dictionary Manager」 (『TargetLink Customization and Optimization Guide』)

コード生成のコア機能

項目の一覧

本章の内容

MISRA-C への準拠	167
コード効率性の向上	168

MISRA-C への準拠

MISRA-C に準拠するため、TargetLink の固定小数点ライブラリに以下の改善を行いました。

- Accumulate レジスタの余分な初期化を FIR フィルタマクロから削除しました。
- コード内のさまざまな場所で、データタイプ制限に関する数値定数をグローバルマクロ (INT32MIN など) で置き換えました。
- 必要に応じて、呼び出しパラメータとして機能する定数にサフィックスを追加、または呼び出しパラメータとして機能する定数を想定されるタイプにキャストしました。
- 必要に応じて、初期値として機能する数値定数にサフィックスを追加しました。

MISRA-C に準拠するためのその他の改善点：

- マクロアクセス関数で、マクロ引数用の次のプレースホルダが括弧で囲まれるようになりました。
 - `_var`
 - `_value`
- Logical Operator ブロックに関して、論理式と算術式を一貫して区別するため、TargetLink では算術式の `^` ではなく論理式の `!=` が生成されます。
- 比較演算と論理演算に関して、非 Boolean 変数に結果を直接代入するのを回避できるようになりました。この動作は、コードジェネレータの新しい「AssignmentOfConditions」オプションを使用して制御することができます。

デフォルトで、Relational Operator および Logical Operator ブロックから、量産コードに `Output = <Condition>;` の代入が(最適化に関係なく)直接生成されるようになりました。ただし、

```
if (<Condition>) {
    Output = 1; /* or 0 */
} else {
    Output = 0; /* or 1, respectively */
}
```

は、Output が Boolean タイプ(かつ Optimization が有効)の場合にのみ、代入ステートメントに最適化されます。

- RDI マクロ定義に関して、TargetLink では、キャストを受け取る初期値の前後に括弧が付くようになりました。

コード効率性の向上

非スカラー信号のループ

TargetLink のループコード生成が改善されています。これには、次の内容が含まれます。

- 特定の構成でエレメントごとのベクトル割り当てや計算につながるより多くのブロックコードパターンで、割り当てや計算がループで実行されます。
これは、特に MinMax、Product、Sum、および Custom Code ブロックに当てはまります。
- 行列コードに対する for ループの生成(Stateflow の行列コードを含む)
- ループのマージの改善

不要な定義の削除

TargetLink で、条件制御フローから不要な以前の定義を削除できるようになりました。これは主に、Merge ブロックを駆動する Stateflow 出力に適用されます。

ベクトルおよびベクトルスライスの最適化

次元が「LoopUnrollThreshold」よりも小さい TargetLink のベクトルおよびベクトルスライスの最適化が変更され、次元が「LoopUnrollThreshold」以上の場合と同じように最初に最適化されるようになりました。これにより、エレメントごとの最適化に加えて、ベクトルでの最適化が可能になります。

暗黙補助変数の作成

TargetLink で、後続のブロックに対してより有利なブロックコードパターンになる場合に、ベクトルの暗黙補助変数を作成することができます。

インデックスでの定数の量み込み

TargetLink で、Assignment および Selector ブロックから生成されるインデックスで定数の量み込みがサポートされます。これにより、初期コードが改善されます。これはコード最適化には依存しません。

- TargetLink 3.5:
Sa2_Assignment[1 - 1] =...
- TargetLink 4.0:
Sa2_Assignment[0] =...

ベクトル変数の置き換え

TargetLink で、コードのセマンティクスで除去できないベクトル変数をスカラー変数で置き換える頻度が向上しました。これにより、生成されたコードの RAM/スタック使用量が少なくなります。

TargetLink 3.5 以前	TargetLink 4.0
<pre> Int16 vec[..]; loop() { if (...) { vec[i] = ... } else { vec[i] = ... } ... = vec[i] ... = vec[i] } Int16 vec[..]; loop() { fnc(&vec[i]); ... = vec[i]; } </pre>	<pre> Int16 scalar; loop() { if (...) { scalar = ... } else { scalar = ... } ... = scalar ... = scalar } Int16 scalar; loop() { fnc(&scalar); ... = scalar; } </pre>

Data Dictionary とデータ管理

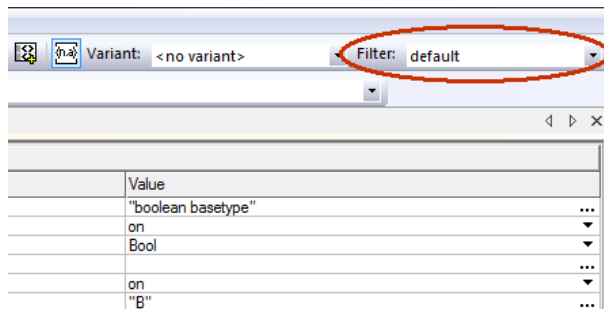
項目の一覧**本章の内容**

Data Dictionary の改善点	170
新しい DD MATLAB API コマンド	172

Data Dictionary の改善点

フィルタールールセットによる特定のオブジェクトやプロパティの非表示

DD Manager で XML ベースのフィルタールールセットを使用して、特定の DD オブジェクトやプロパティを非表示にすることができます。フィルタールールでは、データモデル内で定義されたオブジェクトやプロパティの表示／非表示を指定します。フィルタールールセットを使用することで、さまざまなチームメンバーに合わせて表示をカスタマイズすることができます。フィルタールールセットは、TargetLink Data Dictionary Manager の [Filter] リストまたは MATLAB API を使用して選択することができます。



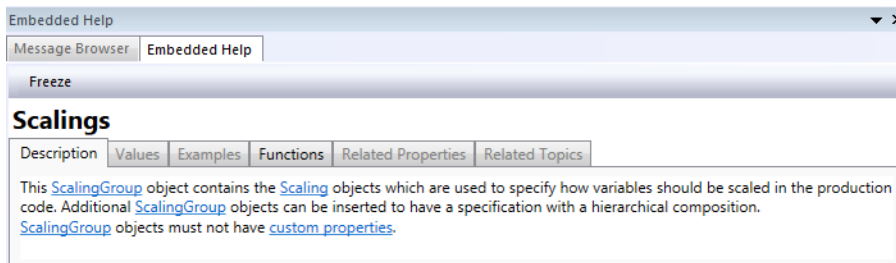
フィルタールールセットの生成方法の詳細については、「Basics on Filter Rule Sets for the Data Model」(『TargetLink Data Dictionary Basic Concepts Guide』)および「How to Create Filter Rule Sets」(『TargetLink Data Dictionary Basic Concepts Guide』)を参照してください。

フィルタールールセットの例は、次の URL にある TargetLink 製品サポートセンターからダウンロードすることができます。

http://www.dspace.com/ja/jpn/home/support/productcenters/targetlink_center.cfm

オブジェクトとプロパティに関する組み込みヘルプ

TargetLink Data Dictionary Manager で、DD オブジェクトやプロパティに関する [Embedded Help] ペインを利用することができます。選択したオブジェクトやプロパティに関する詳細な説明を参照することができます。このペインは、デフォルトで有効になっています。[Help] メニューで [Show Embedded Help] をクリックすると、組み込みヘルプの有効／無効を切り替えることができます。



詳細については、「How to Get Help on DD Objects and Properties」(🔗『TargetLink Data Dictionary Basic Concepts Guide』)および「Show Embedded Help」(🔗『TargetLink Data Dictionary Manager Reference』)を参照してください。

オブジェクトやプロパティに対するマージ機能の改善

Data Dictionary Manager のマージ機能にオプションが追加されました。新しい[Merge and Replace]コンテキストコマンドでは、オブジェクトやプロパティの置き換えがサポートされます。次の操作を行うことができます。

- Merge <left/right> without overwrite: 子オブジェクトおよびプロパティ値を上書きしないで DD オブジェクトをマージします。
- Merge <left/right>, and overwrite properties: DD 子オブジェクトを上書きしないで DD オブジェクトをマージします。プロパティ値は上書きされます。
- Merge <left/right>, and overwrite objects: DD オブジェクトがマージされます。子オブジェクトおよびプロパティ値を上書きします。両方の DD オブジェクトに存在する値および子オブジェクトが上書きされます。
- 新しいオプション: Replace <left/right>: DD オブジェクトが置き換えられます。ソースオブジェクトがターゲットオブジェクトにコピーされ、ターゲットオブジェクトが置き換えられます。

このコンテキストコマンドは、[DD Comparison]ペインでも利用することができます。

詳細については、「How to Merge and Replace DD Objects in DD Workspaces」(🔗『TargetLink Data Dictionary Basic Concepts Guide』)および「How to Merge and Replace DD Objects in the DD Comparison Pane」(🔗『TargetLink Data Dictionary Basic Concepts Guide』)を参照してください。

**最近使用したファイルリストの
拡張**

[Recent Files]リストに、モデル内から開いた DD プロジェクトファイルや MATLAB API を介して開いた DD プロジェクトファイルが表示されるようになりました。従来は、Data Dictionary Manager で開いた DD プロジェクトファイルのみが表示されていました。

完全なメッセージ文

Data Dictionary Manager の Message Browser および Custom Output View に、完全なメッセージが表示されるようになりました。従来は、メッセージのツールチップのみに完全なメッセージ文が含まれていました。

XML インポートの改善

Data Dictionary の XML インポートで、無効な XML ファイルへの対応力が向上しました。これにより、以前のバージョンの Data Dictionary からエクスポートした XML ファイルを、TargetLink 4.0 の Data Dictionary に確実にインポートすることができます。

関連トピック**基礎**

- 「Basics on Filter Rule Sets for the Data Model」([📖『TargetLink Data Dictionary Basic Concepts Guide』](#))

操作手順

- 「How to Create Filter Rule Sets」([📖『TargetLink Data Dictionary Basic Concepts Guide』](#))
- 「How to Merge and Replace DD Objects in DD Workspaces」([📖『TargetLink Data Dictionary Basic Concepts Guide』](#))
- 「How to Merge and Replace DD Objects in the DD Comparison Pane」([📖『TargetLink Data Dictionary Basic Concepts Guide』](#))

新しい DD MATLAB API コマンド

TargetLink で、以下の DD MATLAB API 関数を新たに利用することができます。詳細については、[📖『TargetLink Data Dictionary MATLAB API Reference』](#)を参照してください。

CountItems

```
[numOfObjects,numOfProperties] = dsdd('CountItems',<objectIdentifier>);
```

サブツリー内のオブジェクトとプロパティの数をカウントします。

CreateFilterRuleSet

```
bSuccess = dsdd('CreateFilterRuleSet',<filterRuleSet>);
```

フィルタルールセットを作成します。

DeleteFilterRuleSet

```
bSuccess = dsdd('DeleteFilterRuleSet',<filterRuleSet>);
```

フィルタルールセットを削除します。

DumpDataModelPaths	<pre>bSuccess = dsdd('DumpDataModelPaths', <file>);</pre> <p>すべてのデータモデルのパスをファイルに書き込みます。</p>
GetCurrentFilterRuleSet	<pre>filterRuleSet = dsdd('GetCurrentFilterRuleSet');</pre> <p>現在のフィルタールールセットを取得します。</p>
GetDataModelPath	<pre>dataModelPath = dsdd('GetDataModelPath', <objectIdentifier>[, <propertyName>]);</pre> <p>オブジェクトまたはプロパティのデータモデルのパスを取得します。</p>
GetDataModelPaths	<pre>dataModelPaths = dsdd('GetDataModelPath', <objectKind>);</pre> <p>オブジェクトの種類のすべてのデータモデルのパスを取得します。</p>
GetDataModelTag	<pre>dataModelTag = dsdd('GetDataModelTag', <DataModelPath>);</pre> <p>データモデルのタグを取得します。</p>
GetDefaultFilterRule	<pre>[bVisible, bValidPath] = dsdd('GetDefaultFilterRule', <DataModelPath>);</pre> <p>データモデル項目のデフォルトフィルタールールを取得します。</p>
GetFilterRule	<pre>[bVisible, bValidPath] = dsdd('GetFilterRule', <DataModelPath>);</pre> <p>データモデル項目のフィルタールールを取得します。</p>
GetFilterRuleChecksum	<pre>checksum = dsdd('GetFilterRuleChecksum');</pre> <p>データモデルのフィルタールールのチェックサムを取得します。</p>
GetFilterRuleSets	<pre>filterRuleSets = dsdd('GetFilterRuleSets');</pre> <p>フィルタールールセットのリストを取得します。</p>
GetNumOfFilterRuleSets	<pre>numOfFilterRuleSets = dsdd('GetNumOfFilterRuleSets');</pre> <p>フィルタールールセットの数を取得します。</p>
GetPropertyTable	<pre>propertyTable = dsdd('GetPropertyTable', <objectIdentifier>);</pre> <p>プロパティ名とプロパティ値を含むテーブルを取得します。</p>
GetUnsetPropertyNames	<pre>propertyNames = dsdd('GetUnsetPropertyNames', <objectIdentifier>);</pre> <p>未設定のプロパティの名前を返します。</p>
IsCustomProperty	<pre>bIsCustomProperty = dsdd('IsCustomProperty', <objectIdentifier>, <propertyName>);</pre> <p>プロパティがカスタムプロパティかどうかをチェックします。</p>

IsVisible	<pre>[bVisible] = dsdd('IsVisible', <objectIdentifier>[, <propertyName>];</pre> <p>オブジェクトまたはプロパティが現在のフィルタルールセットに従って表示されるかどうかをチェックします。</p>
ReadFilterRuleSet	<pre>bSuccess = dsdd('ReadFilterRuleSet', attributeName1, attributeValue1, ...);</pre> <p>フィルタルールセットの XML ファイルを読み取ります。</p>
ReloadFilterRuleSets	<pre>bSuccess = dsdd('ReloadFilterRuleSets');</pre> <p>フィルタルールセットの XML ファイルを再度読み取ります。</p>
RemoveVariants	<pre>errorCode = dsdd('RemoveVariants', <objectIdentifier>[, <propertyName>]);</pre> <p>ID != 0 の場合にバリエーションを削除します。</p>
Replace	<pre>[hDDObject, errorCode] = dsdd('Replace'[, attributeName1, attributeValue1, ...]);</pre> <p>オブジェクトを別のオブジェクトのコピーに置き換えます。</p>
ResetFilterRuleSet	<pre>bSuccess = dsdd('ResetFilterRuleSet', <filterRuleSet>);</pre> <p>フィルタルールセットをリセットします。</p>
SetCurrentFilterRuleSet	<pre>bSuccess = dsdd('SetCurrentFilterRuleSet', <filterRuleSet>);</pre> <p>現在のフィルタルールセットを設定します。</p>
SetFilterRule	<pre>bValidPath = dsdd('SetFilterRule', <DataModelPath>, <bVisible>);</pre> <p>データモデル項目のフィルタルールを設定します。</p>
SetFilterRuleByDataModelTag ag	<pre>nRulesSet = dsdd('SetFilterRuleByDataModelTag', <DataModelTag>, <bVisible>);</pre> <p>指定されたデータモデルタグのすべての項目にフィルタルールを設定します。</p>
WriteFilterRuleSet	<pre>bSuccess = dsdd('WriteFilterRuleSet', attributeName1, attributeValue1, ...);</pre> <p>フィルタルールセットを XML ファイルに書き込みます。</p>

AUTOSAR

項目の一覧

本章の内容

サポートされている AUTOSAR リリース	175
新しい AUTOSAR 機能	176

サポートされている AUTOSAR リリース

サポートされている AUTOSAR リリース

次の AUTOSAR リリースがサポートされます。

AUTOSAR リリース	リビジョン
4.1	4.1.3 ¹⁾
	4.1.2 ¹⁾
	4.1.1
4.0	4.0.3
	4.0.2
3.2	3.2.3 ¹⁾
	3.2.2
	3.2.1
3.1	3.1.5
	3.1.4
	3.1.2
3.0	3.1.0
	3.0.7
	3.0.6
2.1	3.0.4
	3.0.2
	2.1.4

