

Embedded Target for RH850 Virtual Platform

V1.00.00

RH850 モデルベース連携ツール

ユーザーズマニュアル

対象デバイス

RH850/F1KM

RH850/U2B

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサスエレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサスエレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行なうにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因またはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24 (豊洲フォレシア)

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。

このマニュアルの使い方

対象者	このマニュアルは、MATLAB/Simulink の機能を理解し、それを用いたアプリケーション・システムを開発するユーザを対象としています
目的	このマニュアルは、モデルベース連携ツール Embedded Target for RH850 Virtual Platform の持つ機能をユーザに理解していただき、システムのソフトウェア開発の参考用資料として役立つことを目的としています。
構成	このマニュアルは、大きく分けて次の内容で構成しています。 第1章 概説 第2章 インストレーション 第3章 機能別操作手順 第4章 注意事項 第5章 エラー・メッセージと対処方法
読み方	このマニュアルを読むにあたっては、電気、論理回路、マイクロコンピュータに関する一般知識が必要となります。
凡例	注： 本文中につけた注の説明 注意： 気をつけて読んでいただきたい内容 備考： 本文中の補足説明 数の表記： 10進数 ... XXXX 16進数 ... XXXXH、または0xFFFF
関連文書	

すべての商標および登録商標は、それぞれの所有者に帰属します。

Microsoft、Windows は Microsoft Corporation の商標、または登録商標です。
MathWorks、MATLAB、MATLAB Coder、Simulink、Simulink Coder、Embedded Coder、Stateflow は、The MathWorks, Inc. の商標、または登録商標です。

目 次

1 概説.....	5
1.1 特長.....	6
1.2 動作環境.....	6
1.3 ライセンスの種類と機能.....	8
2 インストレーション.....	9
2.1 インストール.....	11
2.2 アンインストール.....	13
3 機能別操作手順.....	15
3.1 概要.....	15
3.2 周辺ブロック.....	17
3.2.1 ADC 周辺ブロック.....	18
3.2.2 Port 周辺ブロック.....	20
3.2.3 RS-CANFD 周辺ブロック.....	22
3.2.4. RLIN3 周辺ブロック.....	26
3.2.5. TAUD 周辺ブロック.....	27
3.3 vHILS の実行	30
3.3.1 サンプル・モデル.....	30
3.3.2 コンフィギュレーション パラメーターの設定.....	31
3.3.3 vHILS 環境の生成	37
3.3.4 生成コードのコンパイル.....	43
3.3.5 vHILS の実行	45
3.4 時間計測.....	46
3.4.1 時間計測用 Simulink モデルの構造.....	46
3.4.2 時間計測のための入力ファイル.....	47
3.4.3 Atomic サブシステム変更方法	47
3.4.4 Graph Viewer.....	48
4 注意事項.....	53
4.1 仕様に関する注意事項.....	53
4.2 Simulink モデルに関する注意事項.....	55
4.2.1 パス名やブロック名に使用できる文字列.....	55
4.2.2 複素数データを扱うモデル	55
4.2.3 周辺ブロックへの Constant ブロック入力	55
4.3 実行環境構築時やシミュレーション中の注意事項	55
4.3.1 コード生成フォルダのパスの長さ	55
4.3.2 電源管理	55
4.3.3 スクリプト・ファイル名の長さ	55
4.3.4 インストール・ドライブと作業ドライブ	56
4.3.5 デフォルト設定での CS+プロジェクトのビルド	56
5 エラー・メッセージと対処方法.....	57
5.1 概要.....	57
5.2 コンフィギュレーション パラメーターのエラー	58
5.3 vHILS 中のエラー	60
5.4 CS+起動時、CS+でのビルド中のエラー	61
5.5 vHILS 環境構築時のエラー	62
5.6 MATLAB と VLAB 間の通信中のエラー	62
索引.....	63
改訂記録.....	64

1 概説

本章では Embedded Target for RH850 Virtual Platform (以下、ET-VPF) の機能の概要を説明します。

モデルベース開発で利用される MILS(Model In the Loop Simulation) でアルゴリズムを検討する段階では、ターゲットデバイスの周辺機能を含めた性能を測定する環境がありません。実際のデバイスで、周辺機能の処理時間を見積るには、モデルから生成されたコードに周辺機能を制御するためのコード(以下、周辺コード)を結合させる必要があり、周辺機能の詳細を知らないユーザにとっては大変な手間です。

ET-VPF は、Simulink モデルからターゲットに実装するユーザ・アルゴリズム部分のコード生成に加え周辺コードを生成した後、パートナー製仮想環境 (RH850 用シミュレータ) (以下、VPF) 上で実行し、Simulink のプラント・モデルとの連携シミュレーションを行います。これにより、開発の初期段階で、MILS と vHILS(virtual Hardware In the Loop Simulation) の比較検証(Back to Back テスト)および性能検証が可能になります。

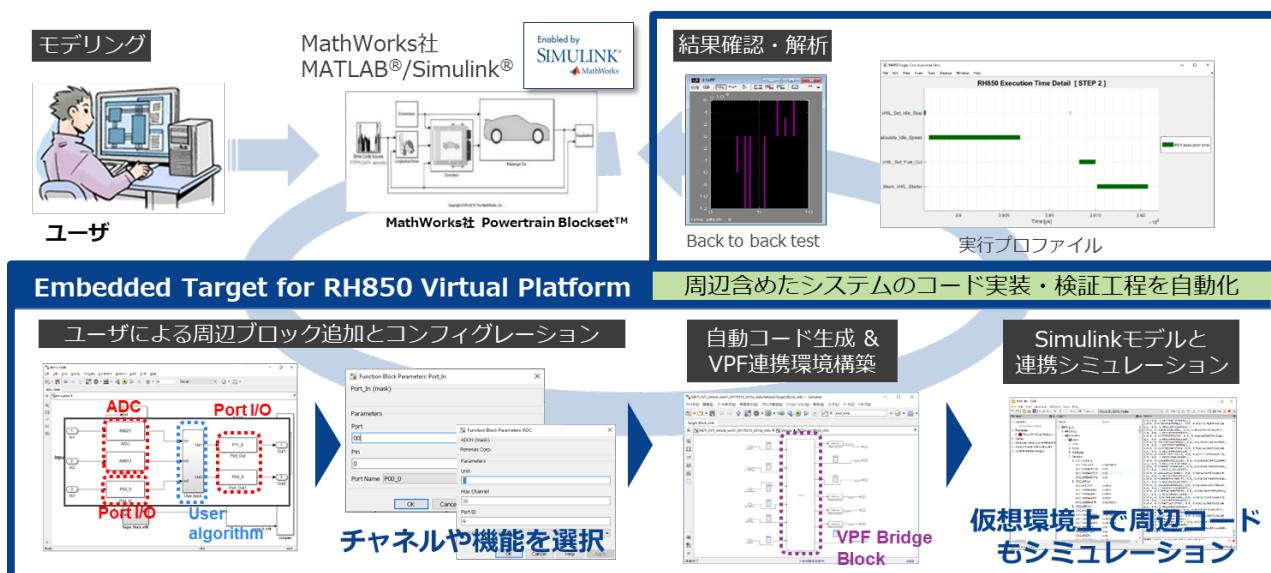


図 1-1 Embedded Target for RH850 Virtual Platform の概要

1.1 特長

ET-VPF の特長を以下に示します。：

- vHILS検証環境の自動生成
 - MATLAB/Simulink、VPFの連携シミュレーションでは、Simulinkモデルから生成されたロード・モジュールをターゲットデバイス上で実行できます。
 - サポートしているデバイスは以下の通りです。

表 1.1 サポートデバイス

シリーズ	デバイス
RH850/F1x	RH850/F1KM-S1, RH850/F1KM-S4 (以降 RH850/F1KM)
RH850/U2x	RH850/U2B6, RH850/U2B10 (以降 RH850/U2B)

- SimulinkモデルのSubsystemブロックにおけるコア固有の実行状態のグラフィカル表示
 - シミュレーション中の各サブシステムの実行時間を表示します。
 - 最大実行時間(ワーストケース)の制御サイクルの処理余裕度を確認できます。

1.2 動作環境

ET-VPF の動作環境を以下に示します。

- ハードウェア環境
 - オペレーティング・システム : Microsoft Windows10 (64bit)
 - プロセッサ : 1GHz 以上 (ハイパースレッディング、マルチコア CPU に対応)
 - メイン・メモリ : 4G バイト、またはそれ以上を推奨

· ソフトウェア環境

- MATLAB、Simulink製品 (The MathWorks, Inc製)

MATLAB	V9. 8 (R2020a), V9. 3 (R2017b)
Simulink	V10. 3 (R2020a), V9. 3 (R2017b)
Stateflow	V10. 4 (R2020a), V9. 0 (R2017b)
MATLAB Coder	V5. 2 (R2020a), V9. 0 (R2017b)
Simulink Coder	V9. 5 (R2020a), V8. 13 (R2017b)
Embedded Coder	V7. 6 (R2020a), V6. 13 (R2017b)
Vehicle Network Toolbox	V5. 0 (R2020a), V3. 4 (R2017b)

補足 : MATLABバージョンR2020aはRH850/U2B使用時、R2017bはRH850/F1KM使用時にのみ利用できます。

· MEXファイル用コンパイラ

- MEXファイルは、MATLABからのCライブラリを起動するインターフェイスです。
 - MEXファイルのコンパイルに使用するMEXコンパイラは、MathWorks社が公開している動作環境のページで確認できます。
- ET-VPFでは、MEXファイル用コンパイラとして、以下のコンパイラにて動作確認済みです。
 - Microsoft Visual C++ 2015 compiler (Microsoft Corporation製)

参考 : System Requirements & Platform Availability

https://jp.mathworks.com/support/sysreq/previous_releases.html

- MinGW V7.3.0
- VPF (Australian Semiconductor Technology Company 製)
 - ・ VLAB
 - V2.7.3 (win-vc140-x64) (RH850/F1KM 使用時)
 - V2.8.3 (win-vc140-x64) (RH850/U2B 使用時)
 - ・ VLAB ツール・ボックス
 - Toolbox RH850 Virtual Platform V3.2.1 (RH850/F1KM 使用時)
 - Toolbox RH850 G4 Virtual Platform V1.22.1 (RH850/U2B 使用時)
 - Toolbox CAN 2.5.0 (RS-CANFD ブロック使用時)
- スマート・コンフィグレータ (ルネサス エレクトロニクス製)
 - Smart Configurator for RH850 V1.5.0 (RH850/F1KM 使用時)
 - V1.8.0 (RH850/U2B 使用時)
- ビルド・ツール (ロード・モジュールを生成する環境) (ルネサス エレクトロニクス製)
 - CS+ V8.07.00 (RH850/F1KM 使用時)
 - V8.09.00 (RH850/U2B 使用時)
- コンパイラ (ルネサス エレクトロニクス製)
 - CC-RH V2.05.00
- Cygwin (www.cygwin.com にて公開)
 - Cygwin V2.11.2

補足 1. MATLAB および Simulink 製品の場合、使用されている MATLAB および Simulink のバージョンに対するオプション製品を使用して環境が構築されます。
2. MATLAB および ET-VPF をインストールする際は、UAC(User Account Control)対象となるフォルダ以外にインストールすることを推奨します。使用している MATLAB のバージョンによっては、インストール・パスが”<システム・ドライブ>:\Program Files” や”<システム・ドライブ>:\Program Files (x86)”などのUAC 対象のフォルダである場合、MEX ビルドできない、または MATLAB パスを保存できないなどの問題が発生する可能性があり、ET-VPF を利用することができません。
3. ET-VPF で使用している機能を変更しない場合は、上位のソフトウェア・バージョンを使用できますが、動作は保証しません。
4. ET-VPF、CS+、Cygwin、Smart Configurator、および VLAB のインストール・パスには、一部の特殊文字を含めることはできません（詳細は、「表 3.7 使用可能な特殊文字」を参照してください）。

1.3 ライセンスの種類と機能

本節ではライセンスについて説明します。

ET-VPF は、組込みモデルのアルゴリズムを検証するための様々な機能を提供しています。各機能の利用にはライセンスが必要です。各ライセンスを所有している場合に使用可能な操作を下表に示します。

表 1.2 ライセンス・リスト

ライセンス名		Embedded Target for RH850/F1x Virtual Platform	Embedded Target for RH850/U2x Virtual Platform
サポートデバイス	RH850/F1KM-S1	○	-
	RH850/F1KM-S4	○	-
	RH850/U2B6	-	○
	RH850/U2B10	-	○
時間計測機能		○	○
ビルド・ツール	GHS Compiler	-	-
	Renesas Compiler	○	○

* ○ : 実行可能
- : 実行不可

2 インストレーション

ET-VPFには以下のプログラム、サンプルが含まれています。

(1) ET-VPF パッケージ・フォルダの構成 (RH850/F1KM版)

<ET-VPFインストール・フォルダ>/ETVPF_package	… 周辺機能の実行に使用するソースコード、バイナリファイル
ETVPF_include/make	… Makefileのテンプレート
ETVPF_include/Renesas	… ルネサスコンパイラで使用する各デバイスのソースコード
注：“startup” フォルダ内のブートコードファイルはサンプルです。変更してお使いいただけますが、品質については保証しません。	
ETVPF_include/XML_input	… 各デバイス用XML入力ファイル
Source code files	… ET-VPFのプログラムの実行に使用するソースコードファイル (*.p, *.tcl, makeファイルを含む)

(2) ET-VPF パッケージ・フォルダの構成 (RH850/U2B版)

<ET-VPFインストール・フォルダ>¥information>¥U2B¥ETVPF_package'	周辺機能の実行に使用するソースコード、バイナリファイル
ETVPF_include¥make¥	... Makefileのテンプレート
ETVPF_include¥Renesas¥	ルネサスコンピラで使用する各デバイスのソースコード ...
	注：“startup” フォルダ内のポートコードファイルはサンプルです。変更してお使いいただけますが、品質については保証しません。
ETVPF_include¥XML_input¥	... 各デバイス用XML入力ファイル
Source code files	ET-VPFのプログラムの実行に使用するソースコードファイル (*.p, *.t c, makeファイルを含む)

補足

1. ET-VPF のインストール時に<ET-VPF インストール・フォルダ>を選択できますが、インストールが制限されていないことを確認する必要があります。また、<ET-VPF インストール・フォルダ>は、ロード・モジュールを仮想環境にロードする際に発生するエラーを回避するために、ユーザのワークスペースと同じドライブにある必要があります。
2. <ET-VPF インストール・フォルダ>に特殊文字(「表 3.7 使用可能な特殊文字」参照)が含まれる場合はエラーになるため、特殊文字を含まないパスにインストールしてください。

2.1 インストール

以下に ET-VPF のインストール手順を示します。

- (1) RH850/F1KM 版は、ET-VPF パッケージは圧縮ファイルとして提供されます。インストールするには任意のフォルダに展開してください。
- RH850/U2B 版はインストーラ ETVPF_U2x_V1.00.00_Setup.exe を起動し、インストールしてください。

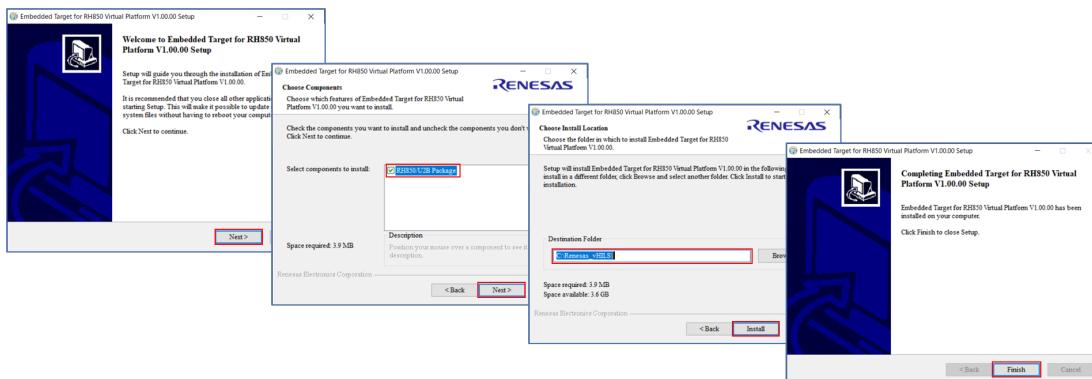


図 2-1 インストール・イメージ

- (2) ET-VPF パッケージのインストールが完了したら、ET-VPF パッケージへのパスを設定します。MATLAB を起動し、[パス設定] メニューで起動するパス設定ダイアログで、以下の 2 つのパスを追加してください。以下は設定例です。
 - ET-VPF include: … <インストール・パス>\ETVPF_package\ETVPF_include
 - ET-VPF S-Function: … <インストール・パス>\ETVPF_package\ETVPF_include\ETVPF_S_function_block