¹⁾ TargetLink 4.0 の新しいリビジョン

TargetLink Data Dictionary での AUTOSAR リリースの指定

TargetLink では、AUTOSAR Release 2.x/3.x/4.x に対応した AUTOSAR 準拠コードを生成することができます。

使用する AUTOSAR リリースは、TargetLink Data Dictionary の DD / Pool / Autosar / Config オブジェクトで指定することができます。

AUTOSAR 準拠コード生成の詳細については、「Generating AUTOSAR-Compliant Code」(☞『TargetLink AUTOSAR Modeling Guide』)を参照してください。

システムテンプレートを使用した AUTOSAR 準拠の DD ワークスペースの作成 AUTOSAR Release 3.x および 4.x の両方に対応した新しいシステムテンプレートを使用して、AUTOSAR 準拠の DD ワークスペースを作成することができます。TargetLink Data Dictionary Manager で[File]-[New]-[Create New DD Workspace]の順にクリックして、以下を選択します。

- AUTOSAR 2.x および 3.x の場合は dsdd_master_autosar3.dd [System]
- AUTOSAR 4.x の場合は dsdd_master_autosar4.dd [System]

詳細については、「How to Create DD Workspaces」(☞『TargetLink Data Dictionary Basic Concepts Guide』)を参照してください。

新しい AUTOSAR 機能

AUTOSAR インターフェースでのメトリックス

TargetLink で、インターランナブル変数を除く、AUTOSAR に関連するすべての信号で 2 次元行列がサポートされます。

カテゴリ IDENTICAL の CompuMethods

TargetLink の AUTOSAR インポートで、カテゴリ IDENTICAL の CompuMethods がサポートされます。

ポートの初期化

TargetLink の AUTOSAR インポートで、AUTOSAR ポートのローカルで指定した初期化値がサポートされます。

グローバル定数

TargetLink の AUTOSAR インポートで、他のグローバル定数を参照するグローバル定数がサポートされます。

PerInstanceCalPrm

TargetLink Data Dictionary Manager での PerInstanceCalPrm オブジェクトの作成が簡略化されています。

AUTOSAR メモリマッピング

VariableClass オブジェクトの DeclarationStatements プロパティの値内で \$(Component) 名前マクロを使用して、AUTOSAR メモリマッピングの使用を簡略化することができます。

モデルリンクコード表示

TargetLink でモデルと生成コードとの間のトレーサビリティを向上させるため、AUTOSAR コード生成モードでモデルリンクコード表示がサポートされます。詳細については、「Tracing Objects between Model and Code (Model-Linked Code View)」(📖『TargetLink Preparation and Simulation Guide』)を参照してください。

IncludedDataTypeSets のサポート

TargetLink で、AUTOSAR 4.x 規格で規定された IncludedDataTypeSets がサポートされます。

NetworkRepresentations のインポート/エクスポート

TargetLink で、NetworkRepresentations エLEMENTのインポート/エクスポートが可能になりました。

SwImplPolicyEnum のサポート

TargetLink で、DataElement オブジェクトの以下のプロパティを利用することができます。

プロパティ	値
ImplementationPolicy	<ul style="list-style-type: none"> ■ standard - キューを使用しないセNDERレシーバ通信 ■ queued - キューを使用したセNDERレシーバ通信 ■ measurementPoint - データELEMENTを計測のみに使用 AUTOSAR ファイルのインポート/エクスポートのみに使用



IsQueued プロパティは ImplementationPolicy プロパティに置き換えられています。互換性に関する考慮事項については、「置き換えられた IsQueued プロパティ」(211 ページ)を参照してください。

テストのサポート

項目の一覧**本章の内容**

オンラインでのパラメータ変更の改善	178
ターゲットシミュレーションモジュールの変更	178

オンラインでのパラメータ変更の改善

変化した MIL 値への自動アップデート

シミュレーションを開始する前に、TargetLink で SIL/PIL シミュレーションアプリケーションのパラメータ値を自動的にアップデートすることができます。詳細には、変数の値の差が変化してユーザ定義の許容レベルを超えた場合に、変数値がそれぞれに対応する MIL 値によって自動的に書き込まれます。また、Stateflow の変数のオンラインパラメータアップデートもサポートされます。

関連ドキュメント

- 「Basics on Modifying Parameter Values for Simulation」
([📖](#) 『TargetLink Preparation and Simulation Guide』)
- 「How to Provide Automatic Parameter Updates via a Hook Function」([📖](#) 『TargetLink Preparation and Simulation Guide』)


ターゲットシミュレーションモジュールの変更

新規および廃止されたコンパイラバージョン

次の表は、TargetLink 4.0 でサポートされるコンパイラバージョンを示しています。新規と変更なしの列を参照してください。サポートが終了したコンパイラバージョンは、廃止の列に示しています。

ターゲット	コンパイラ	新規	変更なし	廃止
C16x	TASKING	—	8.6、8.7	—
HCS12	Cosmic	—	4.8	4.7
	Metrowerk	—	5.1	3.1
M32R	Gaio	—	11、9	—
	Renesas	—	5.1	—
MC56F83	Metrowerk	—	8.3	—
MPC55xx	Diab	—	5.9	5.7
	GreenHill	2013	—	2012
	GNU	—	4.1	—
	Metrowerk	—	2.8	—
MPC55xxVLE	Diab	—	5.9	—
	GreenHill	2013	—	2012
	Metrowerk	—	2.8	—
MPC560xVLE	Diab	—	5.9	—
	GreenHill	2013	2012	5.2

ターゲット	コンパイラ	新規	変更なし	廃止
MPC5xx	Diab	—	—	5.7
	GreenHill	—	—	5.1
S12X	Cosmic	—	4.8	—
	Metrowerk	—	5.1	—
SH2	Renesas	—	9.3、9.4	—
SH2A-FPU	Renesas	—	9.4	—
TriCore17xx	TASKING	4.3	3.2	4.2
TriCore1796	GNU	—	3.4	—
V850	GreenHill	2013	—	2012
	NEC	—	3.4	—
XC22xx	TASKING	—	3.0	—

TargetLink でサポートされている評価用ボードの詳細については、
 『TargetLink Evaluation Board Hardware Reference』を参照してください。



MPC5xx ターゲットの TargetLink でのサポートは終了しました。ただし、dSPACE による販売は継続されます。

有効なソフトウェア保守サービス (SMS) 契約に含まれる PIL サポート対象の組み合わせについては、TargetLink 製品サポートセンターにある dSPACE の TargetLink PIL Support Web サイトを参照してください。

Code Generator オプション

新しい Code Generator オプション

TargetLink 4.0 では、次の新しい Code Generator オプションを使用することができます。

概要	説明	デフォルト
「AssignmentOfConditions」		
Relational Operator および Logical Operator ブロックのコードパターンと関連する最適化を制御します。	<p>Relational Operator および Logical Operator ブロックに対し、TargetLink で次の形式のコードが必要かどうかを判断することができます。</p> <pre>if (<Condition>) { Output = 1; } else { Output = 0; }</pre> <p>または</p> <pre>Output = <Condition>;</pre> <p>さらに、前者のコードパターンを後者のコードパターンに変換する制御フローの最適化をアクティブ化することができます(0 と 1 のデータ入力を 1 つずつ持つ Stateflow チャートや Switch ブロックが元になっているコードなどに対して)。</p> <p>それぞれのオプション値は、次のように作用します。</p> <ul style="list-style-type: none"> ■ 0 - None: Relational Operator および Logical Operator ブロックに対して常に制御フローパターンを生成します。制御フローの最適化は行いません。 ■ 1 - RelationAndLogicBlocks: 可能な場合、Relational Operator および Logical Operator ブロックに対して代入パターンを生成します。制御フローの最適化は行いません。 ■ 2 - AllBooleanOutputs: 'RelationAndLogicBlocks' 設定の場合と同様にコードを生成し、出力変数が Boolean タイプの場合に制御フローの最適化を実行します。 ■ 3 - AllOutputs: 'RelationAndLogicBlocks' 設定の場合と同様にコードを生成し、出力変数が数値タイプの場合に制御フローの最適化を実行します。 <p>最後の設定では、MISRA 規則に違反し、論理演算または比較演算のオペランドを持つ算術演算やビット単位の演算を実行するコードが予期せず生成される可能性があります。</p>	2

概要	説明	デフォルト
「InsertComputeThroughOverflowComments」		
Compute through Overflow 演算用に導入されたキャストを示すためのコードコメントを追加します。	より効率的なコードを生成するため、TargetLink では、加算/減算の計算での制御されたオーバーフローが許容されています (Compute Through Overflow)。このオプションを有効にすると、これらの制御されたオーバーフローを導入する生成コード内のキャストの部分にコメントが追加されます。 <ul style="list-style-type: none"> ■ 0 - off ■ 1 - on 	off
「Allow64BitMultiplicationsForArbitraryScaled16BitOperands」		
16 ビットオペランドを持つ積演算のコードパターンを制御します。	再スケーリング演算に対応した最も高い精度のコードを生成するため、16 ビットオペランドを持つ任意スケーリングの積演算は一般に 64 ビットコードを返します。64 ビットコードを回避する必要がある場合は、このオプションを使用すると、1 以下のスケール係数に対してこの動作を抑制することができます。これにより、精度が大幅に損なわれる可能性があります。 <ul style="list-style-type: none"> ■ 0 - off ■ 1 - on 	on
「UtilizeExplicitDependenciesForScalingInvariantSystems」		
Scaling-Invariant 出力ポートが依存する入力ポートに関する、オプションで指定された情報を考慮します。	このオプションが on に設定されている場合、スケーリング伝播関数でオプションで指定した情報 (Scaling-Invariant 出力ポートが依存する入力ポートを指定) をコード生成で考慮します。それ以外の場合は、Scaling-Invariant システムに対して、ワーストケース (各出力ポートが Scaling-Invariant システムのすべての入力ポートに依存) の想定を使用します。Scaling-Invariant システムでのフィードバックループのモデリングには、依存関係の指定が必要です。 <ul style="list-style-type: none"> ■ 0 - off ■ 1 - on 	off

すべての Code Generator オプションの詳細については、「Code Generator Options」(☞『TargetLink Block and Object Reference』)を参照してください。

Code Generator オプションの移行上の注意点

移行に関しては、以下の点に注意してください。

- 廃止された Code Generator オプション
- 変更された Code Generator オプション
- 生成される量産コードの下位互換性を最大限に維持するために新しい Code Generator オプションで推奨される互換性オプション

詳細については、「Code Generator オプションに関する移行上の注意点」(204 ページ)を参照してください。

ツールチェーンの統合

Windows 適合性の向上

ユーザ定義の場所にあるインストールファイル

このバージョンの TargetLink では、TargetLink インストール環境以外の任意の場所にある、以下のカスタマイズファイルを検索することができます。

- DD テンプレートファイル
- DD データモデルのフィルタールール
- DD メニュー拡張
- A2L スタイルシート

関連ドキュメント

- 「How to Define TargetLink's Search Path for DD-Related Customization Files」([📖](#)『TargetLink Customization and Optimization Guide』)

TargetLink 設定のインポート/エクスポート

コマンドラインを使用して、TargetLink の環境設定をインポートおよびエクスポートすることができます。

関連ドキュメント

- 「Basics on Using the TargetLink Preferences Editor」([📖](#)『TargetLink Customization and Optimization Guide』)
- 「How to Import and Export TargetLink Preferences via Command Line」([📖](#)『TargetLink Customization and Optimization Guide』)

TSM の複製

既にインストールされている TSM を複製し、TSM Extension Packages フォルダに格納することができます(このフォルダは TargetLink Preferences Editor で指定することができます)。これらは、TargetLink Preferences Editor の[Target Simulation Module] - [Preselection]ダイアログにも表示されます。また、TargetLink インストールに属さない TSM(たとえば、インストールされた TSM の複製によって作成された TSM や、追加 TSM パッケージとして追加された TSM など)を削除することができます。

関連ドキュメント

- 「How to Clone Target/Compiler Combinations to Outside the TargetLink Installation」 (📖『TargetLink Customization and Optimization Guide』)

その他

一般的な機能拡張および変更

ドキュメント生成の改善

TargetLink のドキュメント生成では、次の新機能を利用することができます。

PDF 生成での日本語文字のサポート TargetLink では、PDF ドキュメントの生成に日本語文字のサポートが追加されています。TargetLink では、デフォルトで Microsoft® Windows® のインストール言語を自動的に検出し、該当する文字セットとフォントを設定します。ただし、DD オブジェクトツリーの Config/General オブジェクトで CharacterSet プロパティを手動で設定することもできます。また、PDF ドキュメント生成を制御するスクリプトでは、生成される PDF ドキュメントで使用するフォントを設定する必要があります(デフォルト:tldoc_pdf)。

目次と表紙の利用 PDF ドキュメント生成で、目次を利用できるようになりました。目次の階層はユーザが設定することができます(詳細は、tldoc_pdf スクリプトを参照)。

また、必要に応じて、生成した PDF に表紙ページを追加することもできます。そのためには、tldoc_pdf M スクリプトを編集し、スクリプト内の記述に沿って CoverPage エントリを追加します。

関数階層の生成の無効化 生成コード内に含まれる関数の階層リストの生成を無効にすることができます。デフォルトでは、有効になっています。これを無効にするには、ドキュメントの生成を制御する M スクリプトを編集し(tldoc_default、tldoc_pdf、tldoc_rtf など)、FunctionsHierarchy エントリを 'off' に設定します。

ユーザが挿入した章の関数リストでの順序 以前のバージョンの TargetLink では、Autodoc Customization ブロックで作成されたユーザが挿入した章は、生成されるドキュメントのナビゲーションペインに統合されませんでした。ユーザが挿入した章は最後に表示されていました。今後は、ユーザが挿入した章がデフォルトで統合されるようになります。これは、HTML で生成されたドキュメントのみに適用されます。

ドキュメント生成プロセスの詳細については、「Basics on Customizing the Generated Documentation」([📖](#)『TargetLink Interoperation and Exchange Guide』)を参照してください。

インデックスの整合性チェックの改善

TargetLink で、コード生成時にインデックスの整合性チェックが実行されるようになりました。

Assignment ブロックおよび Simulink の Selector ブロックに対し、TargetLink でインデックスの整合性チェックが実行されます。

ブロックのインデックスモードがブロックのインデックスポートの信号と一致しない場合、エラーメッセージが生成されます。

GUIによるカスタマイズファイルの作成

TargetLink で、テンプレートからカスタマイズファイルを生成するツール (GUI)を利用することができます。また、このツールを使用して、DD メニュー拡張や A2L スタイルシートを作成することもできます。

関連ドキュメント

- 「How to Create TargetLink Customization Files」([📖](#)『TargetLink Customization and Optimization Guide』)

廃止された制限事項

TargetLink 4.0 では、いくつかの制限事項が廃止されています。詳細については、「廃止された制限事項」(218 ページ)を参照してください。

API コマンド

新しい API 関数

tlExtractSubsystem	<p><code>tl_extract_subsystem</code> API 関数 (TargetLink 3.5 以前) が置き換えられています。</p> <p>TargetLink の「<code>tlExtractSubsystem</code>」(📖 『TargetLink API Reference』) API 関数を利用して、TargetLink サブシステム部分から新しい独立した TargetLink サブシステムを生成することができます。</p> <p>関連ドキュメント</p> <ul style="list-style-type: none">■ 「tlExtractSubsystem」(📖 『TargetLink API Reference』)
tlSimInterface	<p><code>tl_sim_interface</code> API 関数 (TargetLink 3.5 以前) が置き換えられています。</p> <p>TargetLink で、TargetLink シミュレーションエンジン用の M インターフェースとして「<code>tlSimInterface</code>」(📖 『TargetLink API Reference』) API 関数を利用することができます。</p> <p>関連ドキュメント</p> <ul style="list-style-type: none">■ 「Implementing Online Parameter Modification」(📖 『TargetLink Preparation and Simulation Guide』)■ 「tlSimInterface」(📖 『TargetLink API Reference』)
tlSimParameterUpdate	<p>TargetLink の「<code>tlSimParameterUpdate</code>」(📖 『TargetLink API Reference』) API 関数を利用して、オンラインでの自動パラメータ変更を許可することができます。</p> <p>関連ドキュメント</p> <ul style="list-style-type: none">■ 「Implementing Online Parameter Modification」(📖 『TargetLink Preparation and Simulation Guide』)■ 「tlSimParameterUpdate」(📖 『TargetLink API Reference』)
tlMoveDDObject	<p>TargetLink の「<code>tlMoveDDObject</code>」(📖 『TargetLink API Reference』) API 関数を利用して、Data Dictionary オブジェクトの移動や名前変更、このオブジェクトへの参照の調整を行うことができます。</p>

関連ドキュメント

- 「tlMoveDDObject」 ([📖『TargetLink API Reference』](#))
- 「tlFindDDRReferences」 ([📖『TargetLink API Reference』](#))

tlSyncSystemSignature

TargetLink の「**tlSyncSystemSignature**」 ([📖『TargetLink API Reference』](#)) API 関数を利用して、Data Dictionary の Signature または Block オブジェクトから新規の Simulink システムを生成することができます。

関連ドキュメント

- 「tlSyncSystemSignature」 ([📖『TargetLink API Reference』](#))

tlOperationMode

`tl_switch_blockset` API 関数 (TargetLink 3.5 以前) が置き換えられています。

TargetLink の「**tlOperationMode**」 ([📖『TargetLink API Reference』](#)) API 関数を利用して、TargetLink の動作モード (フル機能またはスタンドアロン) を取得または設定することができます。

関連ドキュメント

- 「tlOperationMode」 ([📖『TargetLink API Reference』](#))

tlUpgrade

`tl_upgrade` API 関数 (TargetLink 3.5 以前) が置き換えられています。TargetLink の「**tlUpgrade**」 API 関数を利用して、TargetLink ブロックを含むブロック線図を最新バージョンにアップグレードすることができます。

関連ドキュメント

- 「tlUpgrade」 ([📖『TargetLink API Reference』](#))

TlTsmManager

TargetLink の `TlTsmManager.exe` コマンドを利用して、既存の TargetLink シミュレーションモジュールのクローンを作成することができます。

関連ドキュメント

- 「How to Clone Target/Compiler Combinations to Outside the TargetLink Installation」 ([📖『TargetLink Customization and Optimization Guide』](#))

TargetLink 4.0 および TargetLink Data Dictionary 4.0 への移行

TargetLink 2.x からのアップグレード



TargetLink 3.1 より前の TargetLink バージョンのライブラリ、モデル、DD ファイルを直接アップグレードすることはできません。ただし、最初に古いライブラリ、モデル、DD ファイルを TargetLink 3.5 に移行し、その後 TargetLink 4.0 にアップグレードすることができます。

項目の一覧

本章の内容

Data Dictionary とデータ管理	187
コードの変更	192
Code Generator オプション	204
アクセス関数の変更	205
AUTOSAR に関する移行上の注意点	207
その他	211
API コマンド	216
廃止事項	217
メッセージ	219
Stateflow 関連の変更	220

Data Dictionary とデータ管理

項目の一覧

本章の内容

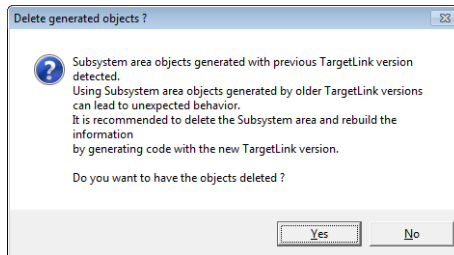
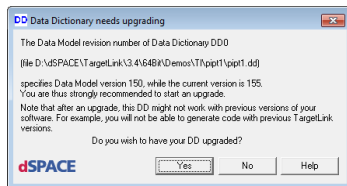
TargetLink 4.0 への移行	188
インクルード DD ファイルのある Data Dictionary をアップグレードする方法	189

TargetLink 4.0 への移行



TargetLink 4.0 では、TargetLink 3.1 以上で作成したモデル、ライブラリ、Data Dictionary が自動的にアップグレードされます。

アップグレード時には、次のダイアログが表示されます。



ユーザによるインタラクティブな操作 次の場合は、ユーザによるインタラクティブな操作が必要になります。

- 「TargetLink 向けに準備されていない古いライブラリ」(188 ページ)
- 「TargetLink 32 ビットバージョンから TargetLink 64 ビットバージョンへのアップグレード」(189 ページ)
- 「インクルードされる部分的な DD ファイルを含む DD ファイル」(189 ページ)
- 「アクセス関数の変更」(189 ページ)

TargetLink 向けに準備されていない古いライブラリ

「`tl_prepare_system`」(☞ 『TargetLink API Reference』) API 関数を使用して準備されていない TargetLink 3.x で作成したライブラリは、TargetLink 4.0 で自動的にアップグレードすることはできません。

解決策

1. 以前の TargetLink バージョンでライブラリを開き、
「`tl_prepare_system`」(☞ 『TargetLink API Reference』)を使用してアップグレード用に準備します。

2. ライブラリを保存します。
3. TargetLink 4.0 でライブラリを開きます。

関連ドキュメント

- 「How to Make TargetLink User Libraries Upgrade-Capable」
([📖『TargetLink Orientation and Overview Guide』](#))

TargetLink 32 ビットバージョンから TargetLink 64 ビットバージョンへのアップグレード

32 ビットバージョンの TargetLink でビルドしたカスタムコード S-function は、64 ビットバージョンの TargetLink では使用することができません。これは、逆の場合も同様です。

解決策 `tlUpgrade('Model', <MyModel>, 'CheckModel', 'FixIssues')` (`tlUpgrade`) ([📖『TargetLink API Reference』](#))を参照) API 関数を使用して、すべてのカスタムコード S-function の再ビルドを行います。

インクルードされる部分的な DD ファイルを含む DD ファイル

インクルードされる部分的な DD ファイルを含む DD ファイルをアップグレードする場合は、「インクルード DD ファイルのある Data Dictionary をアップグレードする方法」(189 ページ)を参照してください。

アクセス関数の変更

`ADDRESS_BY_PARAMETER` アクセス関数が変更されています。

ユーザによるインタラクティブな操作が必要になる場合があります。詳細については、「`ADDRESS_BY_PARAMETER` アクセス関数の変更」(206 ページ)を参照してください。

インクルード DD ファイルのある Data Dictionary をアップグレードする方法

目的

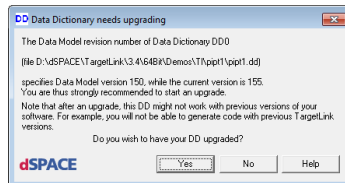
アップグレードしていない古い Data Dictionary ファイルと TargetLink モデルを開く場合は、Data Dictionary ファイルをアップグレードする必要があります。

操作手順

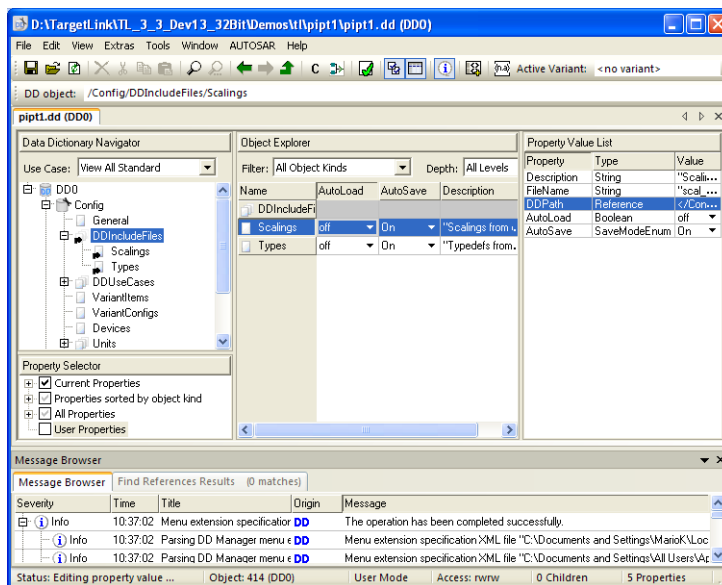
インクルード DD ファイルのある Data Dictionary をアップグレードするには

- 1 モデルおよび参照する TargetLink Data Dictionary を開くか、MATLAB コマンドウィンドウで `dsdd('Open', <DDFile>)` と入力します。

古いバージョンの DD が使用されている場合は、[Data Dictionary needs upgrading]ダイアログが自動的に開きます。



- アップグレードダイアログで[No]を選択します。
- /Config/DDIncludeFiles で、下の画面のように各インクルード DD ファイルの AutoLoad および AutoSave プロパティを設定します。

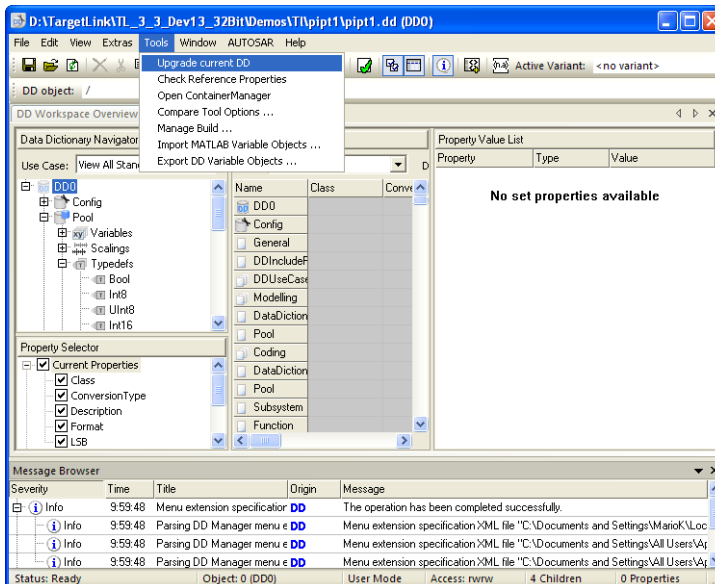


これにより、Data Dictionary とインクルード DD ファイルのアップグレード後、Data Dictionary を保存するときにアップグレードしたインクルード DD ファイルが保存されます。Object Explorer を使用して、複数のインクルード DD ファイルのこれらのプロパティを設定することができます。



[Point of Inclusion]ダイアログを使用してインクルード DD ファイルのプロパティを設定することもできます。

- 4 DD Manager で[Tools] - [Upgrade Current DD]を使用して DD アップグレード(インクルード DD ファイルを含む)を開始するか、MATLAB コマンドウィンドウで `dsdd('Upgrade')` と入力します。



- 5 (関連 DD ファイルへの書き込みを許可して)Data Dictionary を保存します。これで、DD ファイルとインクルードされる部分的な DD ファイルのアップグレードは完了です。

結果

DD ファイルを再度開くと、DD ファイルとインクルードされている部分的な DD ファイルが最新であるため、アップグレードダイアログは開きません。ファイルを正常にアップグレードした後で、インクルード DD ファイルを古い設定に戻すことが必要となる場合もあります。

コードの変更

コードの変更

ADDRESS_BY_PARAMETER アクセス関数

ベクトル変数で `_var` マクロ引数のタイプが scalar type から pointer to scalar type に変更されています。詳細については、「ADDRESS_BY_PARAMETER アクセス関数の変更」(206 ページ)を参照してください。

ブロックコメントでの信号幅の 表示

ブロックのコメントに、信号インデックスに関する情報が表示されなくなりました。

TargetLink 3.5 以前	TargetLink 4.0
<pre>for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++) { /* Gain: Subsystem/Gain [0..4] */ Sal_Gain[Aux_S32] = Sal_InPort[Aux_S32] * 3; /* Gain: Subsystem/Gain [5..9] */ Sal_Gain[Aux_S32 + 5] = Sal_InPort[Aux_S32 + 5] * 5; }</pre>	<pre>for (Aux_S32 = 0; Aux_S32 < 5; Aux_S32++) { /* Gain: Subsystem/Gain */ Sal_Gain[Aux_S32] = Sal_InPort[Aux_S32] * 3; Sal_Gain[Aux_S32 + 5] = Sal_InPort[Aux_S32 + 5] * 5; }</pre>

if-else 分岐のブロックコメント

各ステートメントにブロックコメントが常に配置されるようになりました。

TargetLink 3.5 以前	TargetLink 4.0
<pre>/* Switch: Subsystem/Switch1 */ if (c == 0) { ... <code> ... } else { ... <code> ... }</pre>	<pre>/* Switch: Subsystem/Switch1 */ if (c == 0) { /* Switch: Subsystem/Switch1 */ ... <code> ... } else { /* Switch: Subsystem/Switch1 */ ... <code> ... }</pre>

ベクトル変数と行列変数

Interpret as 1D フラグが設定されていない場合、ブロック変数 (Constant ブロックなどにある) は (Simulink に準じて) 次のように扱われます。

TargetLink 3.5 以前	TargetLink 4.0
ベクトル変数を作成します。	行列変数を作成します。

Simulink Inport/Outport の多重信号

多重信号がアトミックサブシステムの非拡張入力/出力ポートブロックに供給される場合、TargetLink では、ブロック全体が 1 つのベクトル化された関数パラメータまたは 1 つのベクトル化されたグローバル変数として実装されます。

TargetLink 3.5 以前	TargetLink 4.0
例: 1 つの多重信号 (場合によっては、1 つの構造体) ごとに 1 つのファンクションパラメータ: <pre>Void Sa2_Inner(Int16 Sa2_InPort[2], Int16 Sa2_InPort_a);</pre> または <pre>struct BS_IS_Sal_BusInport { Int16 Sal_Signal1[2]; Int16 Sal_Signal2; } [...] Void Sa2_Inner(struct BS_IS_Sal_BusInport * Sa2_InPort);</pre>	例: ポート全体に対して 1 つのファンクションパラメータ: <pre>Void Sa2_Inner(Int16 Sa2_InPort[3]);</pre>

プロパティの継承と構造体コンポーネントの変数へのマッピング

バスエレメントが 1 つの DD 構造体にマッピングされているバスポートブロックから [Inherit properties] を介して、ブロックがデータタイプを継承し、TargetLink で構造体タイプを伝播できない場合、TargetLink は次のように DD 構造体コンポーネントごとに個別の変数を生成します。

TargetLink 3.5 以前	TargetLink 4.0
<pre>static Int16 X_Sal_Unit_Delay[2] = { 0, 0 }; [...] Sal_a = X_Sal_Unit_Delay[0]; X_Sal_Unit_Delay[0] = StructVar.a; Sal_b = X_Sal_Unit_Delay[1]; X_Sal_Unit_Delay[1] = StructVar.b;</pre>	<pre>static Int16 X_Sal_Unit_Delay = 0; static Int16 X_Sal_Unit_Delay_a = 0; [...] Sal_a = X_Sal_Unit_Delay; X_Sal_Unit_Delay = StructVar.a; Sal_b = X_Sal_Unit_Delay_a; X_Sal_Unit_Delay_a = StructVar.b;</pre>

RTE API 関数の Pointer-to-const 戻り値タイプ

非スカラーデータタイプを使用する Rte_IRead、Rte_Calprm、および Rte_CData 関数呼び出しで、/Pool/Autosar/Config/UseRtePointerToConstForNonScalarReturnValue DD プロパティが on に設定されている場合、TargetLink は Pointer-to-const 戻り値タイプを使用します。