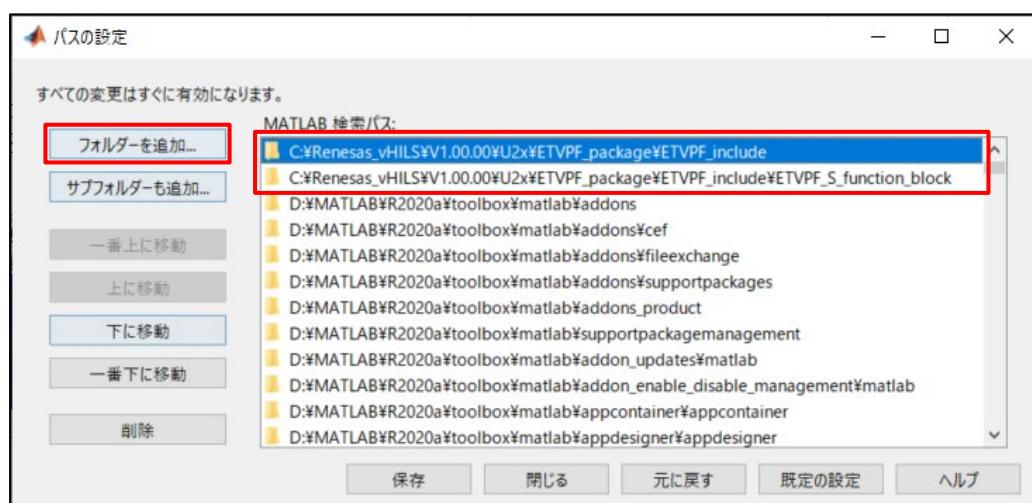


図 2-2 ET-VPF パスの追加例

(3) MATLAB オートメーション サーバー の登録

オートメーション サーバーとして使用する MATLAB バージョンを指定するため、MATLAB のコマンド・ウィンドウから以下のコマンドを実行します。

“>>” はコマンド・プロンプトを、“[Enter]” はエンター・キーの入力を表しています。

例：

```
>> regmatlabserver [Enter]
```

補足

1. このコマンドを実行する際は、管理者権限で MATLAB を起動してください。
2. 使用する MATLAB バージョンを変更する場合、このコマンドを再実行してください。

(4) ライセンスの登録

CS+に同梱されているライセンス・マネージャを使用して、ET-VPF のライセンスを追加してください([ライセンス・マネージャ V2.05.00 ユーザーズ・マニュアル](#)参照)。

2.2 アンインストール

以下に、ET-VPF のアンインストール手順を示します。

- (1) MATLAB を起動し、[パス設定] メニューで起動するパス設定ダイアログで、「2.1 インストール」で追加した 2 つのパスを削除してください。

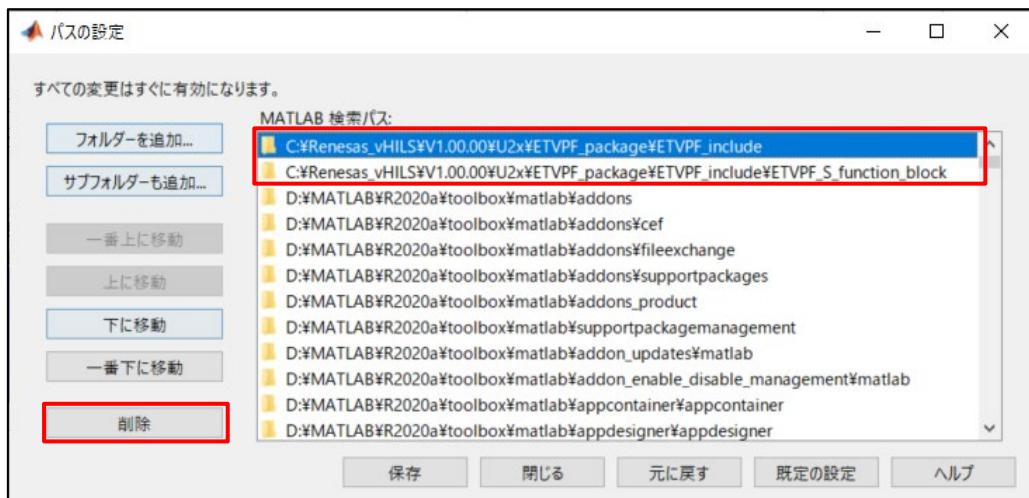


図 2-3 ET-VPF パスの削除

- (2) RH850/F1KM 版の場合、フォルダから ET-VPF パッケージを削除してください。

RH850/U2B 版の場合、アンインストーラ Uninst_ETVPF_U2x_V1.00.00.exe を起動し、以降の図に従い、アンインストールを行ってください。

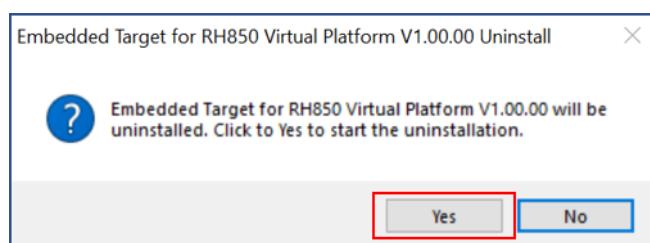


図 2-4 アンインストール確認メッセージ

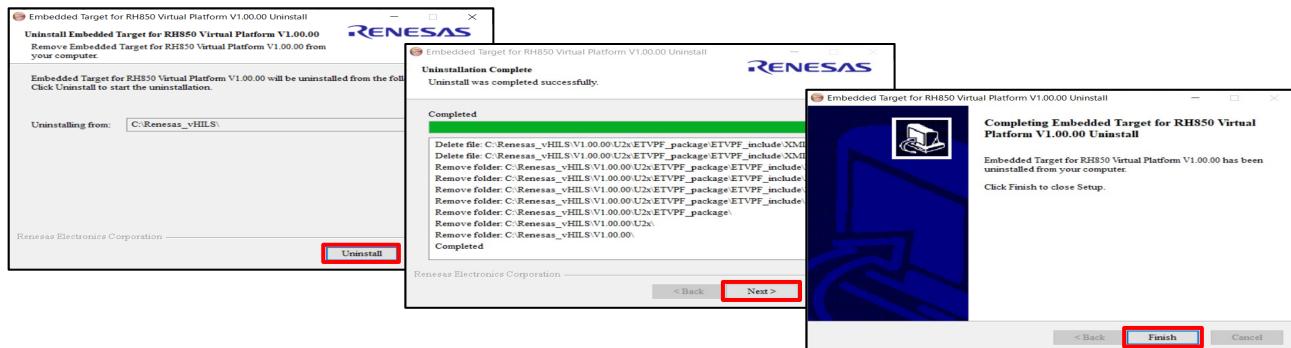


図 2-5 アンインストール・イメージ

補足：上記手順(1)より先に(2)を実行すると、次回 MATLAB 起動時にワーニングが表示されます。

(3) CS+に同梱されているライセンス・マネージャを使用して、ET-VPF のライセンスを削除してください。

(4) ファイル Simulink ライブラリ ブラウザーから ET-VPF の周辺ブロックを削除します。

MATLAB のコマンド・ウィンドウから以下のコマンドを実行します。

“>>” はコマンド・プロンプトを、“[Enter]” はエンター・キーの入力を表しています。

例：

```
>> refresh(LibraryBrowser.LibraryBrowser2) [Enter]
```

補足：ET-VPF をアンインストール後、MATLAB を再起動せずに Simulink ライブラリ ブラウザー中の ET-VPF の周辺ブロックを参照しようとすると、ワーニングが表示されます。MATLAB を再起動すれば、上記手順は不要です。

3 機能別操作手順

本章では、ET-VPF の機能別に、その概要と操作手順を説明します。

3.1 概要

ET-VPF は、Embedded Coder を利用しコード生成した後、vHILS 環境を構築し、アルゴリズムを検証するための機能を提供します。

ET-VPF の主な手順は以下の通りです。

- コンフィギュレーションパラメーターの設定
- 既存のファイル、オブジェクトの削除
- ソースコードファイルの生成
 - マイコン周辺機能を制御するコード(以下、周辺コード)の生成
 - ターゲットデバイス用ソース・ファイルの生成
 - C ソース・ファイルの生成
 - Makefile とコンフィギュレーション・ファイルの生成
 - アプリケーション makefile の生成
 - VPF 実行用ファイルの生成
 - Python ファイルの生成
 - Define.h、OSTM_define.h の生成
- Cygwin によるソースコードのビルド
- VPF の起動とターゲット python スクリプトの実行
- 時間計測結果の表示

補足：周辺ブロックによるデータの送受信はサンプリング時間毎に実行されます。

周辺コードを生成するための S-Function ブロック(以下、周辺ブロック)には、ET-VPF が提供する etvpf_sfunction.slx モデルに含まれる Port、ADC、RS-CANFD、RLIN3n、TAUD があります。

各周辺機能の設定は、周辺ブロックでの簡易的な設定の他、スマート・コンフィギュレータ(以下、SC)にて詳細な設定も可能です。SC での設定方法は、「3.3.3.1 Smart Configurator による周辺コード生成」を参照してください。

下表に、測定対象サブシステムと同じ階層に配置できるブロック（以下、配置可能なブロック）を示します。

表 3.1 配置可能なブロック

番号	配置可能なブロック	イメージ図
1	Subsystem	
2	Import (*1)	
3	Outport	
4	Mux	
5	Demux	
6	Data Type Conversion	
7	ADC 周辺ブロック	
8	PORT 周辺ブロック	
9	TAUD 周辺ブロック	
10	RLIN3 周辺ブロック	
11	RS-CANFD 周辺ブロック	
12	Chart	

(*1) このブロックは、端子番号が重複しない場合に割り振ることができます。

3.2 周辺ブロック

本節では、各周辺ブロックについて、作成方法、設定方法、C コードの生成方法について説明します。

- 補足**
1. 周辺ブロックはデバイスシリーズごとに異なります。したがって、ユーザは使用するデバイスシリーズで正しい周辺ブロックを選択する必要があります。また、モデル上の周辺ブロックの設定は SC の設定と同じでなければなりません（「3.3.3.1 Smart Configurator による周辺コード生成」参照）。各デバイス シリーズの周辺ブロックは“etvpf_sfunction.slx” モデルにて提供されます。
 2. 周辺ブロックの設定では、以下の機能、設定のみを使用してください。
 - a) ADC 周辺ブロック：ユニット 0(ADCA0)、スキャングループ 1、マルチサイクルスキャンモードのみを使用できます。
 - b) Port 周辺ブロック：通常のポートのみを使用できます（例：PORT0、PORT1 など SC で定義されているもの）
 - c) RS-CANFD 周辺ブロック：
 - CANFD ユニット 0(RCFD0)、データフレーム/リモートフレーム、標準 ID/拡張 ID、受信フィルタ機能、CAN ポートのみを使用できます。また、Vehicle Network Toolbox の利用が可能です。
 - d) RLIN3 周辺ブロック：
 - U2B の場合：RLIN3n(n=0, 1, 2, 3, 4, 5, 23、各 1 チャネル) の 7 ユニットをサポートします。
 - e) TAUD 周辺ブロック：
 - F1KM の場合：PWM 出力と三角波 PWM 出力機能をサポートします。マスタ・チャネル、パルス周期、デューティ比の設定のみを使用できます。スレーブ・チャネルやクロックについては SC にて設定してください。
 - U2B の場合：PWM 出力機能のみサポートします。マスタ・チャネル、パルス周期、デューティ比の設定のみを使用できます。スレーブ・チャネルやクロックについては SC にて設定してください。TAUD0～TAUD3(各 16 チャネル) の 4 ユニットをサポートします。
 3. 周辺ブロックはシリーズ内の最大仕様のデバイス用に準備されています。それより小さい仕様のデバイスを使用する場合は、そのデバイスに実装されている機能、ポート名を選択してください。
 4. UART 送信関数が 8 バイトのデータ（一度に 1 バイトずつ）の転送を完了するのを待たなくて済むよう、ユーザーはバッファ設定を変更できます。

ET-VPF の周辺ブロックは、MATLAB Simulink ライブラリ ブラウザの ETVVF パッケージに含まれています。

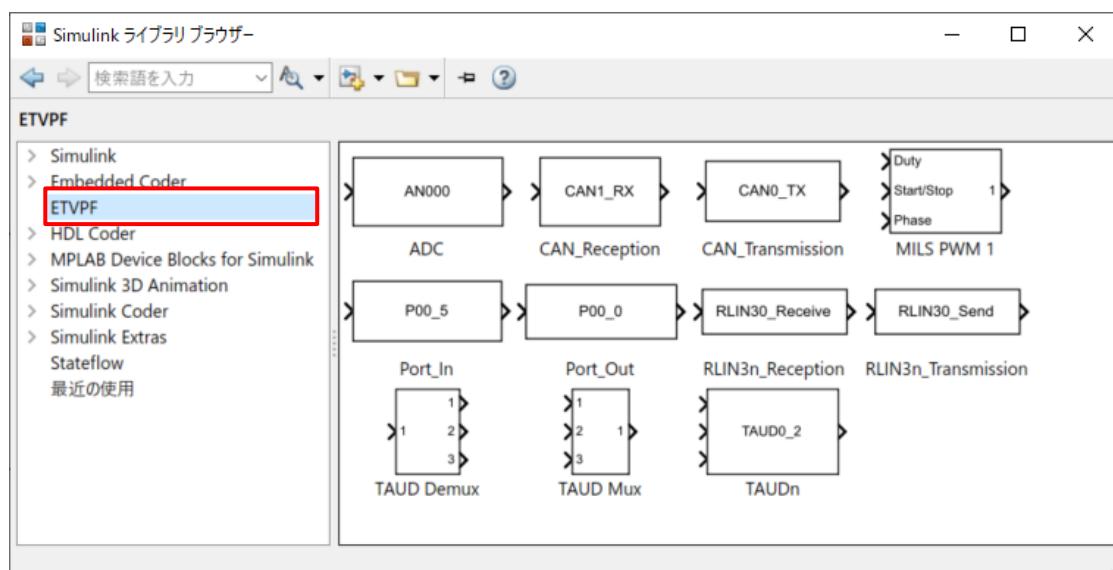


図 3-1 ET-VPF の周辺ブロック

3.2.1 ADC 周辺ブロック

ADC 周辺ブロックの主な機能は以下の通りです。

- プラント・モデルから取得した値を VPF の ADC ポートへ入力する
- ADC による変換結果をユーザ・アルゴリズムへ渡す

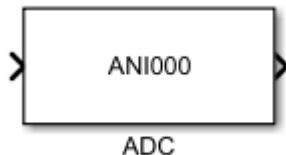


図 3-2 ADC 周辺ブロック

次に、ADC 周辺ブロックのブロック パラメーターについて説明します。

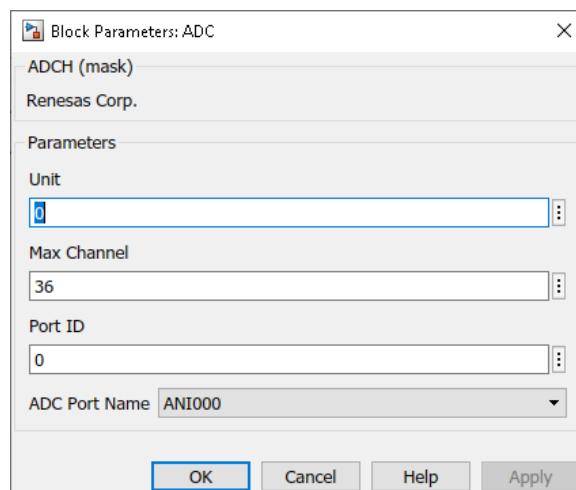


図 3-3 ADC 周辺ブロックのブロック パラメーター

ADC 周辺ブロックには、”ADC Port Name”、”Unit”、”Max Channel”、”Port ID” パラメータがありますが、”ADC Port Name”のみ変更します。”ADC Port Name”を変更すると、それに応じて”Unit”、”Max Channel”、”Port ID”は自動的に変更され、ADC 周辺ブロック名にも反映されます。