TargetLink 3.5 以前	TargetLink 4.0
<pre>StructWSubStructsType * DE_StructWSubStructs; DE_StructWSubStructs = Rte_IRead_Swc_Run_ReceiverPort_DE_StructWSubStructs();</pre>	<pre>const StructWSubStructsType * DE_StructWSubStructs; DE_StructWSubStructs = Rte_IRead_Swc_Run_ReceiverPort_DE_StructWSubStructs();</pre>

RTE API 関数の Pointer-to-const 関数シグネチャ

読み取り専用で非スカラーデータタイプの IN ファンクションパラメータは、/Pool/Autosar/Config/UseRtePointerToConstForInArguments DD プロパティが on に設定されている場合、追加の修飾子 (const) を提供します。

TargetLink 3.5 以前	TargetLink 4.0
<code>Rte_<Fcn>(..., <Vector Address>, ...);</code>	<code>Rte_<Fcn>(..., (const T*) <Vector Address>, ...);</code>

AUTOSAR 戻り変数へのポインタ

可読性を向上させ、グローバル名の競合を回避するため、AUTOSAR ポインタ戻り変数へのポインタで新しい名前テンプレートが使用されます。

TargetLink 3.5 以前	TargetLink 4.0
<code>\$(DataElement)\$R</code>	ローカルポインタ: <code>p_\$(DataElement)\$R</code> グローバルポインタ: <code>p_\$(DataElement)\$R</code>

Selector ブロック

TargetLink で入力の暗黙変数を生成する場合に、わかりやすいように名前の末尾にサフィックス `_In` が付くようになりました。

TargetLink 3.5 以前	TargetLink 4.0
<code>Int16 Sa1_Selector_a[20];</code>	<code>Int16 Sa1_Selector_In[20];</code>

Rate Limiter ブロック

暗黙変数は、入力のデータタイプまたはスケールが異なる場合や、出力に対して飽和が有効になっている場合といった必要な場合にのみ、ブロック入力に対して生成されます。

DisableArbitraryOptimizations コードジェネレータオプション

`DisableArbitraryOptimizations` の名前が `Allow64BitMultiplicationsForArbitraryScaled16BitOperands` で変更され、デフォルト設定が off から on に変更されています。その結果、TargetLink で生成される 64 ビット演算が増加します。ただし、シミュレーション結果の精度は、再スケールリングの精度向上に伴って向上します。

Direct LUT ブロック

`Slices < LoopUnrollThreshold` の Direct LUT ブロックパターンに対し、この TargetLink バージョンでは、次のコード生成方式が適用されます。

- インデックスに対する暗黙変数の削減
- コード行の順序変更による効率性の向上
- コードコメントの追加による可読性の向上

TargetLink 3.5 以前	TargetLink 4.0
<pre>Aux_U8 = C_U8SATTI16_SATu(Sal_Index1D_inport[0], 10); Aux_U8_a = C_U8SATTI16_SATu(Sal_Index1D_inport[1], 10); Aux_U8_b = C_U8SATTI16_SATu(Sal_Index1D_inport[2], 10); /* TargetLink output: RootSystem/OutPort2 # combined # RootSystem/Direct Look-Up Table (n-D) Ext */ Sal_OutPort2[0] = Sal_TableData_inport[Aux_U8]; Sal_OutPort2[1] = Sal_TableData_inport[Aux_U8_a]; Sal_OutPort2[2] = Sal_TableData_inport[Aux_U8_b];</pre>	<pre>Aux_U8 = C_U8SATTI16_SATu(Sal_Index1D_inport[0], 10); /* RootSystem/Direct Look-Up Table (n-D) Ext # combined # TargetLink output: RootSystem/OutPort2 */ Sal_OutPort2[0] = Sal_TableData_inport[Aux_U8]; Aux_U8 = C_U8SATTI16_SATu(Sal_Index1D_inport[1], 10); /* RootSystem/Direct Look-Up Table (n-D) Ext # combined # TargetLink output: RootSystem/OutPort2 */ Sal_OutPort2[1] = Sal_TableData_inport[Aux_U8]; Aux_U8 = C_U8SATTI16_SATu(Sal_Index1D_inport[2], 10); /* RootSystem/Direct Look-Up Table (n-D) Ext # combined # TargetLink output: RootSystem/OutPort2 */ Sal_OutPort2[2] = Sal_TableData_inport[Aux_U8];</pre>

暗黙補助変数の透過性の向上

関数の場合、暗黙変数は最初に常に関数レベルで作成されます。コードジェネレータオプション

ReduceScopeOfVariablesOnlyDownToFunctionLevel をアクティブ化すると、すべての変数が影響を受けます。オプション

ReduceScopeOfVariablesOnlyDownToFunctionLevel および

Optimization にデフォルト値がある場合、変更は暗黙ベクトル変数で観測することができます。このため、Aux_変数の数や名前が変わる可能性があります。

TargetLink 3.5 以前	TargetLink 4.0
<pre>MyFunc() { ... if(...) { Int32 Aux_S32; Aux_S32 = IN * 42; MinOut = C_16FITI32_SAT(Aux_S32, ...); } }</pre>	<pre>MyFunc() { Int32 Aux_S32; ... if(...) { Aux_S32 = IN * 42; MinOut = C_16FITI32_SAT(Aux_S32, ...); } }</pre>

インクリメンタルシステムでの関数戻り値の実パラメータ

以下の場合を考えます。

- インクリメンタルシステムの出力ポートが関数の戻り値として指定されている。
- 実パラメータの名前(つまり、関数呼び出しステートメントに示される変数)が明示的に指定されていないか、<FormalparameterName>\$Rとして指定されている。
- インクリメンタルシステム用に生成されるコードで、最適化によって元の変数が削除されたため、異なる変数が戻り値として使用されている。

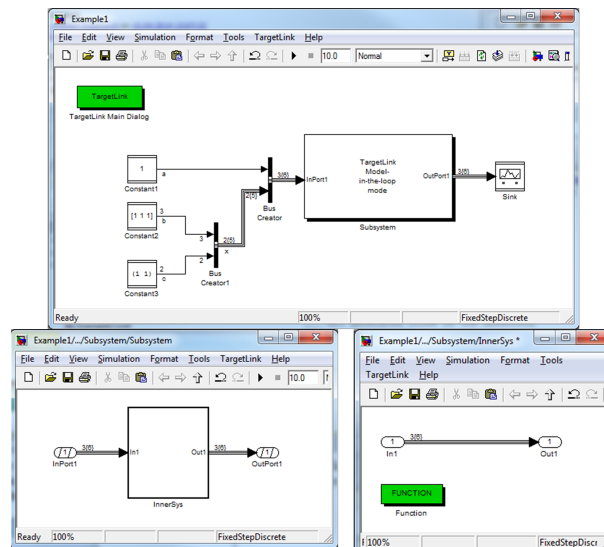
ラッピングシステムの実パラメータの名前は、次のようになります(システムがインクリメンタルかどうかは無関係)。

TargetLink 3.5 以前	TargetLink 4.0
<FormalparameterName>	<FormalparameterName>_a

Simulink Inport/Outport に対する暗黙インターフェース変数

アトミックサブシステムの拡張解除された入力ポートと出力ポートで、暗黙インターフェース変数(IF 変数)に対して生成されるコードが、次のように異なります。

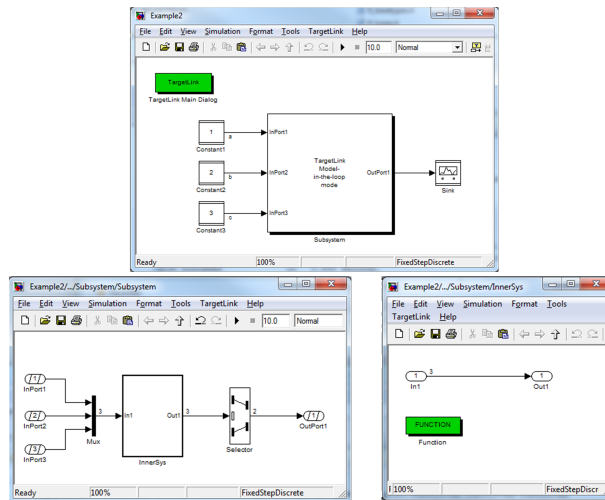
バス信号



ネストされたバス信号 (a;(b[3];c[2])) が拡張解除された出力ポートに供給される場合、出力ポートの IF 変数は、次のように生成されます。

TargetLink 3.5 以前	TargetLink 4.0
バス信号全体に対して 1 つの IF 変数: static Int16 IF_Sa2_Out1[6];	各バスエレメントに対して 1 つの IF 変数: static Int16 IF_Sa2_Out1; static Int16 IF_Sa2_Out1_a[3]; static Int16 IF_Sa2_Out1_b[2];

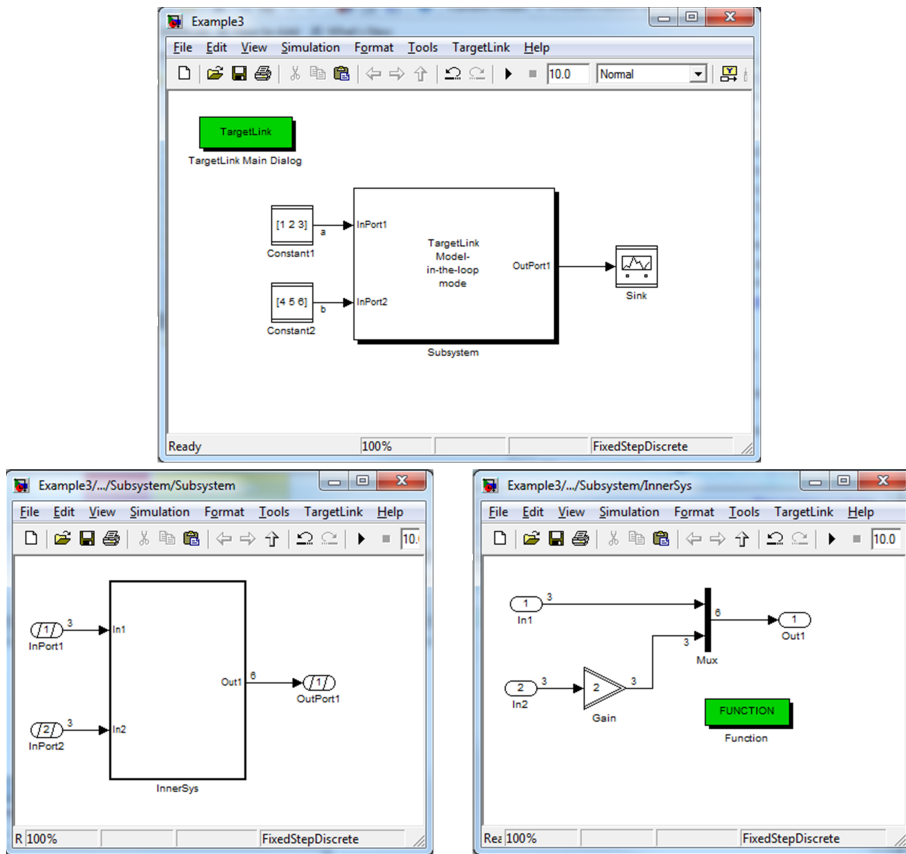
多重信号と Selector ブロックの組み合わせ



多重スカラー信号 (a;b;c) が拡張解除された出力ポートに供給される場合、出力ポートの IF 変数は、次のように生成されます。

TargetLink 3.5 以前	TargetLink 4.0
2 つの IF スカラー変数: <pre>static Int16 IF_Sa2_Out1; static Int16 IF_Sa2_Out1_b;</pre>	1 つの IF ベクトル変数: <pre>static Int16 IF_Sa2_Out1[3];</pre>

アトミックサブシステム内の異なるブロックから発生する多重信号



多重ベクトル信号 [a[3];b[3]] が拡張解除された出力ポートに供給される場合、出力ポートの IF 変数は、次のように生成されます。

TargetLink 3.5 以前	TargetLink 4.0
1 つの IF ベクトル変数: <pre>static Int16 IF_Sa2_Gain[3]; static Int16 IF_Sa2_Out1[3];</pre>	3 つの IF スカラー変数: <pre>static Int16 IF_Sa2_Gain[3]; static Int16 IF_Sa2_Out1; static Int16 IF_Sa2_Out1_a; static Int16 IF_Sa2_Out1_b;</pre>

ループ内の代入でのマイナスのオフセット

for ループ内での代入に対し、TargetLink はマイナスのオフセットを生成しません。

TargetLink 3.5 以前	TargetLink 4.0
<pre>for(int Aux__a= 4; Aux__a < 10; Aux__a++) { Aux[Aux__a - 3] = In[Aux__a] }</pre>	<pre>for(int Aux__a= 1; Aux__a < 7; Aux__a++) { Aux[Aux__a] = In[Aux__a + 3] }</pre>

修飾子の欠損の軽減

修飾子が `const` の場合、TargetLink では、暗黙変数を導入して修飾子の欠損を軽減します。

TargetLink 3.5 以前	TargetLink 4.0
<pre>const volatile Int16 MyConstVolArray[3] = {0,0,0}; void Fcn(volatile Int16 * pFormalParam); Fcn((volatile Int16*) MyConstVolArray); /* Loss of const */</pre>	<pre>const volatile Int16 MyConstVolArray[3] = {0,0,0}; void Fcn(volatile Int16 * pFormalParam); Int16 MyConstVolArray_Aux[3]; for i = 0:2 MyConstVolArray_Aux[i] = MyConstVolArray[i]; Fcn((volatile Int16 *)MyConstVolArray_Aux);</pre>

このような暗黙変数が導入されると、コードジェネレータによってメッセージ「A17363」が出力され、仕様やソリューションに不整合が生じている可能性が通知されます。

これらの暗黙変数は、次のコンテキストで現れます。

- FIR Filter ブロックの関数
- TargetLink Function ブロックを含むサブシステム
- テンプレートを介して再利用構造体が `const` に設定された再利用関数
- サブシステムの直前に配置されたモデルエンティティから発生した RTE 呼び出し

次のコンテキストは分析されません。

- カスタムルックアップスクリプト関数
- カスタムコード入力
- Stateflow の外部 C 関数

ビットフィールドのキャスト

TargetLink では、`BitfieldVar = BoolVar`;または `BitfieldVar = <BoolOperation>`;など、ビットフィールドが関連する割り当て内の不要な `unsigned int` キャストが削除されるようになりました。

<BoolOperation>は次のいずれかを表します:>, >=, ==, !=, <=, <, &&, ||, !

TargetLink 3.5 以前	TargetLink 4.0
GIBFS_b_.X_Sb21_Memory_U = (unsigned int) T_SwHDIPushed_b;	GIBFS_b_.X_Sb21_Memory_U = T_SwHDIPushed_b;

EfficientVectorHandling の削除

コードジェネレータオプションの EfficientVectorHandling が TargetLink 4.0 から削除されました。

TargetLink で、常にベクトルおよびマトリクスの次元を「LoopUnrollThreshold」に照らし合わせてチェックして、定義したしきい値より大きい次元に対して for ループを生成します。詳細については、「Code Generator オプションに関する移行上の注意点」(204 ページ)を参照してください。

さらに、Stateflow 信号と Simulink 信号がループ生成と同様に扱われるようになりました。これは、Stateflow に生成されるコードで特に見取れます。

EfficientVectorHandling を off に設定していた場合や、ベクトル信号の次元が「LoopUnrollThreshold」より小さかった場合は、コードの変更を除外することはできません。

TargetLink 3.5 以前	TargetLink 4.0
<pre>do { Cal_a[idx1] = Cal_b[idx1]; if (Cal_b[idx1] < 0) { Cal_a[idx1] = -Cal_a[idx1]; } idx1++; } while (idx1 < 4);</pre>	<pre>Cal_a[0] = Cal_b[0]; Cal_a[1] = Cal_b[1]; Cal_a[2] = Cal_b[2]; Cal_a[3] = Cal_b[3]; if (Cal_b[0] < 0) { Cal_a[0] = -Cal_a[0]; } if (Cal_b[1] < 0) { Cal_a[1] = -Cal_a[1]; } if (Cal_b[2] < 0) { Cal_a[2] = -Cal_a[2]; } if (Cal_b[3] < 0) { Cal_a[3] = -Cal_a[3]; }</pre>

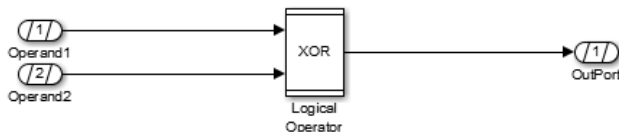
TargetLink 4.0 では、for ループは「Basics on Processing Vectors and Matrices」(PDF『TargetLink Preparation and Simulation Guide』)に記載される原理に従って生成されます。

次元が LoopUnrollThreshold より小さい	次元が LoopUnrollThreshold 以上
<pre> Cal_a[0] = Cal_b[0]; Cal_a[1] = Cal_b[1]; Cal_a[2] = Cal_b[2]; Cal_a[3] = Cal_b[3]; if (Cal_b[0] < 0) { Cal_a[0] = -Cal_a[0]; } if (Cal_b[1] < 0) { Cal_a[1] = -Cal_a[1]; } if (Cal_b[2] < 0) { Cal_a[2] = -Cal_a[2]; } if (Cal_b[3] < 0) { Cal_a[3] = -Cal_a[3]; } </pre>	<pre> for (Aux_S32 = 0; Aux_S32 < 4; Aux_S32++) { Cal_a[Aux_S32] = Cal_b[Aux_S32]; if (Cal_b[Aux_S32] < 0) { Cal_a[Aux_S32] = -Cal_a[Aux_S32]; } } </pre>

Logical Operator ブロック(排他的 OR)

「Operator」が XOR に設定された Logical Operator ブロックで生成されるコードが変更されています。

TargetLink では、論理式と算術式を一貫して区別するため、算術式の ^ ではなく論理式の != が生成されます。これにより、MISRA-C 適合度が向上します。



TargetLink 3.5 以前	TargetLink 4.0
<pre> /* Logical: Logical Operator TL_Root/Logical Operator */ block_out = (!(Operand1)) ^ (!(Operand2)); </pre>	<pre> /* Logical: TL_Root/Logical Operator */ block_out = (Operand1 != 0) != (Operand2 != 0); </pre>

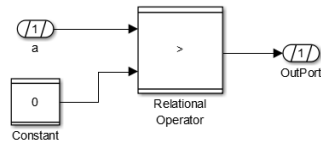
状態更新の再スケジューリング

TargetLink の状態更新の再スケジューリングに関する分析が改善されています。これにより、コード内の状態更新の順序が変わる可能性があります。

関係と論理演算

Logical Operator および Relational Operator ブロックに対する TargetLink のコードパターンと最適化動作が変更されています。

例



最適化 = OFF

TargetLink 3.5 以前	TargetLink 4.0
<pre>if (a > 0) { Sal_Relational_Operator = 1; } else { Sal_Relational_Operator = 0; }</pre>	<pre>(AssignmentOfConditions > 0) Sal_Relational_Operator = (a > 0);</pre>

最適化 = ON

TargetLink 3.5 以前	TargetLink 4.0
<pre>(NoAssignmentOfBooleanIfThenElse=OFF)(デフォルト) Sal_Relational_Operator = (a > 0); — または —(a のデータタイプに依存) Sal_Logical_Operator = (a != 0);</pre>	<pre>Sal_Relational_Operator = (a > 0);</pre>

この変更は、他のソースからの論理演算や比較演算にも影響します。デフォルトで、TargetLink では、代入される側が Boolean 変数またはビットフィールド変数である場合に、最適化されていない制御フローパージョンのみを代入に変換するようになりました。

```
Sal_Logical_Operator1 = b && c;
if (Sal_Logical_Operator1 != 0) {
    Sal_BoolSwitch = 0;
} else {
    Sal_BoolSwitch = 1;
}

Sal_Logical_Operator2 = d <= 0;
if (Sal_Logical_Operator2 != 0) {
    Sal_Int8Switch = 0;
} else {
    Sal_Int8Switch = 1;
}

Sal_Logical_Operator3 = e != 0;
if (Sal_Logical_Operator3 != 0) {
    Sal_FloatSwitch = 0.F;
} else {
    Sal_FloatSwitch = 1.F;
}
```

TargetLink 3.5 以前	TargetLink 4.0
(NoAssignmentOfBooleanIfThenElse=OFF(デフォルト)) Sal_BoolSwitch = (!b !c); Sal_Int8Switch = (d > 0); Sal_FloatSwitch = (e == 0);	(AssignmentOfConditions=2(BooleanOutputsOnly;デフォルト)) Sal_BoolSwitch = (!b !c); if (d <= 0) { Sal_Int8Switch = 0; } else { Sal_Int8Switch = 1; } if (e != 0) { Sal_FloatSwitch = 0.F; } else { Sal_FloatSwitch = 1.F; } (AssignmentOfConditions=3(すべての出力)) Sal_BoolSwitch = (!b !c); Sal_Int8Switch = (Int8) (d > 0); Sal_FloatSwitch = (Float32) (e == 0);

初期コードパターンでのこの変更により、生成コードでの最適化結果に違いが生じる可能性があります(たとえば、論理演算または比較演算を伴う代入が TargetLink 3.5 以前の場合よりも後に実行されます)。

RDI マクロ定義

TargetLink で、キャストを受け取る初期値の前後に括弧が付くようになりました。これにより、MISRA-C 適合度が向上します。

TargetLink 3.5 以前	TargetLink 4.0
#define PIM_RDI (sint32 (*) [2]) Rte_Pim_PIM_Matrix3x2()	#define PIM_RDI ((sint32 (*) [2]) Rte_Pim_PIM_Matrix3x2())

コード効率化に関する変更点

TargetLink 4.0 では、コードの効率性が向上しました。これにより、以前の TargetLink バージョンと比較して、生成コードに違いが生じる可能性があります。詳細については、「コード効率性の向上」(168 ページ)を参照してください。

Code Generator オプション

Code Generator オプションに関する移行上の注意点

Code Generator オプションの名前の変更 次の Code Generator オプションの名前が変更されています。

以前の名前	新しい名前
NoAssignmentOfBooleanExpressions	「AssignmentOfConditions」
PolySpaceSupport	「InsertComputeThroughOverflowComments」
DisableArbitraryOptimizations	「Allow64BitMultiplicationsForArbitraryScaled16BitOperands」

新しい Code Generator オプションに推奨される互換性設定 TargetLink 3.5 との最適な下位互換性を維持するため、次の表に示す新しい Code Generator オプションに**互換設定値**を使用することをお勧めします。

Code Generator オプション	互換設定値	デフォルト値
「AssignmentOfConditions」	3 - AllOutputs ¹⁾ 0 - None ²⁾	2 - AllBooleanOutputs
「Allow64BitMultiplicationsForArbitraryScaled16BitOperands」 ³⁾	on ⁴⁾ off ⁵⁾	on

¹⁾ 「Optimization option」が on に設定されている場合。

²⁾ 「Optimization option」が off に設定されている場合。

³⁾ DisableArbitraryOptimizations の代わりとなる

⁴⁾ TargetLink 3.5 で DisableArbitraryOptimizations が off に設定されていた場合。

⁵⁾ TargetLink 3.5 で DisableArbitraryOptimizations が on に設定されていた場合。

TargetLink 3.5 より前の TargetLink バージョンについては、それぞれのバージョンの『**新機能と移行手順**』ドキュメントを参照してください。詳細については、「以前のリリースのドキュメント」(33 ページ)を参照してください。

削除された Code Generator オプション 次の Code Generator オプションは TargetLink から削除されました。

削除されたオプション	代わりのオプション	互換設定値
EfficientVectorHandling	「LoopUnrollThreshold」(☑「TargetLink Block and Object Reference」) ¹⁾	INF (EfficientVectorHandling が off に設定されていた場合)

¹⁾ 詳細については、「EfficientVectorHandling の削除」(200 ページ)および「Basics on Processing Vectors and Matrices」(☑「TargetLink Preparation and Simulation Guide」)を参照してください。

新しい Code Generator オプション

新しい Code Generator オプションの詳細については、「新しい Code Generator オプション」(180 ページ)を参照してください。

関連トピック

リファレンス

- 「Code Generator Options」(『TargetLink Block and Object Reference』)

アクセス関数の変更

項目の一覧

本章の内容

マクロアクセス関数用のデフォルトのマクロ本体	205
ADDRESS_BY_PARAMETER アクセス関数の変更	206
新しいアクセス関数固有の名前マクロ	207

マクロアクセス関数用のデフォルトのマクロ本体

TargetLink では、マクロアクセス関数用のデフォルトのマクロ本体を利用することができます。これらのデフォルトのマクロ本体は、次の場合に使用します。

- DD AccessFunction オブジェクトで、Macro プロパティが on に設定された DD FunctionClass オブジェクトを参照する場合。
- DD AccessFunction オブジェクトの MacroBody プロパティが空の場合。

MISRA-C に準拠するため、マクロ引数用の次のプレースホルダは括弧で囲まれます。

- `_var`
- `_value`

Data Dictionary Manager の `dsdd_master_advanced` および `dsdd_master_autosar3.dd` [System]/`dsdd_master_autosar4.dd` [System] システムテンプレートに含まれるあらかじめ設定されたアクセス関数テンプレートのマクロ本体のサンプルは、これに合わせて変更されています。

関連ドキュメント

- 「Basics on Access Functions」 (📖 『TargetLink Customization and Optimization Guide』)
- 「Specifying Access Functions as Preprocessor Macros」 (📖 『TargetLink Customization and Optimization Guide』)
- 「Overview of Predefined Access Function Templates」 (📖 『TargetLink Customization and Optimization Guide』)

ADDRESS_BY_PARAMETER アクセス関数の変更

ベクトル変数で、_var マクロ引数のタイプが scalar type から pointer to scalar type に変更されました。これは、次の場合に該当します。

- DD AccessFunction オブジェクトの Kind プロパティが ADDRESS_BY_PARAMETER に設定されている場合。
- DD AccessFunction オブジェクトで、Macro プロパティが on に設定された DD FunctionClass オブジェクトを参照する場合。

TargetLink 3.5	TargetLink 4.0
<code>(GetAddressU16(vector[0])) [1] = 5;</code>	<code>(GetAddressU16(vector)) [1] = 5;</code>



TargetLink 4.0 では、同じマクロ本体を使用してスカラーとベクトルにアクセスできず、変数の種類ごとに個別のマクロ本体を使用する必要があります。

既存の Data Dictionary をアップグレードする場合、TargetLink では、すべての空の Kind プロパティを APPLY_TO_SCALAR に設定します。これは、マクロ AF と関数 AF の両方に作用します。



TargetLink のあらかじめ設定されたアクセス関数テンプレートを使用する場合は、`dsdd_master_advanced/dsdd_master_autosar3.dd` [System]/`dsdd_master_autosar4.dd` [System] システムテンプレートに基づいた DD ワークスペースから新規バージョンをコピーすることができます。

新しいアクセス関数固有の名前マクロ

TargetLink に次の名前マクロが追加されています。

名前マクロ	説明
\$ (Dim1Width)、 \$ (Dim2Width)	これらの名前マクロをアクセス関数テンプレートで使用して、関数名とマクロ本体を指定することができます。 <ul style="list-style-type: none"> ■ \$ (Dim1Width) - 2-D 変数の第 1 次元 (行) のエレメント数に置き換えられます。 ■ \$ (Dim2Width) - 2-D 変数の第 2 次元 (列) のエレメント数に置き換えられます。 スカラー変数には影響しません。ベクトル変数には \$ (Dim1Width) のみ適用することができます。
\$ (VarAccess)	このマクロでは、アクセス関数テンプレートでマクロ本体を指定でき、必要に応じて関連する変数に正しくアクセスすることができます。非構造体コンポーネント変数の場合、\$v と同様に置き換えられ、構造体コンポーネントの場合、\$ (Sv) \$ (SPath) . \$v と同様に置き換えられます。

関連ドキュメント

- 「Basics on Using Name Macros」 ([📖 『TargetLink Customization and Optimization Guide』](#))
- 「Changing Access Function Implementations Using Name Macros」 ([📖 『TargetLink Customization and Optimization Guide』](#))
- 「Overview of Predefined Access Function Templates」 ([📖 『TargetLink Customization and Optimization Guide』](#))

AUTOSAR に関する移行上の注意点

項目の一覧

本章の内容

2 次元行列データエレメントとオペレーション引数	208
AUTOSAR エクスポート	208
Container Manager の個別インストール	209
アプリケーションデータタイプ	210
配列および行列タイプの制約	210
置き換えられた IsQueued プロパティ	211

2次元行列データ要素とオペレーション引数

TargetLink では、AUTOSAR 規格に示されたとおり、ポインタを介して配列タイプを配列の最初のスカラー要素に渡すことができます。これは、RTE API のパラメータや戻り値に適用されます。

コード例: コードの可読性を向上させるため、TargetLink では次の補助ポインタをローカルで導入します。

RTE API 関数に渡される 2次元配列

```
Rte_Write_SP_MyDataElem(&A[0][0]);
```

非 RTE API 関数に渡される 2次元配列

```
CalcSRV(A);
```

RTE API 関数によって返される基本タイプ sint16 および 8 列の 2次元配列

```
p_MyDataElem =
    (const sint16 (*)[8]) Rte_IRead_Run_RP_MyDataElem();
...
Sa4_Selector = p_MyDataElem[IndexX][IndexY];
```

種類が ADDRESS の非 RTE API アクセス関数を含む 2次元配列

```
Sa4_Selector = (GetAddressOfA())[IndexX][IndexY];
```

AUTOSAR エクスポート

バージョン固有のエクスポート

AUTOSAR 4.x 規格に IncludedDataTypeSets を導入する場合、Data Dictionary の[Subsystems]領域のコード表現は、コードを生成した AUTOSAR バージョンによって異なります。

それに応じて、Data Dictionary からの AUTOSAR エクスポートは、次の互換性対応表に従います。

コードを生成したバージョン	AR 2.x エクスポート	AR 3.x エクスポート	AR 4.x エクスポート
AUTOSAR 2.x	可	可	不可
AUTOSAR 3.x	可	可	不可
AUTOSAR 4.x	不可	不可	可

2 次元配列のエクスポート

AUTOSAR 4.x では、2 次元配列のデータタイプはネスト構造のインプリメンテーションデータタイプとしてエクスポートされます。

Container Manager の個別インストール

dSPACE の Container Manager が個別にインストールされるようになり、TargetLink を削除してもシステムに残るようになりました。

カスタムワークフロー定義(CTW)ファイルの提供 カスタムワークフロー定義ファイルを利用すると、ワークフロールールを変更することができます。



ワークフロールールの変更は経験豊富なユーザのみが行うようにしてください。ワークフロールールを定義するための特別な構文が存在します。

1. TargetLink の「`tlCustomizationFiles`」API 関数を介して CTW テンプレートのコピーを提供し、`<MyDir>`に保存します。
2. 必要に応じて、CTW ファイルのワークフロールールを変更します。
3. Data Dictionary Manager で、`<MyDir>\<CTW file name>`へのパスを、`/Pool/Autosar/Config/ContainerExchange/オプション設定`に含まれる `WorkflowDefinitionFile` プロパティの値、または「`tl_export_container`」API 関数の `WorkflowDefinitionFile` プロパティの値として指定します。