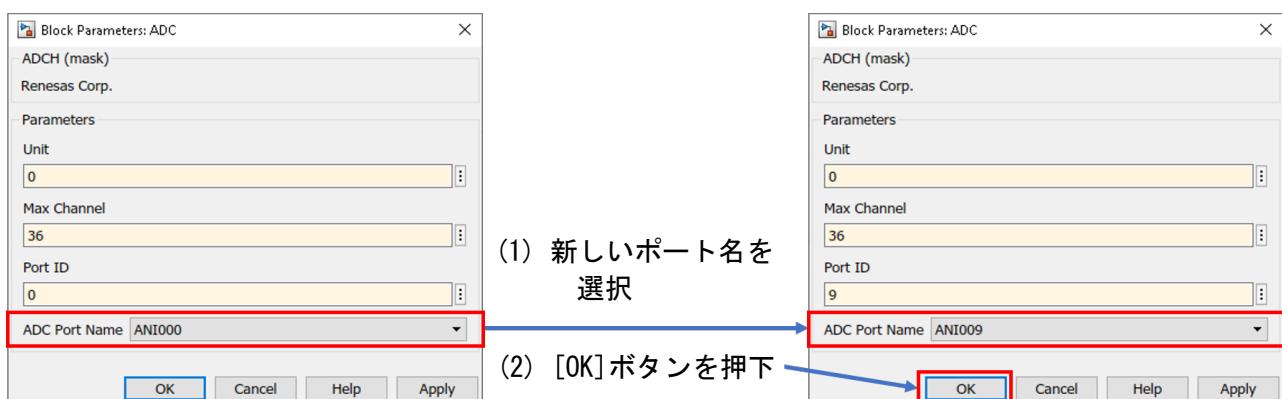
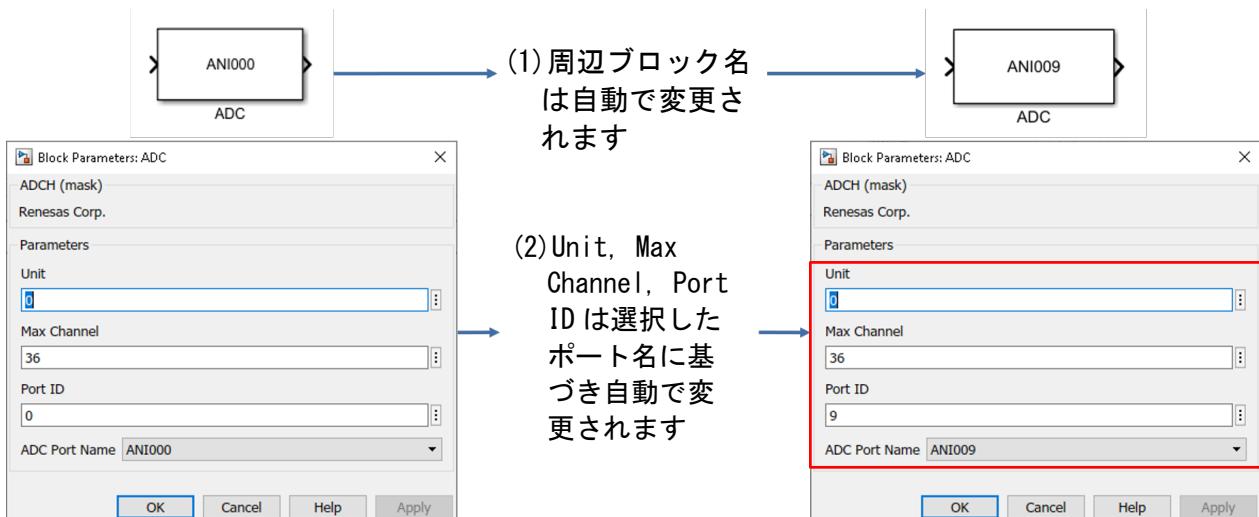


図 3-4 ADC 周辺ブロックでのポート名の変更



3.2.2 Port 周辺ブロック

Port 周辺ブロックの主な機能は以下の通りです。

- Port_In ブロック
 - プラント・モデルの値を VPF のポートへ入力する
 - 入力した値をユーザ・アルゴリズムへ渡す
- Port_Out ブロック
 - ユーザ・アルゴリズムの値を VPF のポートから出力する.
 - ポートからの出力値をプラント・モデルへ渡す



図 3-6 Port 周辺ブロック

次に、Port 周辺ブロック (Port_In および Port_Out) のブロック パラメーターについて説明します。

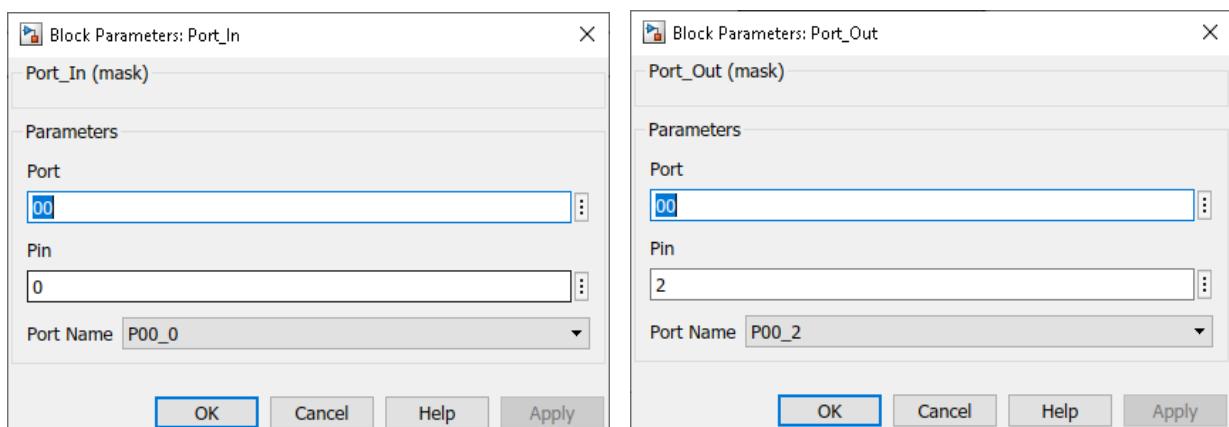


図 3-7 Port 周辺ブロックのブロック パラメーター

Port 周辺ブロックには、“Port Name”、“Port”、“Pin” パラメータがありますが、“Port Name”のみ変更します。”Port Name” を変更すると、それに応じて”Port”、“Pin” は自動的に変更され、Port 周辺ブロック名にも反映されます。

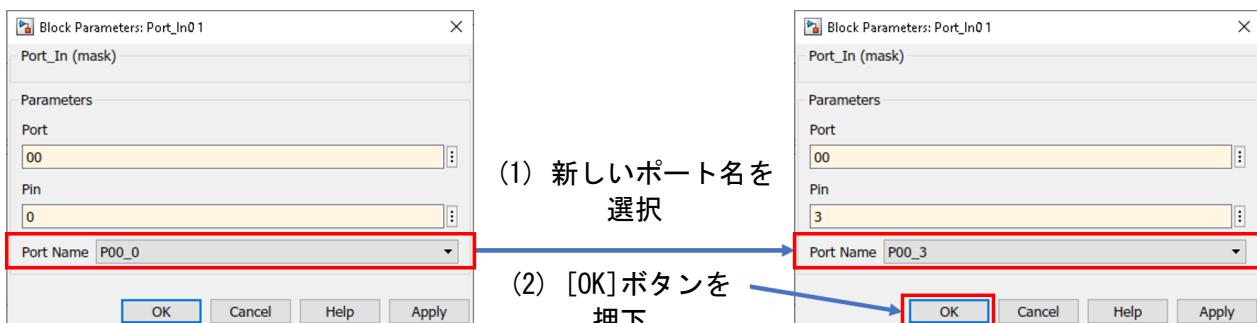


図 3-8 Port 周辺ブロックでのポート名の変更

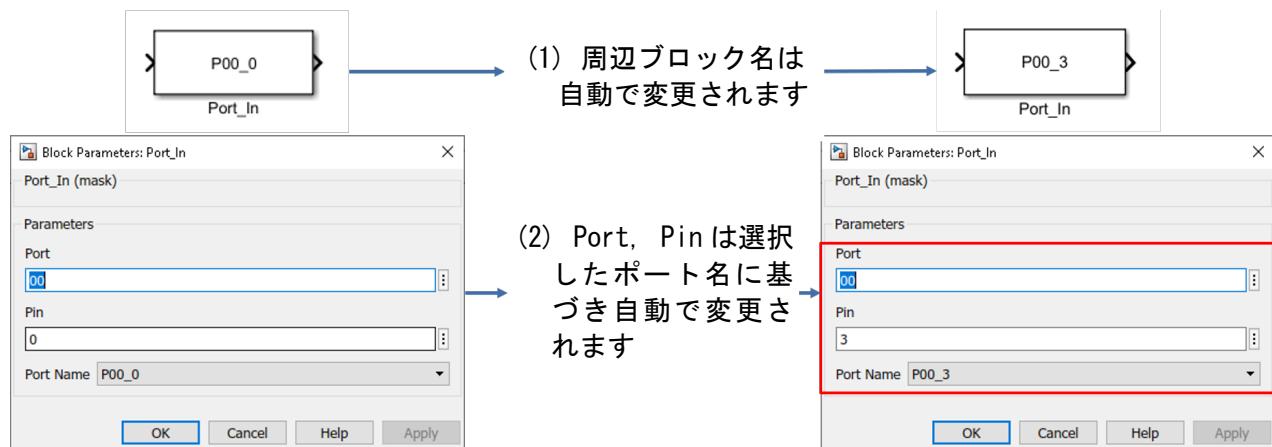


図 3-9 Port 周辺ブロックでのポート名変更結果

3.2.3 RS-CANFD 周辺ブロック

RS-CANFD 周辺ブロックの主な機能は以下の通りです。

- CAN_Transmission ブロック
 - ユーザ・アルゴリズムからデータ配列(サイズは任意)を取得し、VPF の RS-CANFD ポートを介して CAN メッセージをプラント・モデルに送信する
- CAN_Reception ブロック
 - プラント・モデルから VPF の RS-CANFD ポートを介して CAN メッセージを受信し、ユーザ・アルゴリズムへデータ配列を渡す

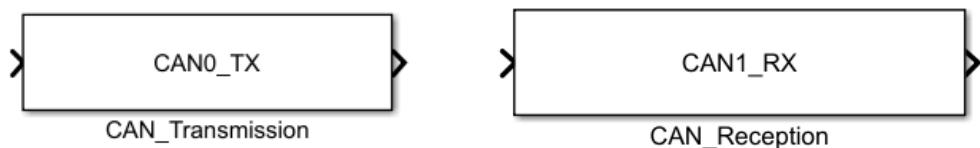


図 3-10 RS-CANFD 周辺ブロック

次に、RS-CANFD 周辺ブロック (CAN_Transmission および CAN_Reception) のブロック パラメーターについて説明します。

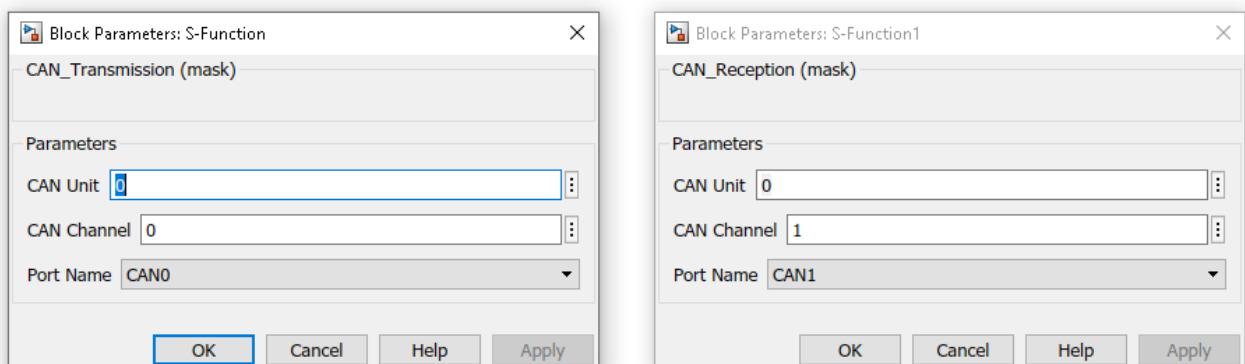


図 3-11 RS-CANFD 周辺ブロックのブロック パラメーター

RS-CANFD 周辺ブロックには、“CAN Unit”、“CAN Channel”、“Port Name” パラメータがありますが、“Port Name”のみ変更します。“Port Name”を変更すると、それに応じて“CAN Unit”、“CAN Channel”は自動的に変更され、CAN 周辺ブロック名にも反映されます。

CAN メッセージ ID は、RS-CANFD 周辺ブロックに接続される CAN Pack、CAN Unpack ブロックにて設定されたブロック パラメーターから自動的に取得されます。

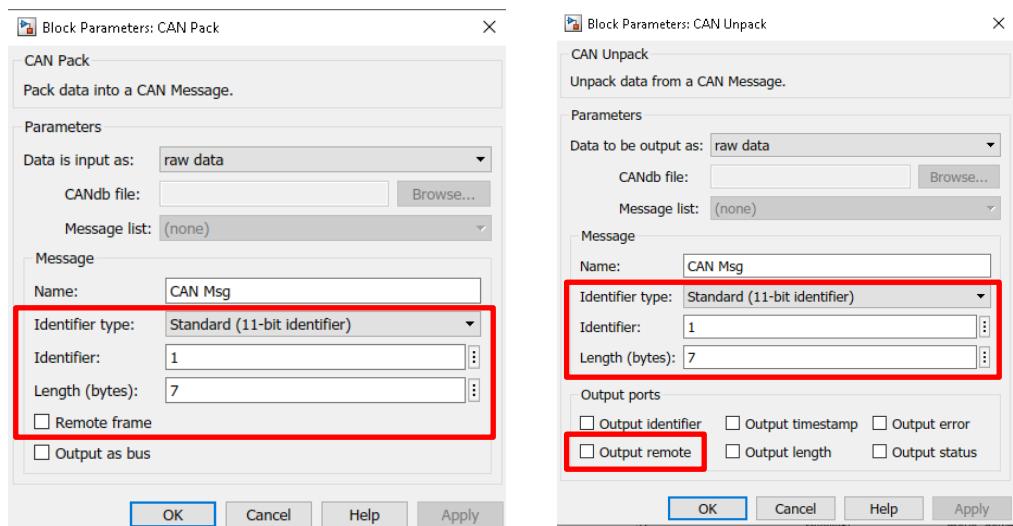


図 3-12 CAN Pack、CAN Unpack ブロックから取得される情報

- MATLAB と VPF のモデル構成を以下に示します。CAN Reception ブロックは CAN Unpack と接続し、CAN transmission ブロックは CAN Pack ブロックと接続する必要があります。また、CAN Pack/Unpack ブロックは、コード生成対象サブシステム(下図の controller block)の上位層に配置する必要があります。

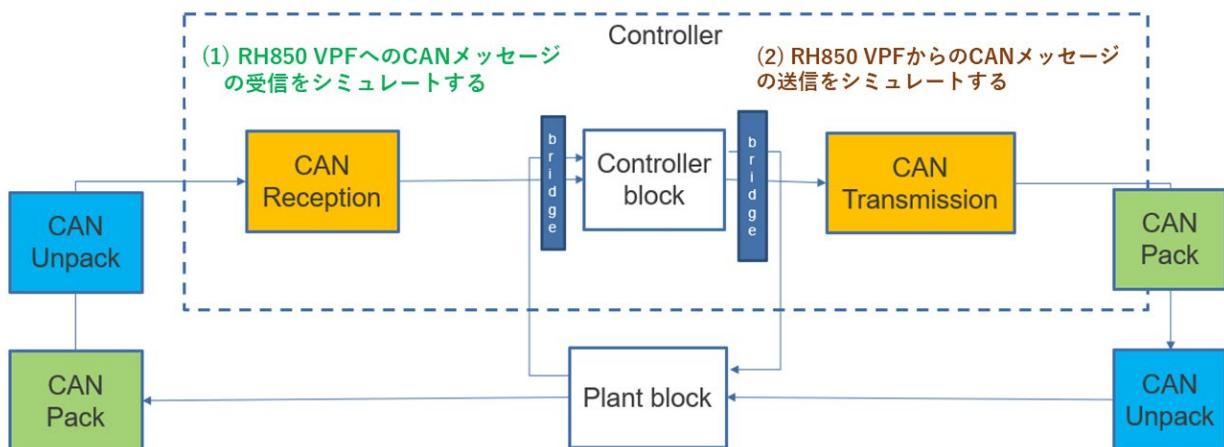


図 3-13 RS-CANFD 周辺ブロックを使ったモデル例(1)

- ET-VPF では MathWorks 製 Vehicle Network Toolbox を使ってモデルを構成することも可能です。

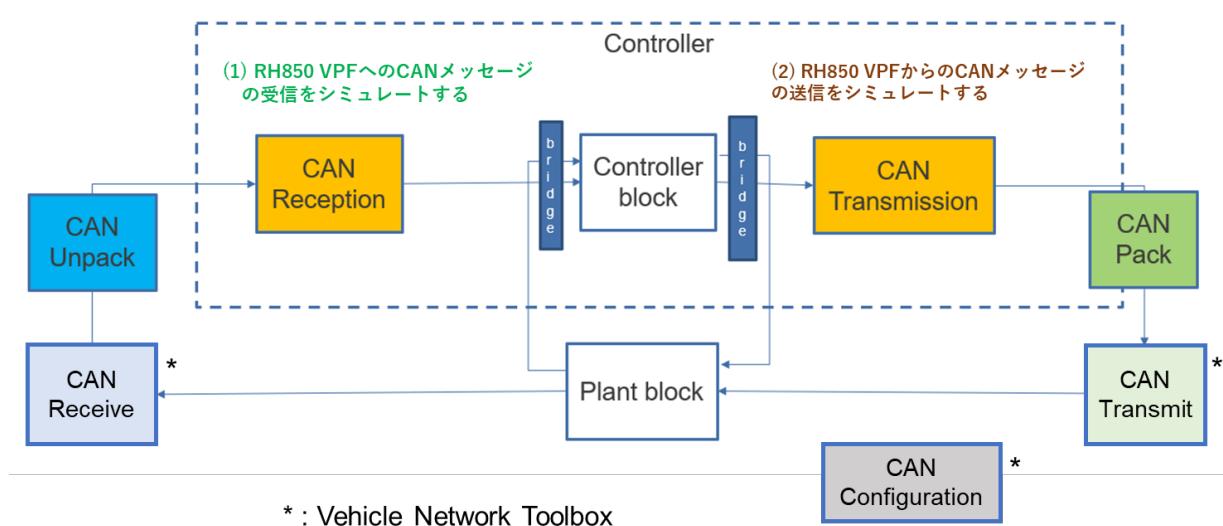


図 3-14 RS-CANFD 周辺ブロックを使ったモデル例(2)

- CAN 受信メッセージのタイムアウト

ET-VPF は、受信中に CAN メッセージを待機するループ処理を用意しています。タイムアウト時間が不十分な場合、CAN メッセージが失われる可能性があるため、インストール・フォルダ中の以下のファイルで適切なループ回数を定義してください。

[F1KM-S1 の場合]

ETVPF_package/ETVPF_include/Renesas/Common_files_F1KM_S1/include/Define_Macro_loop.h

[F1KM-S4 の場合]

ETVPF_package/ETVPF_include/Renesas/Common_files_F1KM_S4/include/Define_Macro_loop.h

[U2B の場合]

ETVPF_package/ETVPF_include/Renesas/Common_files_U2B10/include/Define_Macro_loop.h

ETVPF_package/ETVPF_include/Renesas/Common_files_U2B6/include/Define_Macro_loop.h

ループ回数は以下の式で求めることができます。

$$\text{Loop times} = \frac{\text{Time out} * \text{CPUCLK}}{\text{Number of loop instructions}}$$

Time out: 受信メッセージの待ち時間(sec)

CPUCLK: CPUCLK (Hz)

Number of loop instructions: 1 ループ内の命令数 (通常 15 命令程度)

表 3.2 RS-CANFD 周辺ブロックが使用するポート名

対象デバイス	RS-CANFD 周辺ブロック	ポート名
RH850/F1KM-S1 RH850/F1KM-S4	CANO RX CAN0 TX CAN1 RX CAN1 TX CAN2 RX CAN2 TX CAN3 RX CAN3 TX CAN4 RX CAN4 TX CAN5 RX CAN5 TX	P0_1 P0_0 P0_2 P0_3 P0_5 P0_4 P0_7 P0_8 P0_9 P0_10 P0_13 P0_14
RH850/U2B	CANO RX CAN0 TX CAN1 RX CAN1 TX CAN2 RX CAN2 TX CAN3 RX CAN3 TX CAN4 RX CAN4 TX CAN5 RX CAN5 TX CAN6 RX CAN6 TX CAN7 RX CAN7 TX	P02_2 P02_0 P00_1 P00_0 P02_4 P02_1 P00_3 P00_2 P00_5 P00_4 P14_0 P14_2 P00_10 P00_11 P20_4 P20_5

3.2.4. RLIN3 周辺ブロック

RLIN3 周辺ブロックの主な機能は以下の通りです。

- RLIN3n Transmission ブロック
 - ユーザ・アルゴリズムから取得した double 型データを、VPF の RLIN3 ポートを介して 8 ビット単位で送信し、64 ビット分をまとめて double 型データとしてプラント・モデルへ渡す
- RLIN3n Reception ブロック
 - プラント・モデルから double 型データを VPF の RLIN3 ポートを介して 8 ビット単位で受信し、64 ビット分をまとめて double 型データとしてユーザ・アルゴリズムへ渡す

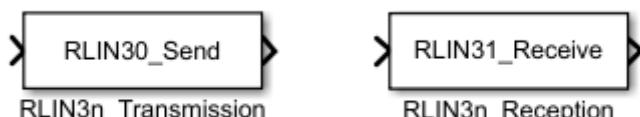


図 3-15 RLIN3 周辺ブロック

次に、RLIN3 周辺ブロック (RLIN3n_Send および RLIN3n_Recv) のブロック パラメーターについて説明します。

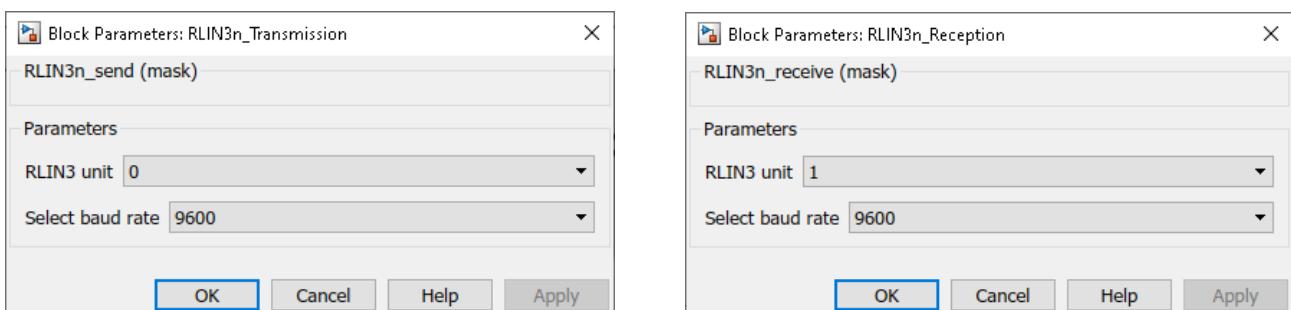


図 3-16 RLIN3 周辺ブロックのブロック パラメーター

RLIN3 周辺ブロックには、”RLIN3 unit”、”Select baud rate” パラメータがあります。”RLIN3 unit” を変更すると、RLIN3 周辺ブロック名にも反映されます。

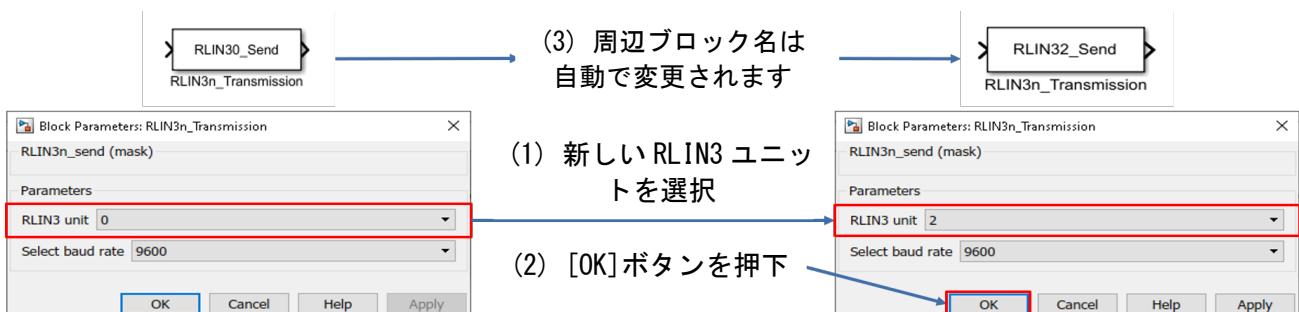


図 3-17 RLIN3 周辺ブロックでのユニット番号変更結果

3.2.5. TAUD 周辺ブロック

TAUD 周辺ブロックの主な機能は以下の通りです。

- PWM 信号の出力を制御するために、ユーザ・アルゴリズムから以下の 3 つの入力を受け取ります。
 - "Duty" (ダブル型配列) : デューティ比
 - "Start/Stop" (boolean 型) : 開始トリガ(1)、停止トリガ (0)
 - "Pulse cycle" (double 型) : パルス周期
 - PWM 信号としてデータ配列 ("Master channel" と "Slave channel") を出力します。スレーブ・チャネルは SC にてユーザが選択します。
 - PWM 出力
 - PWM 信号を生成し、VPF の "Slave channel" を介してプラント・モデルに boolean 型データを送信
 - 三角波 PWM 出力 (F1KM のみサポート)
 - 三角形 PWM 信号を生成し、VPF の "Slave channel" を介してプラント・モデルに boolean 型データを送信
- 補足：U2B の場合、TAUD モジュールは OSTM との共通クロック (CLKC_HSB) (デフォルト 20MHz) を使用します。異なるクロック値を使用する場合は、正しい結果を得るためにパルス周期を変換する必要があります。変換方法は以下の通りです。
- 設定パルス周期 (ms) = <希望パルス周期>*20/<入力クロックの周波数>
- パルス周期を 200ms にする際の変換例
 - 入力クロックが 20MHz の場合、設定パルス周期 = $200*20/20 = 200\text{ms}$
 - 入力クロックが 80MHz の場合、設定パルス周期 = $200*20/80 = 50\text{ms}$

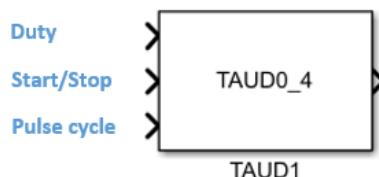


図 3-18 TAUD 周辺ブロック

次に、TAUD 周辺ブロックのブロック パラメーターについて説明します。

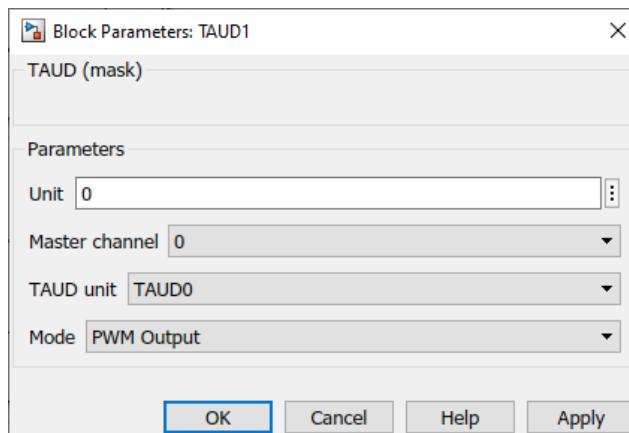


図 3-19 TAUD 周辺ブロックのブロック パラメーター

TAUD 周辺ブロックには、"Master channel"、"TAUD unit"、"Mode" パラメータがあります。"TAUD unit" を変更すると、TAUD 周辺ブロック名にも反映されます。

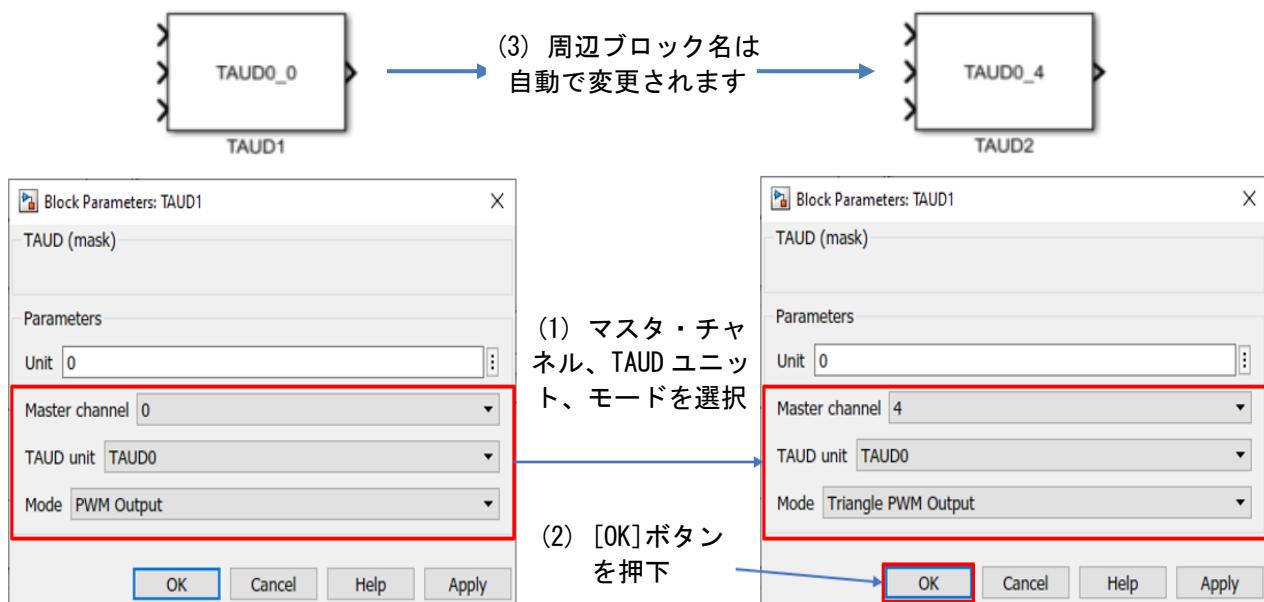


図 3-20 TAUD 周辺ブロックでのユニット名変更結果

- ユーザブロックと TAUD 周辺ブロックのモデル構造

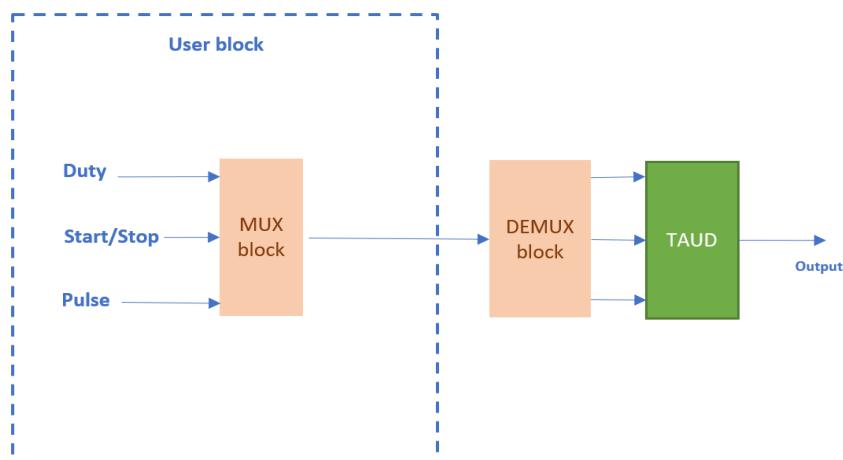


図 3-21 TAUD 周辺ブロック使用時のモデル構造

TAUD 周辺ブロックを使用する場合、上記の様にユーザブロックと TAUD 周辺ブロックを接続する必要があります。複数チャンネルの設定と MATLAB と VLAB 間の接続をサポートするには、Mux ブロックと Demux ブロックを使用してユーザブロックと TAUD 周辺ブロックを接続します。