関連ドキュメント

- 「Preparing Container Export in TargetLink」([📖『Container Management Document』](#))
- 「Advanced: Configuring Container Handling」([📖『Container Management Document』](#))

関連トピック

リファレンス

- 「`tl_export_container`」([📖『TargetLink API Reference』](#))

アプリケーションデータタイプ

TargetLink 4.0 には、AUTOSAR 規格に準拠して以下の変更が加えられています。

InvalidValue、ScalingRef、UnitRef の各プロパティは、Kind プロパティが Primitive に設定された ApplicationDataType オブジェクト (ADT) のみ設定することができます。

DD オブジェクト	DD プロパティ		
	InvalidValue	ScalingRef	UnitRef
Kind プロパティが Array または Record に設定された ApplicationDataType オブジェクト ApplicationDataTypeComponent	不許可 削除済み	不許可 削除済み	不許可 削除済み

整合性チェック 従来の Data Dictionary を最新のデータモデルリビジョンにアップグレードする際に、TargetLink で整合性チェックが実行されません。

配列 ADT	レコード ADT
<ol style="list-style-type: none"> 1. TargetLink は、配列 ADT にある InvalidValue、ScalingRef、UnitRef の各プロパティの値が、ADT で参照されるプリミティブ ADT のこれらのプロパティの値と一致しているかどうかをチェックします。 2. これらの値が一致している場合、配列 ADT で InvalidValue、ScalingRef、UnitRef の各プロパティの設定が解除されます。 3. 値が一致しない場合、これらのプロパティはそのまま残されます。 妥当性確認時にエラーが発生します。	<ol style="list-style-type: none"> 1. TargetLink は、レコード ADT の各 ApplicationDataTypeComponent オブジェクトにある InvalidValue、ScalingRef、UnitRef の各プロパティの値が、各コンポーネントで参照されるプリミティブ ADT の値と一致しているかどうかをチェックします。 2. これらの値が一致している場合、InvalidValue、ScalingRef、UnitRef プロパティは各コンポーネントから削除されます。 3. 値が一致しない場合、これらのプロパティは各コンポーネントにカスタムプロパティとして残ります。 妥当性確認時にエラーは発生しません。



TargetLink で不整合が検出された場合は、競合を手作業で解消し、AUTOSAR 規格に準拠した指定内容にしてください。

配列および行列タイプの制約

AUTOSAR インターフェースで使用する AUTOSAR の配列／行列タイプ (AutosarArrayWidth が > 0 のタイプ) に対して、Constraints オブジェクトを作成する必要がなくなりました。コード生成時に、TargetLink は、プリミティブデータタイプで定義された制約を利用します。

次の場合は、引き続き配列／行列タイプに対して Constraints オブジェクトを作成する必要があります。

- タイプが AUTOSAR 以外のインターフェースで使用される場合。
- タイプが標準コード生成モードで使用される場合。

置き換えられた IsQueued プロパティ

DataElement オブジェクトでは、以下の変更が行われています。

削除されたプロパティ	代替プロパティ	互換設定値
IsQueued	ImplementationPolicy	IsQueued を <i>off</i> に設定していた場合は <i>standard</i> IsQueued を <i>on</i> に設定していた場合は <i>queued</i>



TargetLink 4.0 より前の TargetLink バージョンで作成された Data Dictionary ファイルは、自動的に移行されます。

その他

項目の一覧

本章の内容

移行に関するその他の注意点	212
バス診断のより厳密な設定	213
プロットチャンネルの指定	215
信号プロパティの継承	216

移行に関するその他の注意点

OpenFcn コールバック

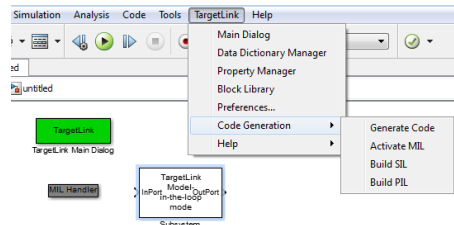
Simulink では、Commented ブロックプロパティを直接的または間接的に変更する OpenFcn コールバックを使用することはできません。TargetLink 4.0 では、TargetLink サブシステムを SIL または PIL シミュレーションモードに切り替えるときに Commented ブロックプロパティを使用するため、コード生成の開始などのために作成されたカスタムボタンは機能しなくなります(「TargetLink のシミュレーションフレームの改善」(163 ページ))。

以下の API 関数で Commented ブロックプロパティを変更します。これらの関数は、OpenFcn コールバックで直接的または間接的に使用することはできません。

- 「t1_generate_code」
- 「t1_build_host」
- 「t1_build_target」
- 「t1_set_sim_mode」

解決策 回避策として以下の方法があります。

- 以下の「TargetLink Utility Blocks」(☞『TargetLink Block and Object Reference』)を使用します。
 - MIL mode(「MIL Mode Block」(☞『TargetLink Block and Object Reference』)を参照)ブロック
 - SIL mode(「SIL Mode Block」(☞『TargetLink Block and Object Reference』)を参照)ブロック
 - Tool Selector(「Tool Selector Block」(☞『TargetLink Block and Object Reference』)を参照)ブロック
- モデルウィンドウの[TargetLink]メニューを使用します。



- TargetLink Main Dialog(「Code Generation Page (Main Dialog Block)」(図 1)『TargetLink Block and Object Reference』を参照)ブロックの制御を使用します。
- ClickFcn コールバックを含む注釈を使用します。
 1. Simulink モデルウインドウの何もない領域をダブルクリックして、注釈を作成します。
 2. 必要に応じてテキストを入力します。
 3. 注釈を選択して、コンテキストメニューから[Properties...]を選択します。[Annotation properties]ダイアログが開きます。
 4. [ClickFcn]グループボックスで、注釈によって実行するコマンドを指定します。
 5. [Appearance]グループボックスで、注釈にボタンのような外観を指定することもできます。
 6. [OK]をクリックしてダイアログを閉じます。
 これで、注釈をシングルクリックしてコマンドを実行することができます。
- コード生成処理の開始のための制御を含むユーザ固有の GUI を記述します。
- Simulink のマニュアルに記載されるように、ユーザのスクリプトを開始するユーザ固有の Simulink メニューを作成します。

バス診断のより厳密な設定

設定の自動調整

TargetLink では、以下のような状況の場合に一部の Simulink 設定パラメータの値を自動的に変更することがあります。

- モデルの準備中
- モデルのアップグレード中
- TargetLink Main Dialog ブロックをモデルに追加するとき
- MIL Handler ブロックをモデルに追加するとき

これにより、ロギングやオーバーフロー検出など、TargetLink の MIL シミュレーション機能をフルに使用することができます。

Simulink 設定パラメータ	TargetLink で設定される値
Element name mismatch	error ¹⁾
Mux blocks used to create bus signals	error ¹⁾
Bus signal treated as vector	error ¹⁾

Simulink 設定パラメータ	TargetLink で設定される値
Non-bus signals treated as bus signals	error ¹⁾
Signal storage reuse	off
Block reduction	off
Signal logging format	Dataset

¹⁾ この環境設定を「error」以外の値にリセットすると、シミュレーション開始時またはコード生成中にメッセージ E02462 が表示されません。

必要なモデルパラメータ [Diagnostics] - [Connectivity] - [Buses]にある複数の Simulink モデル設定パラメータは、次の表に従って設定します。

パラメータ	設定
Element name mismatch	error
Mux blocks used to create bus signals	error
Bus signal treated as vector	error
Non-bus signals treated as bus signals	error

TargetLink では、これらのパラメータ設定をモデルの初期化中にチェックします。これらの設定が存在しない場合は、エラーメッセージ「E02462」が表示され、コード生成を行うことはできません。

さらに、以下の TargetLink の MIL シミュレーション機能もロックされ、メッセージ「E02464」が表示されます。

- 信号のロギング
- 最小値／最大値のロギング
- 信号のプロット
- オーバーフローの検出

API これらのプロパティを API 経由で Simulink の API 関数 `get_param/set_param` を使用して設定するには、以下の設定を行います。

プロパティ	値
StrictBusMsg	'ErrorOnBusTreatedAsVector'
NonBusSignalsTreatedAsBus	'error'
BusObjectLabelMismatch	'error'



Simulink の Bus to Vector ブロックは、ベクトルとして扱われるバス信号と、バス信号パラメータ設定として扱われる非バス信号と競合します。このブロックは使用しないでください。

プロットチャンネルの指定

プロットチャンネルの新しい指定方法

TargetLink では、次の 2 つの方法で行列信号にプロットチャンネルを指定することができます。

- 行列信号要素の線形インデックス(列順序)
- 各行列信号要素に 1 ベースのインデックスペア([row column])



行列信号のプロットチャンネルの指定には、インデックスペアのアプローチを使用することをお勧めします。

インデックスペアのアプローチの例 プロットチャンネル [1,4; 2,3; 5,6] で、シミュレーション中に、3 つの行列要素(1,4)、(2,3)、(5,6)を指定します。

移行の考慮



行列信号に、1 ベースのインデックスペアを 1 つのみ含むプロットチャンネルを指定すると、以下のいずれかを表すことができるため、あいまいになります。

- 2 つの行列信号要素(線形インデックスの列単位)
- 1 つの行列信号要素(インデックスペアの行/列)

TargetLink では、このあいまいさを解決するために、このようなプロットチャンネルの指定がインデックスペアであるものとして解釈します。

TargetLink のバージョン 3.4 以前で作成されたモデルでは、モデルに行列([1xn])または列行列([mx1])が含まれる場合に、問題が発生する可能性があります。

問題	説明
2 つではなく 1 つの信号がプロットされる	行/列行列信号に接続される TargetLink ブロックで、2 つではなく 1 つの信号がプロットされます。
プロットチャンネルが無効	無効なプロットチャンネルが検出され、警告「W02441」(☞『TargetLink Message Reference』)が表示されます。

無効なプロットチャンネルの例 [6x1] 行列信号が、プロットチャンネルが [3 5] と定義される TargetLink ブロックに接続されます。

解決策

多くの場合、問題の原因は InPort ブロックまたは Constant ブロック から発生する信号にあります。

原因	解決策
InPort ブロック	信号の次元を明示的に指定して、それをベクトルにします。
Constant ブロック	['Interpret as 1-D'] チェックボックスをオンにします。

信号プロパティの継承

TargetLink のコードジェネレータで、さまざまなデータタイプの複数の信号の多重化や連結を行うことができなくなりました。

例 ブロック A とブロック B にさまざまなデータタイプの出力変数があるとします。それらの出力信号は多重化されてブロック C に取り込まれ、その信号の継承は有効化されます。

TargetLink 3.5 以前	TargetLink 4.0
ブロック C の 2 つの出力変数のそれぞれに、継承されるデータのいずれかが含まれます	「E15738」(📖 『TargetLink Message Reference』)

解決策 Mux ブロックの代わりに Bus Creator ブロックを使用します。

API コマンド

TargetLink と TargetLink Data Dictionary API 関数の変更

名前が変更された API コマンド

次の TargetLink API コマンドが置き換えられました。

TargetLink 3.5 以前	TargetLink 4.0 ¹⁾
<code>tl_extract_subsystem</code>	<code>tlExtractSubsystem</code> (📖 『TargetLink API Reference』)
<code>tl_sim_interface</code>	<code>tlSimInterface</code> (📖 『TargetLink API Reference』)
<code>tl_switch_blockset</code>	<code>tlOperationMode</code> (📖 『TargetLink API Reference』)
<code>tl_upgrade</code>	<code>tlUpgrade</code> (📖 『TargetLink API Reference』)

¹⁾ これらのコマンドは、大文字／小文字が区別されます。

廃止事項

項目の一覧

本章の内容

廃止された TargetLink の機能	217
廃止された Data Dictionary 機能	217
廃止された制限事項	218
廃止された API 関数	218

廃止された TargetLink の機能

廃止された TOM

このバージョンの TargetLink では、次の TOM を使用することはできません。

- HCS12:
 - HCS12/Cosmic 4.7
 - HCS12/CodeWarrior 3.1
- MPC5xx:
 - MPC5xx/Green Hills 5.1
 - MPC5xx/WindRiver Diab 5.7
- MPC55xx:
 - MPC55xx/WindRiver Diab 5.7
- M32R:
 - M32R/GAIO 9

廃止された Data Dictionary 機能

Data Dictionary Manager でのユーสケースフィルタの使用

Data Dictionary Manager のユースケースフィルタ機能は廃止されました。独自の XML ベースのフィルタルールセットを使用および指定することができます。詳細については、「Data Dictionary の改善点」(170 ページ)を参照してください。

廃止された制限事項

TargetLink 4.0 では、以前の TargetLink バージョンの以下の制限事項がなくなりました。

全般的な制限事項

多次元信号

TargetLink では、ブロック出力において多次元信号（行列信号）はサポートされません。行列パラメータは、Look-Up Table (2-D) ブロック、Interpolation Using Prelookup ブロック、Direct Look-Up Table (n-D) ブロック、Discrete State-Space ブロック、および Custom Code ブロックでのみサポートされています。なお、多次元パラメータは使用することができません。

Stateflow では、インターフェース中に多次元信号を含めることはできませんが、ローカル変数に行列を使用することはできます。Stateflow の制限事項の詳細については、「Stateflow Limitations」(E)『TargetLink Orientation and Overview Guide』を参照してください。



この TargetLink バージョンで、2-D 行列信号がサポートされるようになりました。ただし、3-D 信号（以上）はサポートされません。

ブロック固有の制限事項

Gain ブロック

TargetLink でこれらのブロックがサポートされるのは、Simulink のブロックダイアログで Multiplication パラメータが Element-wise(K.*u) (デフォルト) に設定されている場合に限られます。

TargetLink AUTOSAR モ

ジュールの制限事項

モデルリンクコード表示

TargetLink では、AUTOSAR 通信が有効になったブロックでのモデルリンクコード表示をサポートしていません。

フレームモデルのアップデート/名前の変更

AUTOSAR データから生成したフレームモデルをアップデートする場合に、TargetLink は名前が変更された AUTOSAR エlement を新規 Element として処理します。TargetLink では、モデルの古い名前を引き継いでいるブロックを削除して、新しい名前のブロックを追加するように指示します。

廃止された API 関数

関数	ステータス	代替プロパティ
build_customcode_sfcn	Error ¹⁾	「tl_build_customcode_sfcn」
get_mdltldata	Error ¹⁾	-

関数	ステータス	代替プロパティ
get_sfobjects	Error ¹⁾	「tl_get_sfobjects」
get_tlblocks	Error ¹⁾	「tl_get_blocks」
tl_adapt_dd_references	Warning ²⁾	「tlMoveDDObject」
tl_extract_subsystem	Warning ²⁾	「tlExtractSubsystem」
tl_find_system	Error ¹⁾	find_system
tl_get_blockset_mode	Warning ²⁾	「tlOperationMode」
tl_get_project	Error ¹⁾	dsdd_manage_project('GetProjectFile', ...)
tl_set_project	Error ¹⁾	dsdd_manage_project('SetProjectFile', ...)
tl_sim_interface	Warning ²⁾	「tlSimInterface」
tl_switch_blockset	Warning ²⁾	「tlOperationMode」

¹⁾ この関数は TargetLink から削除されました。

²⁾ この関数は廃止され、TargetLink の今後のバージョンでは削除されます。

互換性の考慮 これに応じて、ユーザスクリプトとツールチェーンを適合させてください。

メッセージ

メッセージの変更

メッセージタイプの変更

TargetLink 4.0 では、次のメッセージのタイプが *Advice* に変更されました。これらのメッセージはコードの効率性向上に関する通知で、無視しても問題ありません。

従来のメッセージ番号	新しいメッセージ番号
W15658	「A15658」
N17350	「A17350」
W17352	「A17352」
W17358	「A17358」
W17356	「A17356」

Stateflow 関連の変更

項目の一覧

本章の内容

エクスポートされたグラフィカル関数	220
Stateflow の行列	221

エクスポートされたグラフィカル関数

SIL/PIL でのシミュレーション

TargetLink のシミュレーションフレームの改良(「TargetLink のシミュレーションフレームの改善」(163 ページ)を参照)により、TargetLink サブシステムの外部で使用するエクスポートされたグラフィカル関数が TargetLink サブシステム内で定義されている場合、このエクスポートされたグラフィカル関数は SIL または PIL シミュレーションモードでシミュレートすることができません。

これは、SIL または PIL シミュレーションモードで TargetLink サブシステムにコメントが付くためです。

コード生成

TargetLink では、エクスポートされたグラフィカル関数に対する量産コードを生成することができます。

Stateflow Coder では、コメントの付いたモデルパーツ内で定義された、エクスポートされたグラフィカル関数をビルドすることができません。そのため、SIL または PIL シミュレーションモードに設定された TargetLink サブシステム内で定義された、エクスポートされたグラフィカル関数をビルドすることはできません。

モデリングに関する推奨事項

TargetLink サブシステムの外部でのみエクスポートされたグラフィカル関数を使用する場合は、TargetLink サブシステム内でこれらを定義しないでください。これにより、デッドコードを回避することができます。

Stateflow の行列

ループの生成

TargetLink で、Stateflow 行列と Simulink 行列に同じメカニズムと最適化を使用できるようになりました。それに応じて、両方の次元を生成された量産コードのループに含めることができます。

TargetLink 4.0	TargetLink 3.x
for ループとして生成される外部次元の反復ローリングは「LoopUnrollThreshold」に依存	do-while ループとして生成される外部次元の反復「LoopUnrollThreshold」への依存なし

可能なコード変更

- マージループ
- ベクトル／行列のその他の補助変数
- 従来の do while ループを for ループにすることができます。

関連ドキュメント

- 「Code Pattern for Vectors and Matrices」([📖](#) 『TargetLink Preparation and Simulation Guide』)
- 「Merging consecutive loops」([📖](#) 『TargetLink Customization and Optimization Guide』)

TargetLink の今後のバージョンでの変更予定

廃止予定

A2L のインポート

A2L のインポートは、今後の TargetLink バージョンで廃止される予定です。

RTF ドキュメントの生成

リッチテキスト形式 (RTF) でドキュメントを生成するオプションは、今後の TargetLink バージョンで廃止される予定です。

VEOS

項目の一覧

本章の内容

VEOS 3.3 の新機能	223
VEOS 3.3 への移行	224

VEOS 3.3 の新機能

LIN バスシミュレーションのサポート

VEOS の LIN バスシミュレーションでは、以下の機能をサポートしていません。

- LIN スレーブノードのシミュレーション
LIN マスターノードと LIN スレーブノードの生成に同じ ECU パラメータの定義が使用されます。
- 1 つの V-ECU での複数の LIN マスターノードと LIN スレーブノードのシミュレーション
- LIN スレーブノード間の通信のシミュレーション
- バス上での LIN メッセージの衝突のシミュレーション
- LIN バス転送回数に準拠したシミュレーション
- ブロードキャストフレームのシミュレーション

オフラインシミュレーションで V-ECU の変数にスティミュラス信号を入力

VEOS でのオフラインシミュレーションで、ControlDesk の Signal Editor を使用して V-ECU の変数にスティミュラス信号を入力することができます。「Signal Editor の新機能 (ControlDesk 5.3)」(93 ページ)も参照してください。

サポートされるコンパイラ

VEOS 3.3 で Microsoft Visual C/C++ Compiler (MSVC) 11 (Microsoft Windows® SDK 8.1 により提供) がサポートされるようになりました。



MSVC 9.0 のサポートは VEOS 3.3 で廃止されています。

VEOS 3.3 への移行

互換性一覧

VEOS と、OSA ファイルおよび V-ECU インプリメンテーションとの互換性を次の表に示します。

	OSA ファイルを作成した製品の dSPACE Release			V-ECU インプリメンテーションを作成した製品の dSPACE Release		
	2013-B 以前	2014-A	2014-B	2013-B 以前 ¹⁾	2014-A ²⁾	2014-B ³⁾
VEOS 3.2	… ⁴⁾	✓ ⁵⁾	–	✓	✓	–
VEOS 3.3	… ⁴⁾	✓	✓	✓	✓	✓

¹⁾ V-ECU インプリメンテーションバージョン 1.0

²⁾ V-ECU インプリメンテーションバージョン 2.0

³⁾ V-ECU インプリメンテーションバージョン 2.1

⁴⁾ 移行の詳細については、dSPACE Release 2014-A の『**新機能と移行手順**』を参照してください。このドキュメントは、<http://www.dspace.jp/goto.cfm/supver.rcphil> からダウンロードすることができます。

⁵⁾ dSPACE Release 2014-A の製品で作成され、VEOS 3.3 で変更された OSA ファイルは、VEOS 3.2 でロードすることができません。

互換性情報

項目の一覧

本章の内容

サポートしている MATLAB リリース	226
オペレーティングシステム	227
dSPACE ソフトウェアのランタイム互換性	230
Windows(64 ビット版)で dSPACE ソフトウェア(32 ビットバージョン)を使用する場合の制限事項	231
64 ビット dSPACE DVD に含まれる製品についての制限事項	231
Windows 7 の場合の制限事項	235

サポートしている MATLAB リリース

サポートしている MATLAB リリース

MATLAB のリリース	dSPACE Release 2014-B の各コンポーネントによるサポート						
	32 ビット DVD に収録の RCP and HIL Software (MATLAB 32 ビットバージョンに対応)	64 ビット DVD に収録の RCP and HIL Software (MATLAB 64 ビットバージョンに対応)	AutomationDesk 4.1 ¹⁾²⁾	TargetLink 4.0	Model Compare 2.3	VEOS 3.3	dSPACE Python Extensions 1.7 ³⁾
R2014b	✓	✓	✓	✓	✓	✓	✓
R2014a	✓	✓	✓	✓	✓	✓	✓
R2013b	✓ ⁴⁾	✓ ⁴⁾	✓	✓	✓	✓	✓
R2013a	✓ ⁵⁾	✓	✓	✓	✓	✓	✓

¹⁾ AutomationDesk の MATLAB Access ライブラリには MATLAB が必要です。

²⁾ ご使用のホスト PC に同じ MATLAB リリースの 32 ビットバージョンが並行してインストールされている場合は、32 ビットバージョンで MATLAB の 64 ビットバージョンもサポートされます。詳細については、本表下の注を参照してください。

³⁾ matlablib2 of dSPACE Python Extensions requires MATLAB.

⁴⁾ RTI FPGA Programming Blockset - FPGA Interface ではサポートされません。

⁵⁾ R2013a: dSPACE Automotive Simulation Models (ASMs) とともに使用する場合はパフォーマンスの問題により、ASM と R2012b または R2013a を使用する前に次のバグフィックスをインストールすることをお勧めします。

<http://www.mathworks.com/support/bugreports/916069>

dSPACE ソフトウェアと組み合わせて使用可能なその他の MATLAB Release の最新情報については、

http://www.dspace.jp/goto.cfm/jp_compati_RCP_HIL を参照してください。

MATLAB R2014a のサポート MATLAB R2014a では、Simulink Coder によるコード生成が変更されています。これに従い、RTI、ConfigurationDesk および VEOS による変数記述ファイル (TRC ファイル) の生成も変更する必要があります。これらの変更は、dSPACE Release 2014-A で完全に考慮されているわけではありません。そのため、dSPACE Release 2014-A は MATLAB R2014a を公式にはサポートしていません。ただし、従来のコードと TRC ファイルの生成動作を使用できる特殊な MATLAB コマンドがあります。このように、dSPACE Release 2014-A では、正式ではありませんが MATLAB R2014a をサポートしています。詳細については、<http://www.dspace.com/ja/jpn/home/support/supvers/supverscompmlcomp/compmlr2014/dsml2014a.cfm> を参照してください。

オペレーティングシステム

ホスト PC のオペレーティングシステム

Release 2014-B の dSPACE 製品では、次のオペレーティングシステムをサポートしています。

32 ビットバージョン dSPACE ソフトウェア	64 ビットバージョン dSPACE ソフトウェア
<ul style="list-style-type: none"> Windows 7 Professional、Ultimate、Enterprise Service Pack 1 (32 ビット版または 64 ビット版) 上記のエディションのみサポートされます。Windows 7 Home および Starter エディションはサポートされません。 	<ul style="list-style-type: none"> Windows 7 Professional、Ultimate、Enterprise Service Pack 1 (64 ビット版) 上記のエディションのみサポートされます。Windows 7 Home および Starter エディションはサポートされません。

32 ビットバージョン dSPACE ソフトウェア	64 ビットバージョン dSPACE ソフトウェア
注意点と制限事項	
<ul style="list-style-type: none"> ■ Windows 7 を使用する場合は、制限事項が適用されます。「Windows 7 の場合の制限事項」(235 ページ)を参照してください。 ■ 64 ビットオペレーティングシステムのサポート: 32 ビットバージョンの dSPACE ソフトウェアは、Windows 7 の 64 ビット版のみサポートします。その他の 64 ビット版オペレーティングシステム (Windows XP および Windows Vista) はサポートされません。 32 ビットバージョンの dSPACE ソフトウェアは、Windows オペレーティングシステム (64 ビット版) の WoW64 (Windows-on-Windows 64-bit) サブシステムで動作します。 Windows (64 ビット版) で 32 ビットバージョンの dSPACE ソフトウェアを使用する場合は、制限事項が適用されます。「Windows (64 ビット版) で dSPACE ソフトウェア (32 ビットバージョン) を使用する場合の制限事項」(231 ページ)を参照してください。 ■ ControlDesk Next Generation は、MicroAutoBox Embedded PC (Microsoft Windows 7 Ultimate (32 ビット版) で稼動) にもインストールすることができます。 	<ul style="list-style-type: none"> ■ Windows 7 を使用する場合は、制限事項が適用されます。「Windows 7 の場合の制限事項」(235 ページ)を参照してください。 ■ TargetLink、Model Compare および RCP and HIL Software パッケージの各種製品のみ、64 ビットバージョンをご利用いただけます。これらの 64 ビットバージョンを使用する場合は、制限事項が適用されます。詳細については、「64 ビット dSPACE DVD に含まれる製品についての制限事項」(231 ページ)を参照してください。

64 ビット版の Windows オペレーティングシステムで動作する 32 ビットバージョンの dSPACE ソフトウェア dSPACE ソフトウェアは、Windows オペレーティングシステム (64 ビット版) の WoW64 (Windows-on-Windows 64-bit) サブシステムで 32 ビット版アプリケーションとして動作します。WoW64 は、Windows 64 ビット版で Windows 32 ビット版ベースのアプリケーションをシームレスに実行できるようにするための Windows の x86 エミュレータです。これにより、大きなメモリ領域を使用できるようアプリケーションが準備されている場合は、32 ビット版の各プロセスで最大 4 GB の仮想メモリを使用できるようになります。そうでない場合、プロセスの仮想アドレス空間は 2 GB に制限されます。

ファイアウォールルールを追加して通信を許可

各種 dSPACE ソフトウェア製品のインストール時には、Windows のファイアウォールルールが追加してインストールされます。たとえば、あるルールによって AutoBox などの dSPACE 拡張ボックスとの通信を行い、また他のルールによって MotionDesk でネットワークチャネルからモーションデータを受信します。これらのルールは、次のコマンドで生成されます。

- netsh advfirewall firewall add rule name="dSPACE Net Service"
 service=any dir=in action=allow profile=any
 protocol=icmpv4:0, any description="Allow the dSPACE Net Service to connect to a dSPACE expansion box via network."
- netsh advfirewall firewall add rule name="dSPACE MotionDesk"
 program="%dspace_root%\MotionDesk\Bin\MotionDesk.exe"
 dir=in action=allow profile=any description="Allow dSPACE MotionDesk to receive motion data via network."

ホスト PC でサードパーティ製ファイアウォールソフトウェアを実行している場合は、dSPACE ソフトウェアの TCP/IP 通信がブロックされないかどうか確認してください。

dSPACE License Server のオペレーティングシステム

フローティングネットワークライセンスを購入した場合は、ネットワーク接続されている PC の 1 台を dSPACE License Server としてインストールおよび設定する必要があります。

dSPACE License Server のオペレーティングシステムは、次のいずれかである必要があります。

- Windows XP Professional (32 ビット版) Service Pack 3
- Windows Vista Business、Ultimate、または Enterprise (32 ビット版または 64 ビット版) 最新のサービスパック
- Windows 7 Professional、Ultimate、または Enterprise (32 または 64 ビット版) 最新のサービスパック
- Windows Server 2003 (32 ビット版または 64 ビット版)
- Windows Server 2008 R2
- Windows Server 2012



dSPACE License Server は Windows 以外のオペレーティングシステムをサポートしていません。

dSPACE ソフトウェアのランタイム互換性

定義

ランタイム互換性とは、以下のことを意味します。

- 別々のフォルダにインストールされている場合でも、複数の dSPACE 製品の同時使用が可能
- 相互作用なく個別に dSPACE 製品を使用可能

dSPACE Release 2014-B の製品互換性

ランタイム互換性を最大限に確保するため、dSPACE では、同一の dSPACE Release のソフトウェア製品のみを使用することをお勧めします。次の点に注意してください。

- 異なる dSPACE Release の製品を併用した場合、dSPACE ツールチェーンでランタイム互換性に関連する制限が生じる可能性があります。

dSPACE 製品が(自動化インターフェースなどを介して)直接連携する場合や、(.A2L のような共通のファイルタイプなどを介して)間接的に連携する場合は、制限事項が適用されることがあります。です。詳細な制限事項については、該当する製品のマニュアルを参照してください。主要な制限事項については、次を参照してください。

まれに、ランタイム互換を実行するために製品に追加のパッチをインストールする必要がある場合があります。パッチが必要かどうかについて、およびパッチ自体の情報については、<http://www.dspace.com/ja/jpn/home/support/patches/suppatchrelease.cfm> を参照してください。

- Release 2014-B の RCP and HIL Software 製品は、それより前の dSPACE Release の RCP and HIL Software 製品と併用することはできません。

TargetLink および Model Compare に関する主要な制限事項 ビット互換のある MATLAB バージョン(32 ビットまたは 64 ビット)のみ使用可能であるため、64 ビットバージョンの TargetLink を 32 ビットバージョンの Model Compare と併用することはできません。また、32 ビットバージョンの TargetLink を 64 ビットバージョンの Model Compare と併用することはできません。

SCALEXIO システムの使用に関する主要な制限事項 SCALEXIO システムで使用する製品には互換性が必要です。同一の dSPACE Release で提供される製品のみ、互換性が保証されます。ご不明な点がございましたら、dSPACE にお問い合わせください。

以前のリリースの dSPACE 製品との併用

以前のリリースの複数の製品を併用する場合の詳細と注意事項については、http://www.dspace.jp/goto.cfm/ja_0501 を参照してください。

Windows(64ビット版)で dSPACE ソフトウェア(32 ビットバージョン)を使用する場合の制限事項

目的	Windows(64ビット版)で dSPACE ソフトウェア(32 ビットバージョン)を使用する場合は、さらにいくつかの制限事項が適用されます。
デバイスドライバの制限事項	メーカーから 64 ビット版ドライバが提供されている場合のみ、サードパーティ製バスインターフェース(CAN、LIN、または FlexRay)がサポートされます。
TargetLink:ターゲットコンパイラの制限事項	特定のターゲットコンパイラのサポート情報については、当該のコンパイラメーカーにお問い合わせください。
MATLAB	MATLAB の 32 ビットバージョンを Windows 7(64 ビット版)にインストールすると、MATLAB のインストールプログラムにより MATLAB の 64 ビットバージョンが提供されていることを示すメッセージが表示されます。MATLAB の 32 ビットバージョンをインストールするには、[OK]をクリックします。