入力と出力の配列の次元を同じに保つには、入力の “Duty” 配列(スレーブ・チャネル)の最後にマスター・チャネルをダミーとして含める必要があります。

PWM 出力開始後にデューティ比、またはパルス周期を変更しない場合、Constant ブロック(定数値)を接続してください。定数値は SC での設定値と同じにする必要があります。

TAUD 周辺ブロックの出力はデータ信号の配列です。各チャネルからデータを取得するには、Demux ブロックを使用して、配列から単一のデータ信号データを抽出する必要があります。

出力配列には、マスター・チャネルおよびスレーブ・チャネルの信号を含みます。“Duty” の入力も配列です。

- TAUD 周辺ブロックの MILS:

TAUD 周辺ブロックを追加したモデルにて MILS は実行できません。TAUD 周辺ブロックを MILS 可能なブロック (MILS 用 TAUD ブロック) に置き換えてください。

MILS 用 TAUD ブロックの入力は、” Duty ” , “ Start/Stop ” , “ Pulse ” です。MILS を実行時、MILS 用 TAUD ブロック内の S-Function “mils_taud” は、” duty ” 値を Pulse Generator ブロックの” パルス幅 ” に設定し、” Pulse ” 値を Pulse Generator ブロックの” 周期 ” に設定します。MILS 用 TAUD ブロックの入力と Pulse Generator ブロックの初期値(“パルス幅 ” と ” 周期 ”)を一致させてください。

MILS 実行の際は、Mux ブロックと Demux ブロックを使用する必要はありません。また、複数の PWM 波形を出力する場合は、必要な数のブロックを配置してください。

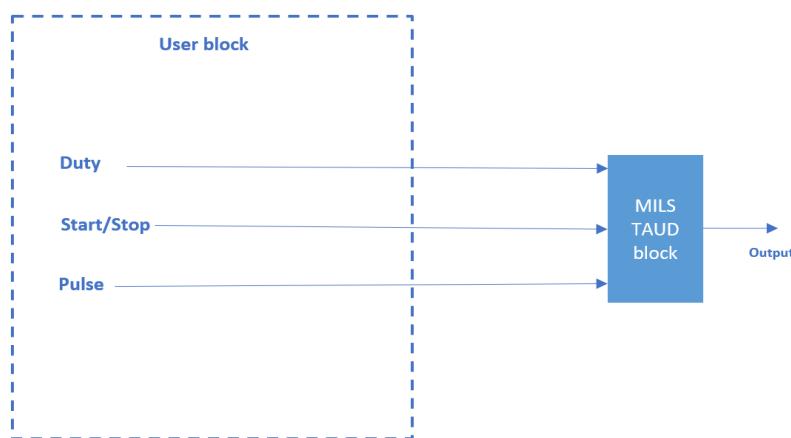


図 3-22 TAUD 周辺ブロック使用時のモデル構造 (MILS)

3.3 vHILS の実行

本節では、ET-VPF による vHILS 環境を生成する方法について説明します。

3.3.1 サンプル・モデル

ここで使用するサンプル・モデルは、MathWorks 社から提供されているパワーウィンドウ・モデル (`slexPowerWindowExample.slx`) です。各周辺ブロックはコード生成対象サブシステムの下に配置します。

- 補足**
1. コード生成対象サブシステムの最初の階層は、周辺ブロック以外のすべてのブロックはサブシステムにまとめられている必要があります。
 2. 各周辺ブロックがコード生成対象サブシステムの外にある場合はエラーになります。

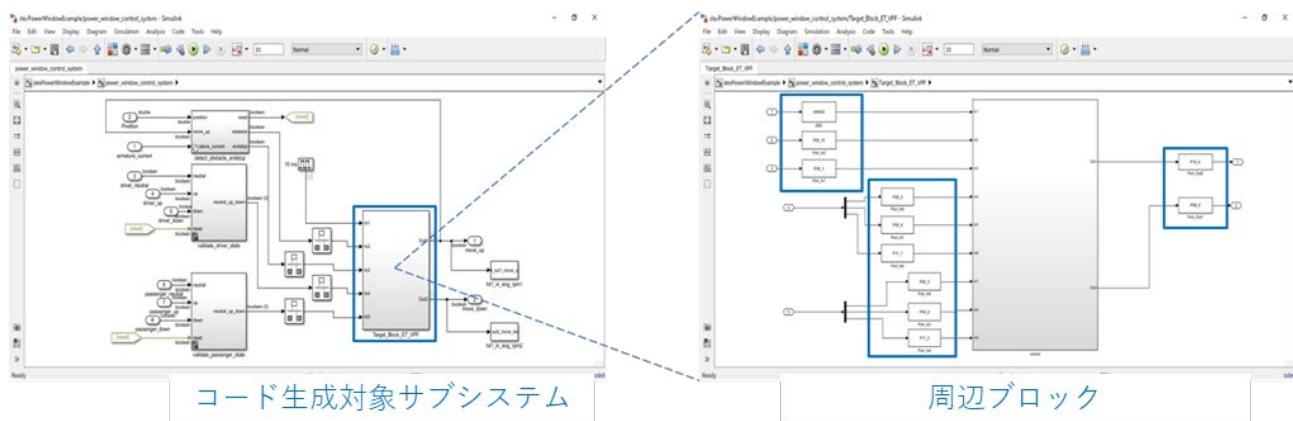


図 3-23 コード生成対象サブシステムと周辺ブロック

下表に、サンプル・モデルのコンポーネントに関する情報を示します。.

表 3.3 サンプル・モデルのコード生成対象

サンプル・モデル	コード生成対象	ブロック・タイプ
<code>slexPowerWindowExample.slx</code>	Target_Block_ET_VPF	サブシステム

表 3.4 コード生成対象下の周辺ブロック

コード生成対象	周辺ブロック	ポート名	ブロック名
Target_Block_ET_VPF	ADC	ANI<Unit><Port ID> (F1KM の場合) AN<Unit><Port ID> (U2B の場合)	ADC_<Number>
	Port	P<Port>_<Pin>	Port_In_<Number> Port_Out_<Number>
	RS-CANFD	CAN<unit>_TX CAN<unit>_RX	CAN_Transmission_<Number> CAN_Reception_<Number>
	RLIN3	RLIN3<RLIN3 unit>_Send RLIN3<RLIN3 unit>_Receive	RLIN3n_Transmission_<Number> RLIN3n_Reception_<Number>
	TAUD	TAUD<Unit>_<master channel>	TAUD_<Number>

3.3.2 コンフィギュレーション パラメーターの設定

ET-VPF は、MathWorks 製 Embedded Coder によりコード生成を行い、vHILS 実行環境を構築します。そのため、ET-VPF が提供する vHILS 環境生成機能を利用する際には、Embedded Coder の設定が必要です。

モデルに周辺ブロックを配置した後、以下の手順でコンフィギュレーション パラメーターを設定します。

- (1) MATLAB を起動します。
- (2) パワーウィンドウ・モデルがあるフォルダにてモデルを開き、モデル変数を設定し、周辺ブロックの配置と設定を行います。
- (3) [コンフィギュレーション パラメーター] ダイアログを開き、以下の設定を行います。

➤ 手順 1: [ソルバー] を選択し [タイプ] を [固定ステップ] に設定し、[適用] ボタンをクリック

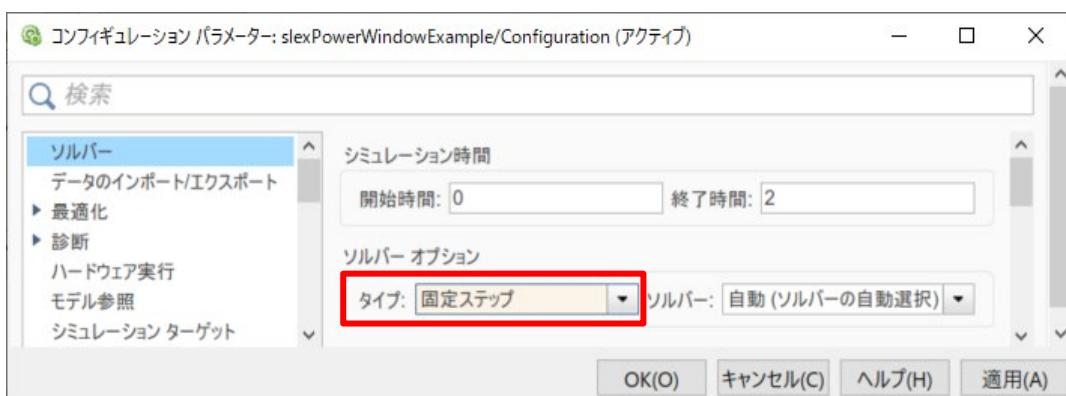


図 3-24 [ソルバー] 設定

➤ 手順 2: [コード生成] の設定

- [システム ターゲット ファイル] に “etvpf.tlc” を設定
- [コード生成のみ] にチェック

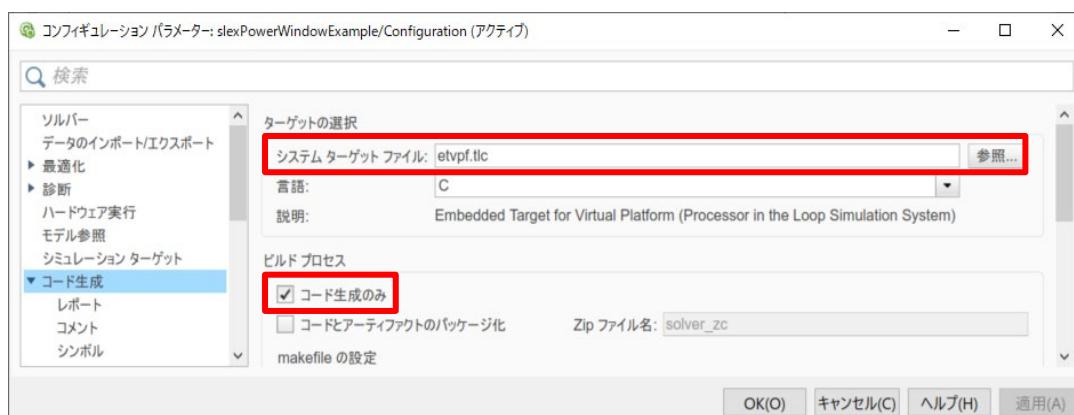


図 3-25 [コード生成] 設定

➤ 手順 3: [ET-VPF options] を設定(詳細は「表 3.5 ET-VPF Options」参照)

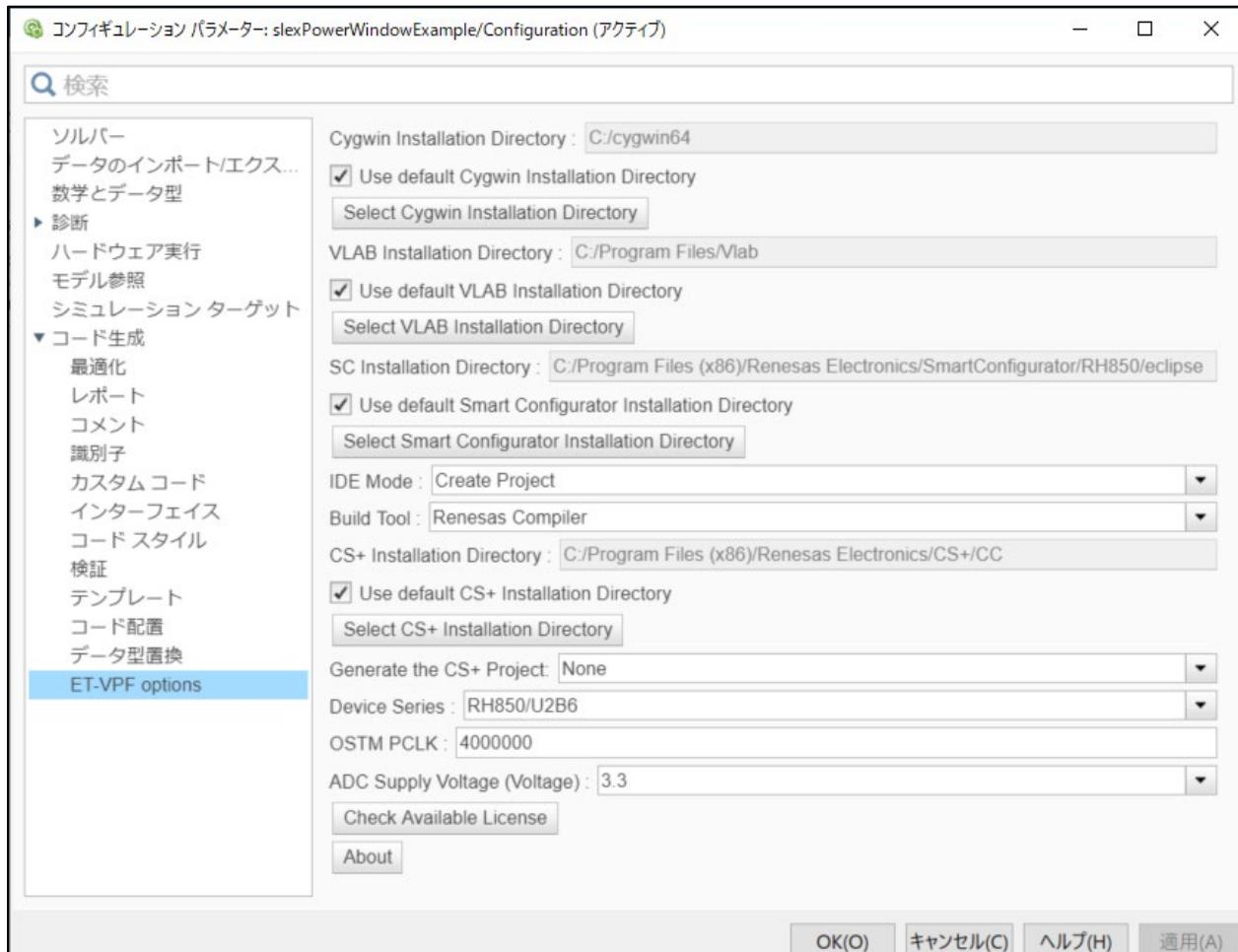


図 3-26 [ET-VPF options] 設定

下表に、[ET-VPF Options]の項目を示します。

表 3.5 ET-VPF Options

Item name	Description	
Cygwin Installation Directory *1 *15	Cygwin がインストールされているフォルダ(bin/bash.exe が格納されているフォルダ)を絶対パスとして指定します。.	
[Use default Cygwin Installation Directory] チェックボックス	Cygwin がインストールされているデフォルトのフォルダ("C:/cygwin64")を指定します。	
[Select Cygwin Installation Directory] ボタン *1 *2	このボタンをクリックすると、Cygwin がインストールされているフォルダの絶対パスを選択するためのダイアログが表示されます。ここで選択したフォルダは、[Cygwin Installation Directory]に反映されます。	
VLAB Installation Directory *3 *15	VLAB がインストールされているフォルダ(vlab-ide.exe が格納されているフォルダ)を絶対パスとして指定します。	
[Use default VLAB Installation Directory] チェックボックス	VLAB がインストールされているデフォルトのフォルダ("C:/Program Files/VLAB")を指定します。	
[Select VLAB Installation Directory] ボタン *3 *4	このボタンをクリックすると、VLAB がインストールされているフォルダの絶対パスを選択するためのダイアログが表示されます。ここで選択したフォルダは[VLAB Installation Directory]フィールドに反映されます。	
SC Installation Directory *5 *15	SC がインストールされているフォルダ(SmartConfigurator.exe が格納されているフォルダ)を絶対パスとして指定します。	
[Use default Smart Configurator Installation Directory] チェックボックス	SC がインストールされているデフォルトのフォルダ("C:/Program Files (x86)/Renesas Electronics/SmartConfigurator/RH850/eclipse")を指定します。	
[Select Smart Configurator Installation Directory] ボタン *5 *6	このボタンをクリックすると、SC がインストールされているフォルダの絶対パスを選択するためのダイアログが表示されます。ここで選択したフォルダは[SC Installation Director]に反映されます。	
IDE Mode	VLAB 起動時にロードされるプロジェクト・ファイルのタイプと、VLAB 起動後にロード・モジュールのダウンロードを含む一連の処理を実行するかどうかを選択します。	
	Create Project (デフォルト)	ET-VPF によって提供されるデフォルトのプロジェクト ファイルが使用されます。
Build Tool *7	生成されたプロジェクトのビルド・ツールを選択します。これは、コンパイラがロード・モジュールを作成するために使用されることを意味します。	
	Renesas Compiler *8	[CS+ Installation Directory]で指定された CS+に同梱されているコンパイラが使用されます。

表 3.5 ET-VPF Options (続き)

Item name	Description	
CS+ Installation Directory *9 *15	CS+がインストールされているフォルダ(CubeSuiteW+.exe が格納されているフォルダ)を絶対パスとして指定します。ルネサス製コンパイラを起動するために使用されます。	
[Use default CS+ Installation Directory] チェックボックス	CS+がインストールされているデフォルトのフォルダ(“C:/Program Files (x86)/Renesas Electronics/CS+/CC”)を指定します。	
[Select CS+ Installation Directory] ボタン (*9 *10)	このボタンをクリックすると、CS+がインストールされているフォルダの絶対パスを選択するためのダイアログが表示されます。ここで選択したフォルダは[CS+ Installation Directory]に反映されます。	
Generate the CS+ Project *16	スタンドアロン・ビルド用の CS+ プロジェクトの生成を指定します。	
	None	デフォルト値です。 CS+プロジェクト・ファイルは生成されず、ソースコードは makefile によってコンパイルされます。
	Generate the CS+ project only	CS+ プロジェクト・ファイルが生成され、ソースコードは CS+ によって手動でコンパイルが必要です。
	Generate the CS+ project and automatically build	CS+ プロジェクト・ファイルが生成され、ソースコードは CS+ によって自動でコンパイル、ビルドされます。
Device Series *11	使用するマイクロコントローラのシリーズ名を選択します。	
<Device Series Name> *12		サポートされているマイクロコントローラのシリーズ名が表示されます。
		N/A デフォルト値です。
OSTM PCLK *13	各マイクロコントローラに対応する OS タイマの動作周波数(Hz)を設定します。	
[Check Available License] ボタン	ET-VPF 使用時に必要なライセンス一覧を表示します。	
[About] ボタン	ET-VPF のバージョン情報およびコピーライ特情報を表示します。	

*1… ダイアログで指定したフォルダに Cygwin がインストールされていない（指定したフォルダに bin/bash.exe ファイルが存在しない）場合は、エラーを出力し、指定したフォルダの情報は[Cygwin Installation Directory]に反映されません。[IDE Mode]に[Create Project]が選択され、かつ

[Use default Cygwin Installation Directory] チェックボックスが選択されていない場合のみ、設定が有効になります。

*2… [Use default Cygwin Installation Directory] チェックボックスを選択した場合、[Select Cygwin Installation Directory]ボタンをクリックすると、エラー・メッセージが表示されます。

*3… ダイアログで指定したフォルダに VLAB がインストールされていない（指定したフォルダに vlab-ide.exe ファイルが存在しない）場合は、エラーを出力し、指定したフォルダの情報は[VLAB Installation Directory]に反映されません。[IDE Mode]に[Create Project]が選択され、かつ

[Use default VLAB Installation Directory] チェックボックスが選択されていない場合のみ、設定が有効になります。

*4… [Use default VLAB Installation Directory] チェックボックスを選択した場合、[Select VLAB Installation Directory]ボタンをクリックすると、エラー・メッセージが表示されます。

- *5… ダイアログで指定したフォルダに SC がインストールされていない（指定したフォルダに SmartConfigurator.exe ファイルが存在しない）場合は、エラーを出力し、指定したフォルダの情報は[SC Installation Directory]に反映されません。[IDE Mode]に[Create Project]が選択され、かつ [Use default SC Installation Directory] チェックボックスが選択されていない場合のみ、設定が有効になります。
- *6… [Use default Smart Configurator Installation Directory] チェックボックスを選択した場合、[Select Smart Configurator Installation Directory] ボタンをクリックすると、エラー・メッセージが表示されます。
- *7… [IDE Mode]に[Create Project]が選択されている場合のみ、設定が有効になります。
- *8… [Build Tool]を“Renesas Compiler”に設定した場合、ビルド・ツールはソースコードのコンパイル時に CS+によって決定されます。
- *9… ダイアログで指定したフォルダに CS+がインストールされていない（指定したフォルダに CubeSuiteW+.exe ファイルが存在しない）場合は、エラーを出力し、指定したフォルダの情報は[CS+ Installation Directory]に反映されません。[IDE Mode]に[Create Project]が選択され、かつ [Use default CS+ Installation Directory] チェックボックスが選択されていない場合のみ、設定が有効になります。
- *10… [Use default CS+ Installation Directory] チェックボックスを選択した場合、[Select CS+ Installation Directory] ボタンをクリックすると、エラー・メッセージが表示されます。
- *11… [IDE Mode]に[Create Project]が選択され、かつ ET-VPF のライセンスが有効な場合のみ、設定が有効になります。
- *12… 「表 1.1 サポートデバイス」参照
- *13… 選択した[Device Series]に応じて、以下の値を設定する必要があります。これらの値はデフォルト値として設定されます。

表 3.6 OSTM PCLK 設定値

Device Series	OSTM PCLK [Hz]
RH850/F1KM-S1	4000000
RH850/F1KM-S4	2000000

- *14… U2B の場合、OSTM PCLK は CLKC_HSB クロックから派生しますが、「3.3.3.1 Smart Configurator による周辺コード生成」を参照して CLKC_HSB クロックを変更できます。ただし、変更後の値が ET-VPF options の[OSTM PCLK]の設定値と同じにする必要があります。CLKC_HSB クロックの定義は対象デバイスによって異なるため、ハードウェア・マニュアルの“Configuration of Clock Controller”を参照してください。

*15 … CS+、Cygwin、Smart Configurator、VLAB のインストール・パスは、以下の表に記載されている特殊文字のみをサポートしています。

表 3.7 使用可能な特殊文字

特殊文字	CS+、Cygwin	Smart Configurator	VLAB	ET-VPF インストール・フォルダ
!	○	○	○	○
@	○	○	○	○
#	-	-	○	-
\$	-	○	○	-
%	-	-	-	○
^	-	-	○	○
&	-	-	○	○
~	○	○	○	○
`	-	○	○	○
-	○	○	○	○
-	○	○	○	○
+	○	○	○	○
=	-	○	○	○
(-	○	○	-
)	-	○	○	-
[○	○	○	○
]	○	○	○	○
{	○	○	○	○
}	○	○	○	○
,	-	-	-	-
.	○	○	○	-
'	-	-	-	-
,	-	○	○	-
;	-	-	-	-
空白	○	○	○	-

* ○ : 使用可能

- : 使用不可

*16… “Generate the CS+ project only” または “Generate the CS+ project and automatically build” を選択した場合、“Build Tool” には “Renesas Compiler” が自動的に設定されます。

- 手順 4: [Apply] ボタンを押下し、モデルをセーブ
- 手順 5: [OK] ボタンあるいは[X]ボタンを押下し、[コンフィギュレーション パラメーター] ダイアログを閉じる

3.3.3 vHILS 環境の生成

本節では、vHILS 実行に必要な vHILS 環境を構築する方法について説明します。

ET-VPF は、MATLAB コマンド・ウィンドウで使用できる次のコマンドが用意されています。このコマンドは、vHILS 環境を生成するための一連の操作を自動的に実行します。

表 3.8 提供コマンド

コマンド名	説明
run_vlab	vHILS 環境の自動構築および vHILS の自動実行

- (1) モデル中のコード生成対象サブシステムを選択します。
- (2) vHILS 環境を構築するには、MATLAB コマンド・ウィンドウに以下の構文で、指定されたコマンドを入力します。
“>>” はコマンド・プロンプトを、“[Enter]” はエンター・キーの入力を表しています。

例：

```
>> run_vlab [Enter]
```

- (3) “run_vlab” コマンドを実行すると、SC_project フォルダが作成されます。また、SC_project のフルパスについてユーザに通知するためのメッセージボックスが表示されます。

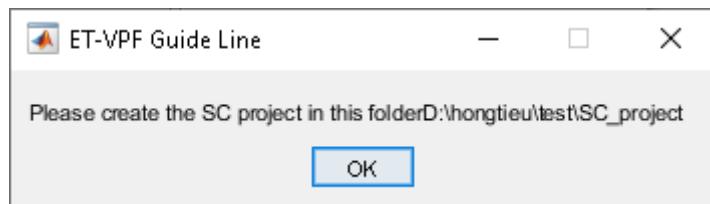


図 3-27 SC_project フルパス通知

3.3.3.1 Smart Configurator による周辺コード生成

(1) “run_vlab” コマンド実行後、SC ウィンドウが表示されます。ユーザは以下のように詳細設定を行い、周辺機能のソースコードを生成できます：

- 手順 1：SC のプロジェクトを生成するため、[新規コンフィグレーションファイル]ボタンを押下、あるいはメニュー [ファイル] - [新規…] を選択

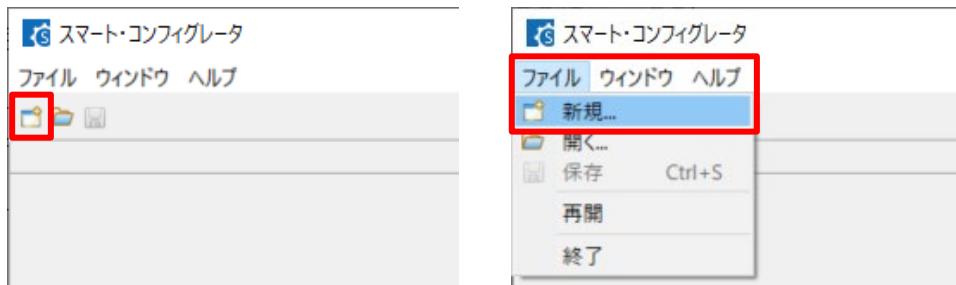


図 3-28 新規 SC プロジェクトの生成

- 手順 2：プラットフォームとツールチェーンの設定を選択：

- (1) デバイス名を選択 (F1KM-S1、F1KM-S4、U2B6、U2B10 のみ選択可能)
- (2) ツールチェーンを選択 (Renesas CCRH850 Toolchain のみ選択可能)
- (3) ファイル名に”SC_project”を入力し、ロケーションを選択(「図 3-27 SC_project フルパス通知」で表示されたフルパスを選択)
- (4) 終了ボタンを押下

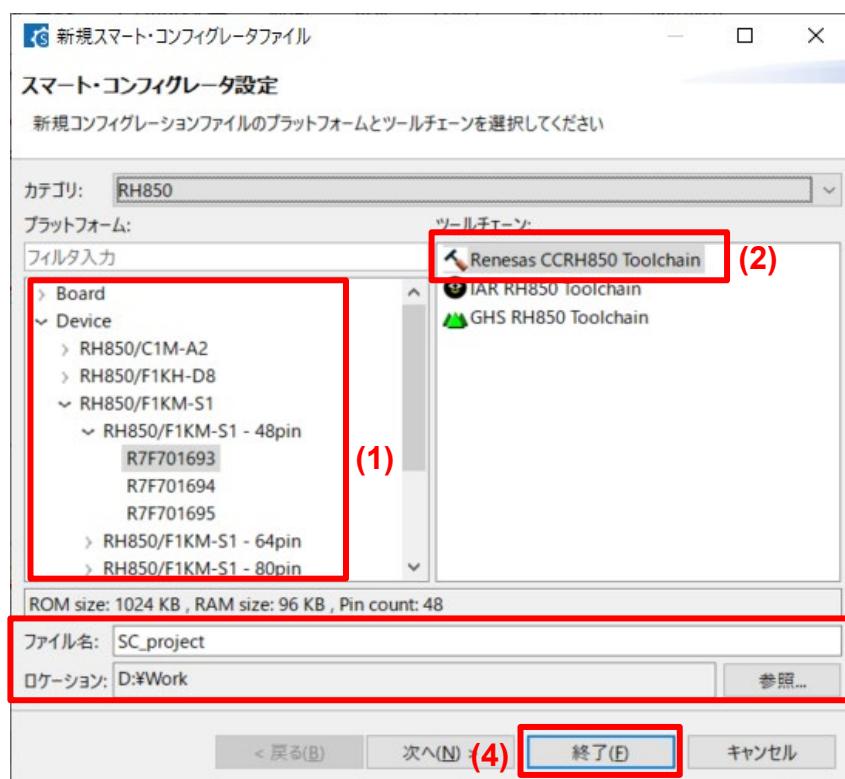


図 3-29 SC 設定の選択

- 手順 3: SC のコンフィグレーション情報を含む XML ファイルを SC にインポートします（モデルに CAN 周辺ブロックのみが含まれている場合は、この手順はスキップします）。
- SC の各コンポーネントを簡単にインポートできる XML ファイル（“newxml.xml”）が作業フォルダに生成されます。
 - XML ファイルをインポートするには、下図の通り、
 - (1) [コンポーネント] タブを選択します。
 - (2) [コンフィグレーションのインポート] ボタンを押下します。

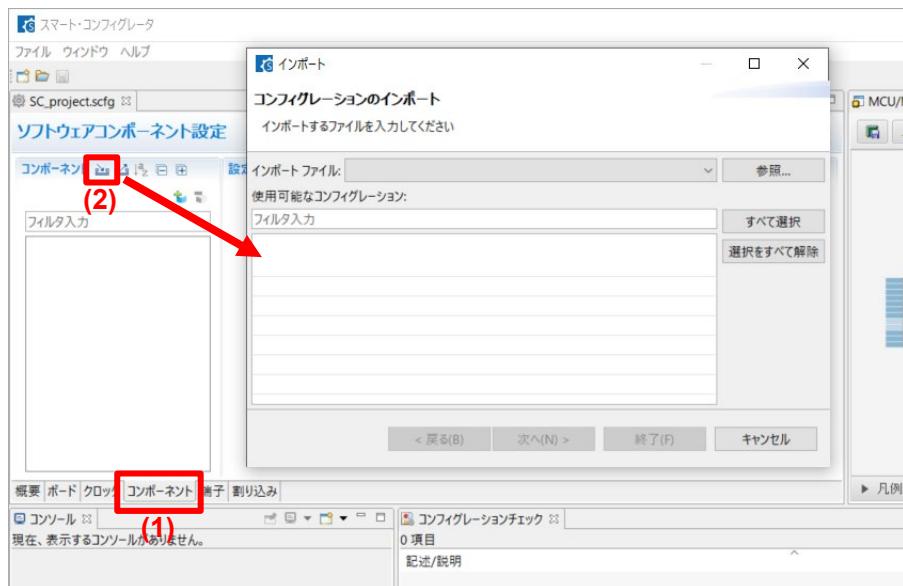


図 3-30 コンフィグレーションのインポート

- 次に、下図の通り
- (1) インポートファイルに XML ファイル（“newxml.xml”）を指定します。
- (2) [すべて選択] ボタンを押下します（使用しないコンフィグレーションはチェックを外してください）。
- (3) [次へ] ボタンを押下します。