64 ビット dSPACE DVD に含まれる製品についての制限事項

目的	<p>基本的に、64 ビット dSPACE DVD セットには、32 ビット dSPACE DVD セットと同じ製品が含まれています。ただし、64 ビット DVD セットには次の製品が含まれています。</p> <ul style="list-style-type: none"> ■ MATLAB の 64 ビットバージョンをサポートするよう移植された、MATLAB を使用するすべての dSPACE 製品 ■ MATLAB の 64 ビットバージョンをサポートするすべての 32 ビットバージョンの dSPACE 製品 ■ MATLAB に関連しない 32 ビットバージョンの dSPACE 製品すべて (ControlDesk Next Generation など) <p>64 ビット DVD セットを使用する場合は、以下に記載の制限事項にも注意する必要があります。</p>
----	---

64 ビットに移植された
dSPACE ソフトウェア製品

次の表に、MATLAB をサポートするすべての dSPACE 製品の一覧と、64 ビット dSPACE DVD セットから 64 ビットバージョンが利用可能かどうかを示します。



MATLAB サポートがない dSPACE 製品 (ControlDesk Next Generation など) は、32 ビットバージョンのみ含まれていることにご注意ください。

MATLAB サポートがある dSPACE 製品	64 ビット DVD セットに含まれるバージョン	
	64 ビットバージョン	32 ビットバージョン
AutomationDesk	–	✓ ¹⁾
TargetLink	✓	–
Model Compare	✓	–
VEOS	–	✓ ²⁾
Platform API Package		
	dSPACE Python Extensions	✓
	HIL API .NET	✓
	XIL API .NET 2.0 MAPort	✓

MATLAB サポートがある dSPACE 製品		64 ビット DVD セットに含まれるバージョン	
		64 ビットバージョン	32 ビットバージョン
RCP and HIL Software パッケージ	RTI および RTI-MP	✓	—
	RTI Gigalink Blockset	✓	—
	RTI CAN Blockset	✓	—
	RTI CAN MultiMessage Blockset	✓	—
	RTI LIN MultiMessage Blockset	✓	—
	RTI FlexRay Configuration Blockset	✓	—
	RTI FPGA Programming Blockset	✓	—
	RTI Electric Motor Control Blockset	✓	—
	RTI Ethernet Blockset	✓	—
	RTI Ethernet UDP Blockset	✓	—
	RTI XCP on Ethernet Blockset	✓	—
	RTI Watchdog Blockset	✓	—
	RTI RapidPro Control Unit Blockset	✓	—
	RTI Bypass Blockset	✓	—
	RTI USB Flight Recorder Blockset	✓	—
	ConfigurationDesk	—	✓ ²⁾
	FlexRay Configuration Blockset	✓	—
	FlexRay Configuration Tool	—	✓ ³⁾
	ModelDesk	—	✓ ²⁾
	Automotive Simulation Model	✓	—
	MotionDesk	—	✓ ²⁾
	MotionDesk Blockset	✓	—
	Flight Rec Data Merger	—	✓ ³⁾
RCP and HIL Software パッケージ	—	✓ ³⁾	
その他の製品			

¹⁾ ご使用のホスト PC に同じ MATLAB リリースの 32 ビットバージョンが並行してインストールされている場合は、32 ビットバージョンで MATLAB の 64 ビットバージョンもサポートされます。詳細については、本表下の注を参照してください。

²⁾ 32 ビットバージョンでは MATLAB の 64 ビットバージョンもサポートされます。

³⁾ この製品は MATLAB アーキテクチャ(32 ビット/64 ビット)に依存しません。

その他の製品固有の制限事項

AutomationDesk、dSPACE Python Extensions ご使用のホスト PC に同じ MATLAB リリースの 32 ビットバージョンが並行してインストールされている場合は、AutomationDesk および dSPACE Python Extensions の 32 ビットバージョンでも MATLAB の 64 ビットバージョンがサポートされます。両方の MATLAB バージョン(32 ビットおよび 64 ビット)が、dSPACE Installation Manager で対応するインストールセットに接

続されている必要があります。64ビットバージョンを優先接続として指定する必要があります。この場合は、MATLAB Access Library (AutomationDesk) および matlablib2 Python モジュール (Python Extensions) でも、バージョン 5.0 以降のファイル形式の MAT ファイルの読み書きをサポートします。

MAT ファイルのサポートの制限 次の製品は、ファイル形式バージョン 5.0 の MAT ファイルの読み書きのみサポートしています。このバージョンの MAT ファイルは、MATLAB で save コマンドの '-v6' オプションを使用して作成することができます。

- ModelDesk 4.0 (Maneuver Editor、Road Generator)
- ControlDesk Next Generation (ControlDesk 5.3)
- 32 ビットバージョンの MATLAB がインストールされていない AutomationDesk 4.1 (MATLAB Access Library)



ご使用のホスト PC に同じ MATLAB リリースの 32 ビットバージョンが 64 ビットバージョンとともにインストールされている場合は、MATLAB Access Library ではバージョン 5.0 以降のファイル形式の MAT ファイルの読み書きがサポートされます。上記の説明を参照してください。

- 32 ビットバージョンの MATLAB がインストールされていない matlablib2 Python モジュール



ご使用のホスト PC に同じ MATLAB リリースの 32 ビットバージョンが 64 ビットバージョンとともにインストールされている場合は、matlablib2 Python モジュールではバージョン 5.0 以降のファイル形式の MAT ファイルの読み書きがサポートされます。上記の説明を参照してください。

dSPACE HIL API .NET dSPACE HIL API .NET では、MATLAB の 64 ビットバージョンはサポートされていません。

RTI-MP rtimpdiag コマンドは機能しません。このコマンドは、MATLAB の 64 ビットバージョンがサポートされていない dSPACE HIL API .NET をベースとしています。

MATLAB の 32 ビットバージョンと 64 ビットバージョンの並行インストール 64 ビット版のオペレーティングシステムに、R2013b などの特定の MATLAB Release の、32 ビットバージョンと 64 ビットバージョンの両方をインストールすることができます。ただし、MATLAB Release の両方のバージョンは同じ環境設定を共有するため、1 つの MATLAB Release の異なるバージョン間で切り替えを行うたびに、MEX コンパイラの設定などアーキテクチャ依存のすべての設定を行う必要があります。

TargetLink の 64 ビットバージョンの制限事項

A2L ファイルのインポート A2L ファイルは 64 ビットバージョンの TargetLink にインポートすることはできません。ただし、解決策は「Basics of Importing A2L Files」(📄『TargetLink Data Dictionary A2L Import and Export』)に記載されています。

Windows 7 の場合の制限事項

目的	Windows 7 と dSPACE ソフトウェアを組み合わせて使用する場合には、注意する必要がある事項が存在します。
MATLAB のサポート	MathWorks®社製ソフトウェアのシステム要件については、 http://www.mathworks.com/support/sysreq/current_release を参照してください。
ユーザの簡易切り替えのサポートなし	dSPACE ソフトウェアは、Windows のユーザの簡易切り替えをサポートしません。
PC をシャットダウンする前に dSPACE ソフトウェアを閉じる	Windows オペレーティングシステムのシャットダウン手順では、いくつかの必要なプロセスが、dSPACE ソフトウェアによって利用されている状態であっても中断されることがあります。データの損失を回避するには、PC のシャットダウンを実行する前に dSPACE ソフトウェアを手動で終了することをお勧めします。
ユーザアカウント制御	dSPACE ソフトウェアをインストールするときは、Windows の「ユーザアカウント制御 (UAC)」を無効にすることをお勧めします。UAC を無効にできない場合は、Windows の次のような動作に注意してください。UAC を有効にしていると、セットアッププログラムはユーザのアカウントではなく管理者アカウントで実行されます。したがって、管理者アカウントで必要なドライブ (特にネットワークドライブ) にアクセス可能である必要があります。
USB デバイス	光絶縁対応ケーブルを使用する dSPACE USB デバイスを初めて PC に接続すると、デバイスドライバソフトウェアが正常にインストールできなかったことを示すメッセージが表示されます。ただし、dSPACE デバイスはその後正常に動作します。

数字

- 64ビット dSPACE DVD
制限事項 231
- 64ビット dSPACE DVD に含まれる製品に
ついての制限事項 231

A

- ASM Base InCylinder Blockset
移行 44
- ASM Diesel Engine Blockset
移行 47
新機能 45
- ASM Diesel Exhaust Blockset
新機能 50
- ASM Drivetrain Basic Blockset
移行 52
新機能 52
- ASM Electric Components Blockset
新機能 54
- ASM Engine Gasoline Basic Blockset
移行 56
新機能 55
- ASM Engine Gasoline Blockset
移行 61
新機能 59
- ASM Gasoline InCylinder Blockset
移行 64
- ASM Parameterization Tool
移行 65
新機能 65
- ASM Pneumatics Blockset
移行 67
新機能 67
- ASM Traffic Blockset
移行 68
新機能 68
- ASM Turbocharger Blockset
新機能 69
- ASM Vehicle Dynamics Blockset
移行 72
新機能 70
- AutomationDesk
最新リリースでの使用 39
AutomationDesk の使用 39
- AUTOSAR
TargetLink 関連
移行 207

C

- CommonProgramDataFolder 12
- ControlDesk Next Generation
移行 95
新機能 86

D

- DocumentsFolder 12
- DS1006 GNU C/C++コンパイラ 124
- dSPACE FlexRay Configuration Package
新機能 109
- dSPACE HIL API .NET

- 新機能 99
- dSPACE Python Extensions
新機能 101
- dSPACE XIL API
移行 103
新機能 103
- DSPACE_CONFIG 16
- DSPACE_ROOT 16
- DVD の内容 17

E

- ECU Interface Manager
移行 106
新機能 105

F

- Firmware Manager
新機能 107

L

- LocalProgramDataFolder 12

M

- MATLAB
サポートされるリリース 226
- MATLAB のサポート対象外の機能
(R2014a) 78, 123
- MATLAB のサポート対象外の機能
(R2014b) 78, 124
- MicroAutoBox
新機能 123
- MicroLabBox
新機能 121
- Microtec PowerPC C/C++ Compiler 125
- Model Compare
移行 112
新機能 111
- ModelDesk
新機能 115
- MotionDesk
移行 118
新機能 117

R

- RCP and HIL Software
定義 17
- Real-Time Testing
新機能 119
- RTI Bypass Blockset
移行 130
新機能 129
- RTI CAN Blockset
新機能 133
- RTI CAN MultiMessage Blockset
移行 136
新機能 135
- RTI Electric Motor Control Blockset
新機能 139
- RTI Ethernet Blockset
新機能 141

- RTI FPGA Programming Blockset
移行 145
新機能 143
- RTI LIN MultiMessage Blockset
移行 147
新機能 147
- RTI USB Flight Recorder Blockset
新機能 149
- RTI/RTI-MP
新機能 121
- RTLib
新機能 121

S

- SCALEXIO Firmware
新機能 151
- Simulink
設定パラメータ
調整 213
- SystemDesk
新機能 154

T

- TargetLink
API コマンド
変更 216
- AUTOSAR 機能、新規
サポートされるリリース 175
- Code Generator オプション
後方互換性 204
変更されたデフォルト値 204
- 新しい API 関数 185
- 新しい Code Generator オプション 180
- 新しくサポートされる Simulink ブロック
161
移行
AUTOSAR 関連 207
コードの変更 192
新規バージョン 187
その他の注意点 212
廃止された制限事項 218
- コード効率性、改善 168
- コードの変更
移行 192
新機能 160
一般的な機能拡張 183
一般的な変更 183
- 新規バージョン
移行 187
- ターゲットプロセッササポート
新しいコンパイラバージョン 178
新しい評価用ボード 178
サポートされるターゲット 178
廃止されたコンパイラバージョン
178
廃止された評価用ボード 178
廃止された機能 217
- TargetLink Data Dictionary
API コマンド
新しいコマンド 172
変更 216
移行 188

- 既存のデータディクショナリのアップグレード 189
 新規バージョン 187
 廃止されたドキュメント 188
- 新機能 160
 新規バージョン
 移行 187
 廃止された機能 217
- V**
- VEOS
 新機能 223
- W**
- Windows 64 ビット版
 制限事項 231
 Windows 64 ビット版と dSPACE 64 ビットバージョンソフトウェアの制限事項 231
 Windows 7
 制限事項 235
 Windows 7 の場合の制限事項 235
- ア**
- 新しいハードウェア 15
- イ**
- 移行
 ASM Base InCylinder Blockset 44
 ASM Diesel Engine Blockset 47
 ASM Drivetrain Basic Blockset 52
 ASM Engine Gasoline Basic Blockset 56
 ASM Engine Gasoline Blockset 61
 ASM Gasoline InCylinder Blockset 64
 ASM Parameterization Tool 65
 ASM Pneumatics Blockset 67
 ASM Traffic Blockset 68
 ASM Vehicle Dynamics Blockset 72
 ControlDesk Next Generation 95
 dSPACE XIL API 103
 ECU Interface Manager 106
 Model Compare 112
 MotionDesk 118
 RTI 125
 RTI Bypass Blockset 130
 RTI CAN MultiMessage Blockset 136
 RTI FPGA Programming Blockset 145
 RTI LIN MultiMessage Blockset 147
 一般的な機能拡張および変更 15
- キ**
- 共通プログラムデータフォルダ 12
- サ**
- サポートしている MATLAB リリース 226
- シ**
- システム要件
 オペレーティングシステム 227
- 新機能
 ASM Diesel Engine Blockset 45
 ASM Diesel Exhaust Blockset 50
 ASM Drivetrain Basic Blockset 52
 ASM Electric Components Blockset 54
 ASM Engine Gasoline Basic Blockset 55
 ASM Engine Gasoline Blockset 59
 ASM Parameterization Tool 65
 ASM Pneumatics Blockset 67
 ASM Traffic Blockset 68
 ASM Turbocharger Blockset 69
 ASM Vehicle Dynamics Blockset 70
 ControlDesk Next Generation 86
 dSPACE FlexRay Configuration Package 109
 dSPACE HIL API .NET 99
 dSPACE Python Extensions 101
 dSPACE XIL API 103
 ECU Interface Manager 105
 Firmware Manager 107
 MicroAutoBox 123
 MicroLabBox 121
 Model Compare 111
 ModelDesk 115
 MotionDesk 117
 Real-Time Testing 119
 RTI Bypass Blockset 129
 RTI CAN Blockset 133
 RTI CAN MultiMessage Blockset 135
 RTI Electric Motor Control Blockset 139
 RTI Ethernet Blockset 141
 RTI FPGA Programming Blockset 143
 RTI LIN MultiMessage Blockset 147
 RTI USB Flight Recorder Blockset 149
 RTI/RTI-MP 121
 RTLib 121
 SCALEXIO Firmware 151
 SystemDesk 154
 VEOS 223
- ト**
- ドキュメントフォルダ 12
- ハ**
- バージョン履歴 21
- ホ**
- ホスト PC のソフトウェア
 オペレーティングシステム 227
- ヨ**
- 要件
 ホスト PC のソフトウェア
 オペレーティングシステム 227
- ロ**
- ローカルプログラムデータフォルダ 12
- オ**
- 主な機能 25
- カ**
- 環境変数
 コマンドプロンプトを介して設定 16
- セ**
- 制限事項
 TargetLink
 廃止された制限事項 218
 製品の概要 21
- チ**
- 調整
 Simulink 設定パラメータ 213