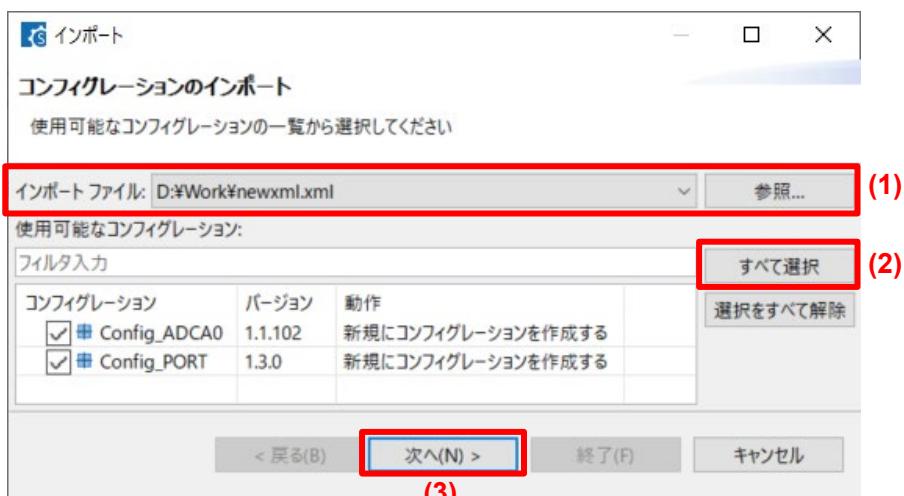


図 3-31 XML ファイルのインポート

- [終了]ボタンを押下すると、コンフィグレーションのインポートが完了します。

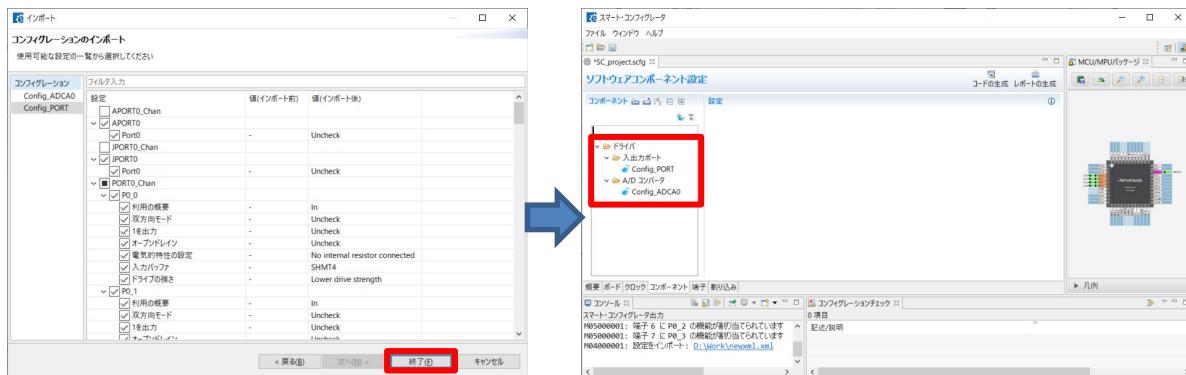


図 3-32 コンフィグレーションのインポートの完了

補足

- ET-VPF が vHILS 実行中に呼び出す、SC が生成した周辺コードの関数名は決められた形式である必要があります。そのため、SC にて各周辺機能のコンポーネントを追加する際、コンフィグレーション名はデフォルト名でなければなりません(例 : Config_ADCAO)。
- デバイスが搭載している各 RLIN3n ユニット 1 つに対して、UART リソースは 1 つだけ割り当ててください(例えば、RLIN30 は "UART0" リソースに割り当てる)。同じ UART リソースで 2 つの RLIN3n ユニットを使用すると、エラーが発生します。
- TAUD 周辺ブロックを使用する場合、SC のコンポーネントの設定にて、クロック、およびスレーブ・チャネルを指定する必要があります。

➤ 手順 3 の補足 クロック設定(任意)

クロックを更新するには、SC の [クロック] タブに移動します。クロック設定パネルでクロックを変更できます。

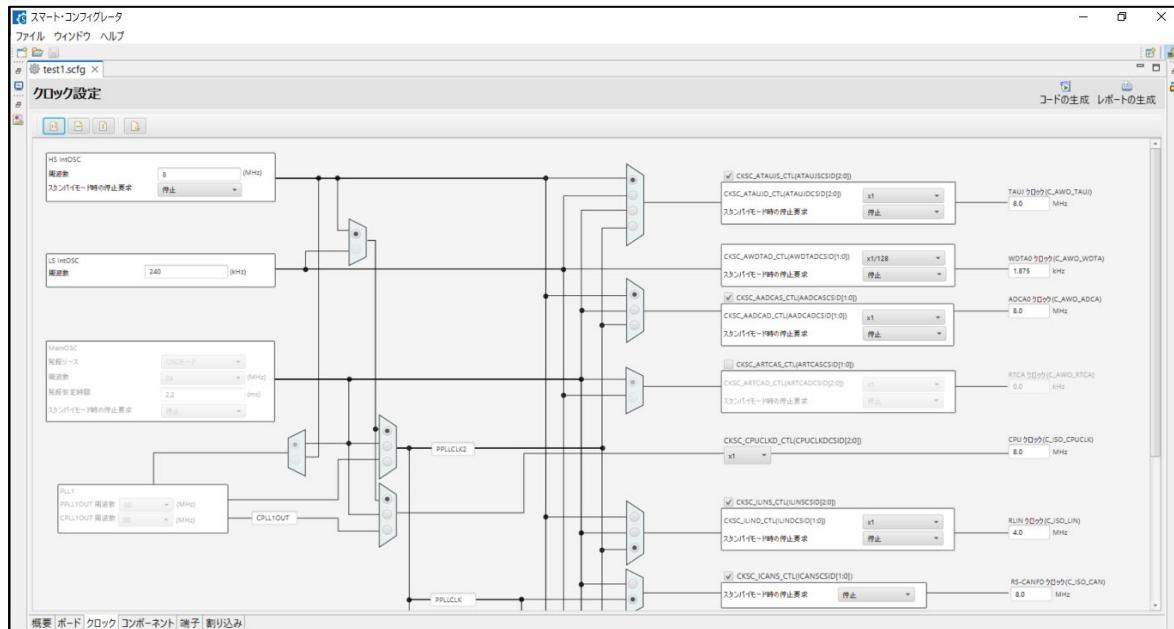


図 3-33 クロック設定パネル(F1KM の場合)

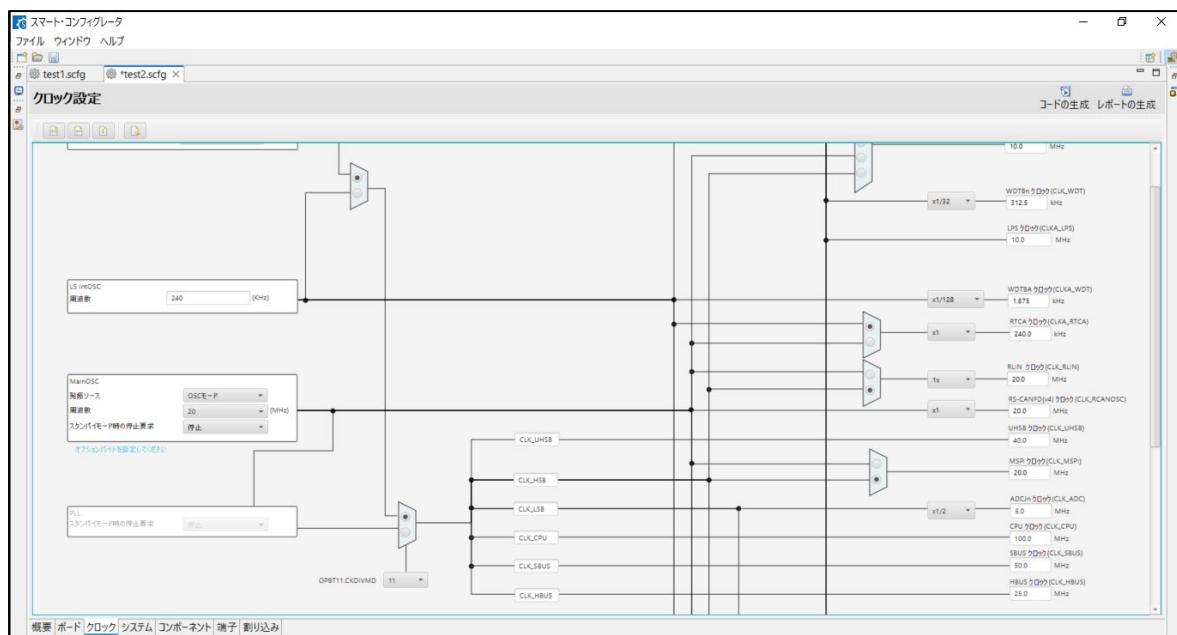


図 3-34 クロック設定パネル(U2B の場合)

- 手順 4: コンポーネントのコンフィグレーションを変更し(モデルに CAN 周辺ブロックのみが含まれている場合は除く)、周辺コードを生成します。

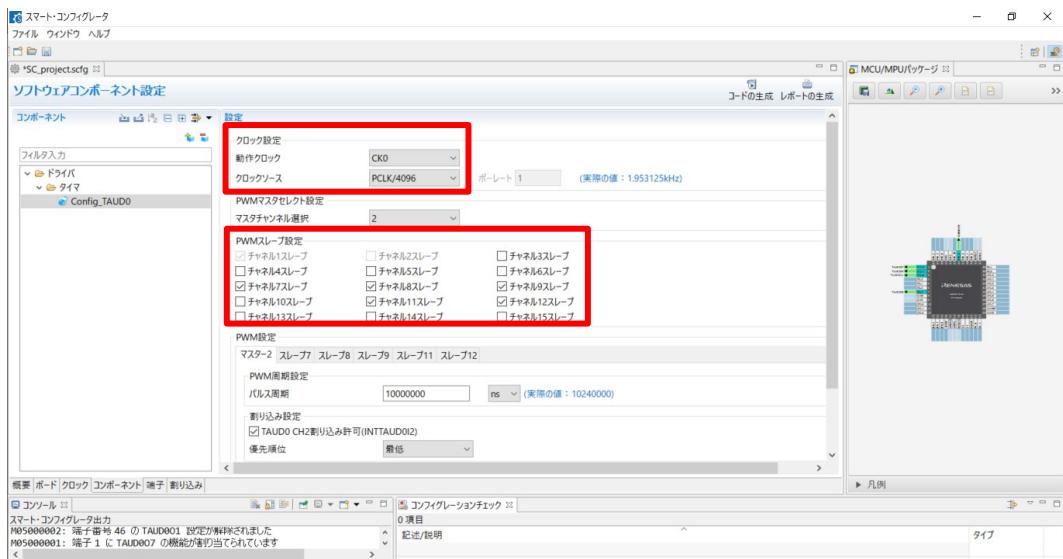


図 3-35 TAUD のクロックソースとスレーブ・チャネルの設定

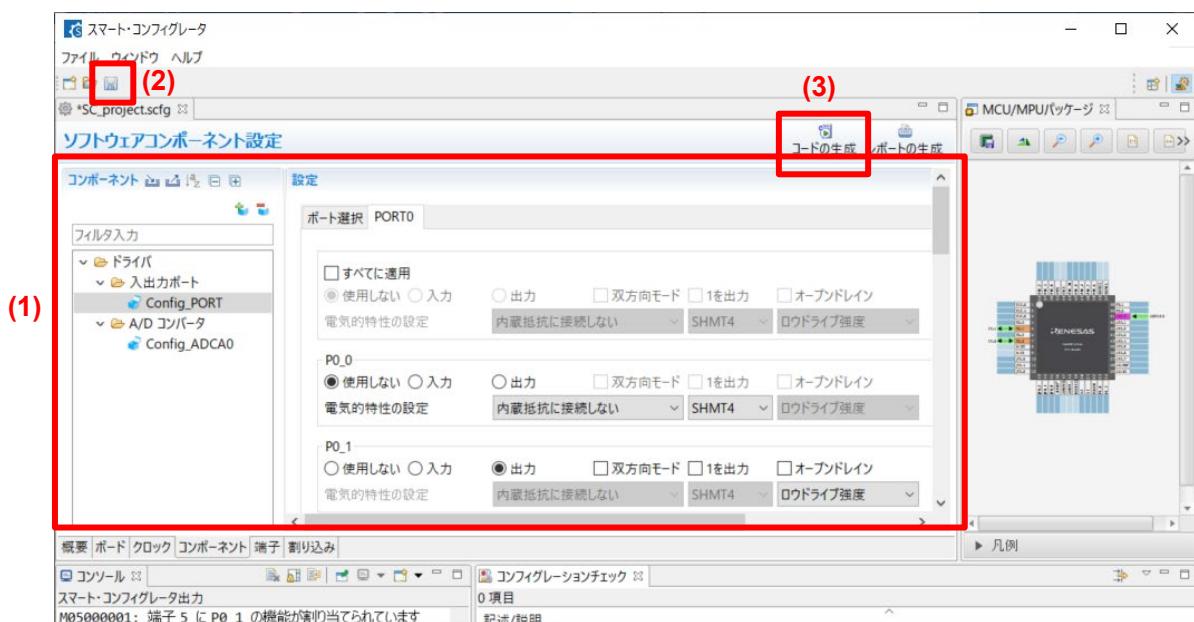


図 3-36 コンポーネントの詳細設定と周辺コードの生成

補足

詳細設定を変更する場合、モデル上の周辺ブロックの設定と同じにする必要があります。変更した設定がモデル上の周辺ブロックの設定と異なる場合、vHILS 実行が正常に行えません。

➤ 手順 5: SC ウィンドウを閉じる。

(2) 周辺コードは、SC_project フォルダの下に生成されます。

> work > SC_project > src > smc_gen			
名前	更新日時	種類	
Config_ADCA0	2022/06/02 18:37	ファイル フォルダー	
Config_PORT	2022/06/02 18:37	ファイル フォルダー	
general	2022/06/02 18:37	ファイル フォルダー	
r_pincfg	2022/06/02 18:37	ファイル フォルダー	

図 3-37 生成される周辺コード

3.3.3.2 vHILS 環境の生成

vHILS 環境は、ターゲットモデルを含むフォルダ(以下、作業フォルダ)に生成されます。vHILS 環境には、“slprj” フォルダと “<コード生成対象サブシステム名>_etvpf” フォルダが含まれ、vHILS 実行に必要な以下のファイルはこれらのフォルダ下に生成されます。

- ターゲットデバイス用ソース・ファイル
- C ソース・ファイル
- Makefile
- VPF 実行用ファイル
- 周辺コード用のラッパー・ソース・ファイル
- Python ファイル
- Define.h、OSTM_define.h、target.out

モデルファイルは別名でコピーされます(コピー先のモデルファイルは、元のモデルファイル名にサフィックス”_etvpf” が付加されます)。

コード生成対象サブシステムの下のサブシステムは、vHILS 実行用ブロック(“VLAB Bridge”)に置き換えられます。

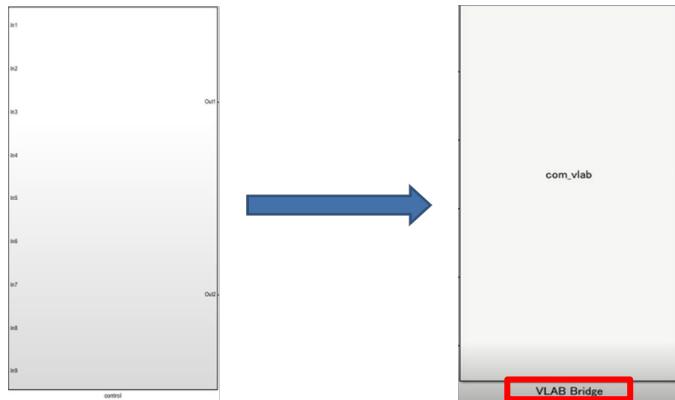


図 3-38 vHILS 実行ブロックへの置き換え例

3.3.4 生成コードのコンパイル

- (1) その後、[Generate the CS+ Project]が“None”に設定されている場合、C ソース・ファイルとターゲットデバイスのソース・ファイル(拡張子は”.c”、“.asm”)は、Cygwin の make コマンドにより、オブジェクトファイル(拡張子は”.o”)にコンパイルされます。コンパイルに関する情報は、MATLAB コマンド・ウィンドウに表示されます

```
Command Window
In MATLAB (line 1)
*** Finished generating files ***
*** Build and run co-simulation ***
*****
make[1]: Entering directory '/cygdrive/h/99_work/hongtieu/ETVPF_package/Sample_model_m9/work/Target_Block_ET_VPF_etvpf'
Makefile:109: H:/99_work/hongtieu/ETVPF_package/Sample_model_m9/work/Target_Block_ET_VPF_etvpf/MakeDepend.mk: No such file or directory
*** Create target dependency definitions done. ***
*****
Compiling etvpf_main.c ...
Generating etvpf_main.o ...
D:/Workspace/01_CS+/CS+V8.07.00/CS+/CC/CC-RH/V2.04.00/bin/ccrh.exe -Xcpu=g3kh -g -c -g_line -Xmsg_lang=english -lang=c99 -IH:/99_wo
Done ...
*****
Compiling H:/99_work/hongtieu/ETVPF_package/ETVPF_include/Renesas/Common_files_F1KM_S1/src/define_interrupt.c ...
Generating define_interrupt.o ...
D:/Workspace/01_CS+/CS+V8.07.00/CS+/CC/CC-RH/V2.04.00/bin/ccrh.exe -Xcpu=g3kh -g -c -g_line -Xmsg_lang=english -lang=c99 -IH:/99_wo
Done ...
*****
f1 *****
```

図 3-39 コンパイル状況の表示

次に、[コンフィギュレーション パラメーター]ダイアログの [ET-VPF options]-[IDE mode] を [Create Project] に設定した場合の、CS+プロジェクトの生成とビルドについて説明します。

- (a) [Generate the CS+ Project] が “Generate the CS+ project only” に設定されている場合、ET-VPF CS+プロジェクトを自動生成します。

以下に、CS+で必要な設定と CS+のプロジェクト構築について説明します。

手順 1: CS+にて、コンパイラやアセンブラーなどのビルド・ツールのオプション設定を行います。

ターゲットデバイス名は、SC の設定と同じ設定にします（「[3.3.3.1 Smart Configurator による周辺コード生成](#)」を参照）。なお、以下の場合は CS+の初期設定を変更してください。

[F1KM および Renesas Compiler 使用時]

CS+のプロジェクト・ツリーで [ビルド・ツール] を選択し、[コンパイル・オプション]-[出力コード]-[浮動小数点演算方法] に “ソフトウェアで行う” を設定してください。

補足 CS+にて他の設定を変更し、CS+のプロジェクトをビルドすることができますが、ET-VPF は CS+の設定が適切であるか判断しないため、予期しない動作が発生する可能性があります。

手順 2: CS+のプロジェクトを保存します。

手順 3: CS+メニュー [ビルド] から “ビルド・プロジェクト” を押下します。

補足 ロード・モジュールの生成については、「[CS+ユーザーズマニュアル CC-RH ビルド・ツール操作編](#)」を参照してください。

手順 4: CS+を閉じます。

- (b) [Generate the CS+ Project] を “Generate the CS+ project and automatically build” に設定した場合、vHILS 環境生成一連の動作として、CS+プロジェクトの生成とビルドが自動的に行われます。ビルド完了後は、CS+のウィンドウも自動的に閉じるため、ユーザによる操作は不要です。

補足 “Generate the CS+ project and automatically build” 選択時は、CS+のデフォルト設定で自動ビルドを行うため、ユーザによる設定変更はできません。

- (2) 最後に、“target.out” ファイルが生成されます。オブジェクトファイルと “target.out” ファイルは作業フォルダの”<コード生成対象サブシステム名>_etvpf” フォルダに格納されます。

3.3.5 vHILS の実行

- (1) コンパイルが完了すると、VLAB ウィンドウが開き、ターゲットデバイスのツール・ボックスを読み込みます。その後、“target.out”、作業ディレクトリ内のソース・ファイルをロードします。
- (2) vHILS の実行は、Simulink モデルと VLAB の両方で開始されます。MATLAB と VLAB 間のデータの送受信は、サンプリング時間後に周期的に行われます。

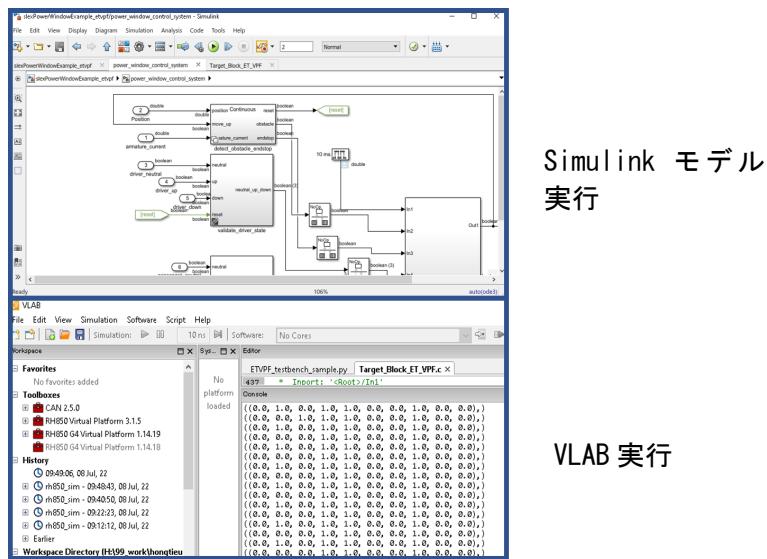


図 3-40 vHILS 実行

- (3) 下図は Scope ブロックに表示された実行結果の例です。

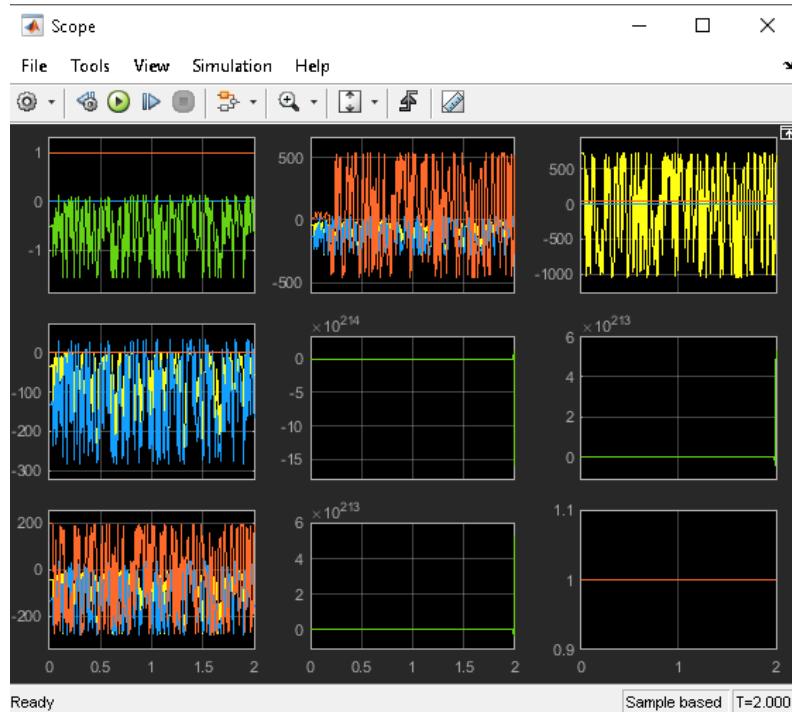


図 3-41 実行結果例

補足 最初のステップの実行結果は無視してください。最初のステップで MATLAB と VLAB が接続される、MATLAB

はダミーの値で実行を行うためです。

3.4 時間計測

本節では、ET-VPF の時間計測方法について説明します。

3.4.1 時間計測用 Simulink モデルの構造

ET-VPF で時間計測対象となるサブシステムは、以下のすべての要件を満たす必要があります。

1. 下図に示すユーザブロック内のブロックであること
2. Atomic サブシステムであること
3. 「3.4.2 時間計測のための入力ファイル」で説明する `input_subsystem.txt` で定義されているブロックであること

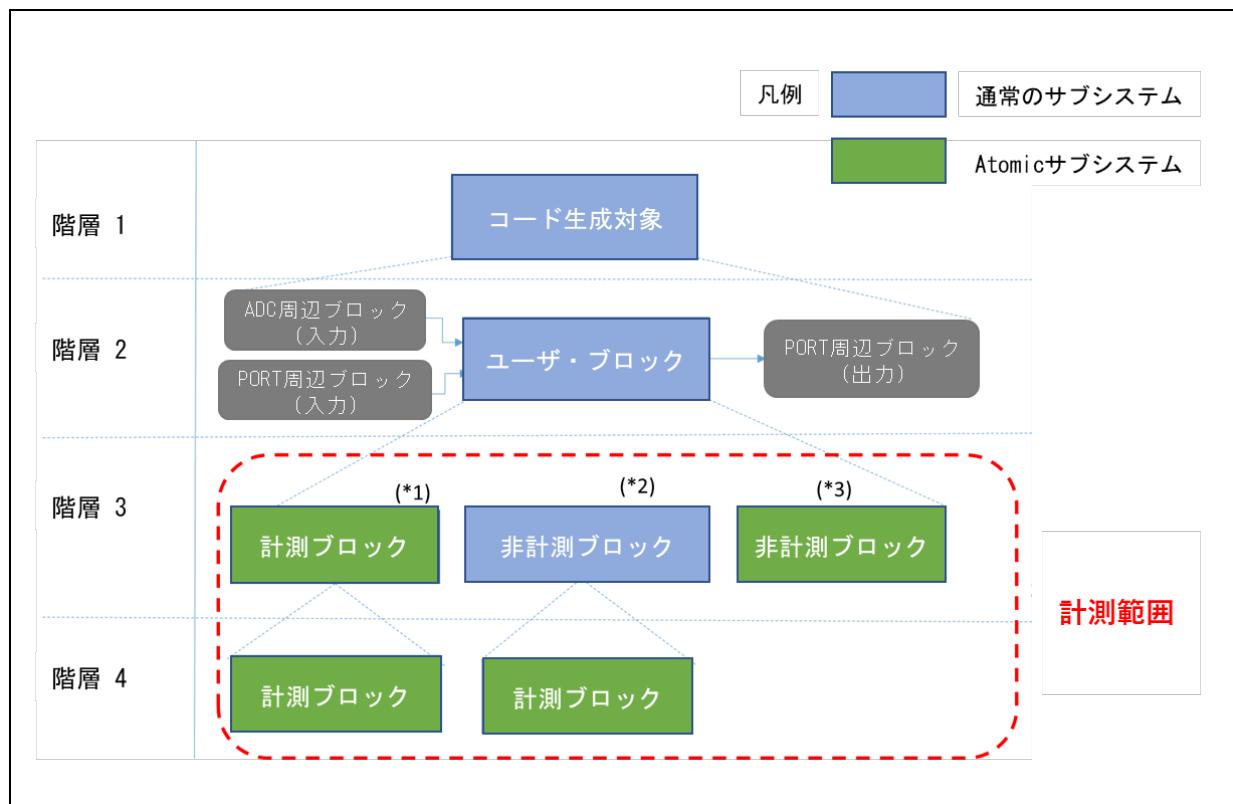


図 3-42 時間計測用 Simulink モデルの構造

補足

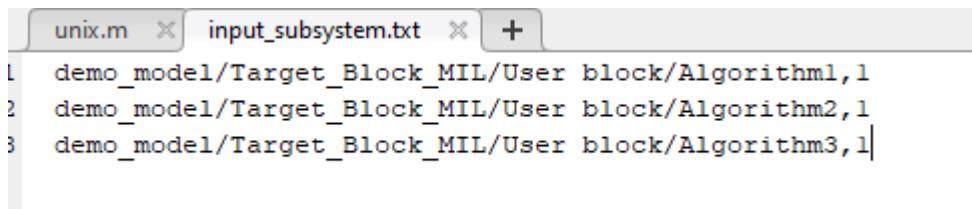
- *1 : 計測対象になります。
- *2 : Atomic サブシステムではないため、計測対象ではありません。Atomic サブシステムに変換するには、「3.4.3 Atomic サブシステム変更方法 3.4.3 Atomic」を参照してください。
- *3 : Atomic サブシステムですが、`input_subsystem.txt` ファイルで定義されていない場合、計測対象にはなりません。

3.4.2 時間計測のための入力ファイル

時間計測対象としたいサブシステムは、以下のフォーマットでファイル `input_subsystem.txt` に記述します。

```
<計測対象 Atomic サブシステムのパス 1>, <最初のコア>[Enter]
<計測対象 Atomic サブシステムのパス 2>, <最初のコア>[Enter]
...
<計測対象 Atomic サブシステムのパス N>, <最初のコア>[Enter]
```

例 :



```
unix.m ✘ input_subsystem.txt ✘ +  
1 demo_model/Target_Block_MIL/User block/Algorithm1,1  
2 demo_model/Target_Block_MIL/User block/Algorithm2,1  
3 demo_model/Target_Block_MIL/User block/Algorithm3,1
```

図 3-43 時間測定サブシステム指定ファイル `input_subsystem.txt` の例

3.4.3 Atomic サブシステム変更方法

デフォルト設定のサブシステムを Atomic サブシステムに変更するには、次の手順を実行します。

- (1) デフォルト設定のサブシステムを右クリックし、[ブロック パラメーター]を選択
- (2) [メイン]タブにて”Atomic サブシステムとして扱うにチェック
- (3) [コード生成]タブにて”再利用可能な関数”を選択

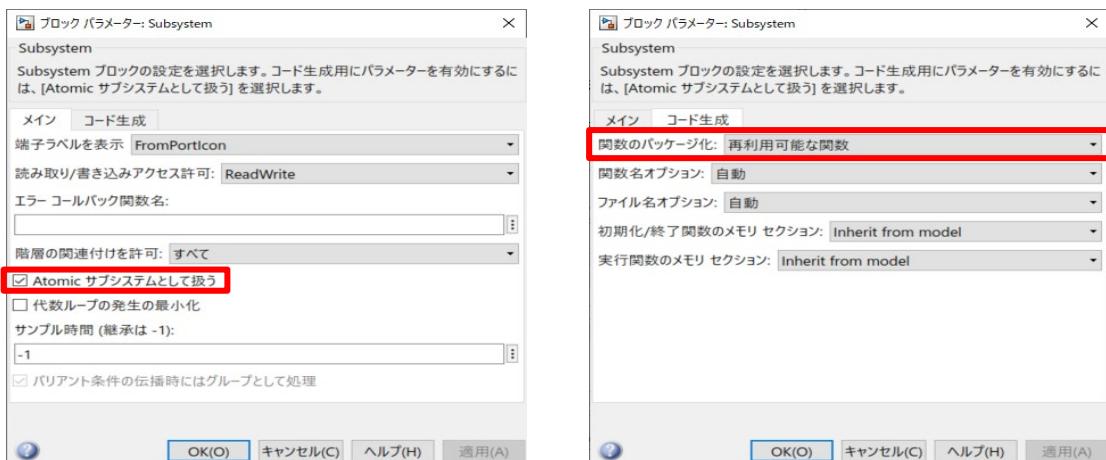


図 3-44 デフォルト設定のサブシステムから Atomic サブシステムへの変更

3.4.4 Graph Viewer

本節では、Graph Viewer を使った時間計測結果の表示方法について説明します。

3.4.4.1 入力データ

Graph Viewer の入力ファイルは以下の 2 つです。

(1) 時間計測結果ファイル : execution_data.csv

- ・ 時間単位はピコ秒です。
- ・ 各ステップの総実行時間と各サブシステムの開始/終了時間が含まれます。

	A	B	C	D	E	F	G	H	I	J
1	Target_Block_vHIL_step			Target_Block_vHIL_Algorithm1		Target_Block_vHIL_Algorithm2	Target_Block_vHIL_Algorithm3			
2				Start	End	Start	End	Start	End	
3	Step	Valid	Time							
4	1	1	1 68971500	67089000	67479000	67491500	67649000	68501500	68521500	
5	2	1	1 68971500	67089000	67479000	67491500	67649000	68501500	68521500	
6	3	1	1 68971500	67089000	67479000	67491500	67649000	68501500	68521500	
7	4	1	1 69001500	67119000	67509000	67521500	67679000	68531500	68551500	
8	5	1	1 68971500	67089000	67479000	67491500	67649000	68501500	68521500	
9	6	1	1 68971500	67089000	67479000	67491500	67649000	68501500	68521500	
10	7	1	1 68966500	67084000	67474000	67486500	67644000	68496500	68516500	
11	8	1	1 69001500	67119000	67509000	67521500	67679000	68531500	68551500	

(2) 計測対象サブシステムを指定したファイル : input_subsystem.txt (「3.4.2 時間計測のための入力ファイル」参照)

3.4.4.2 グラフ表示

Graph Viewer (ETVPF_Launch_GraphViewer.m) を使用し時間計測結果を表示するには、MATLAB のコマンド・ウィンドウから以下のコマンドを実行します。

“>>” はコマンド・プロンプトを、“[Enter]” はエンター・キーの入力を表しています。

```
>> ETVPF_Launch_GraphViewer [Enter]
```

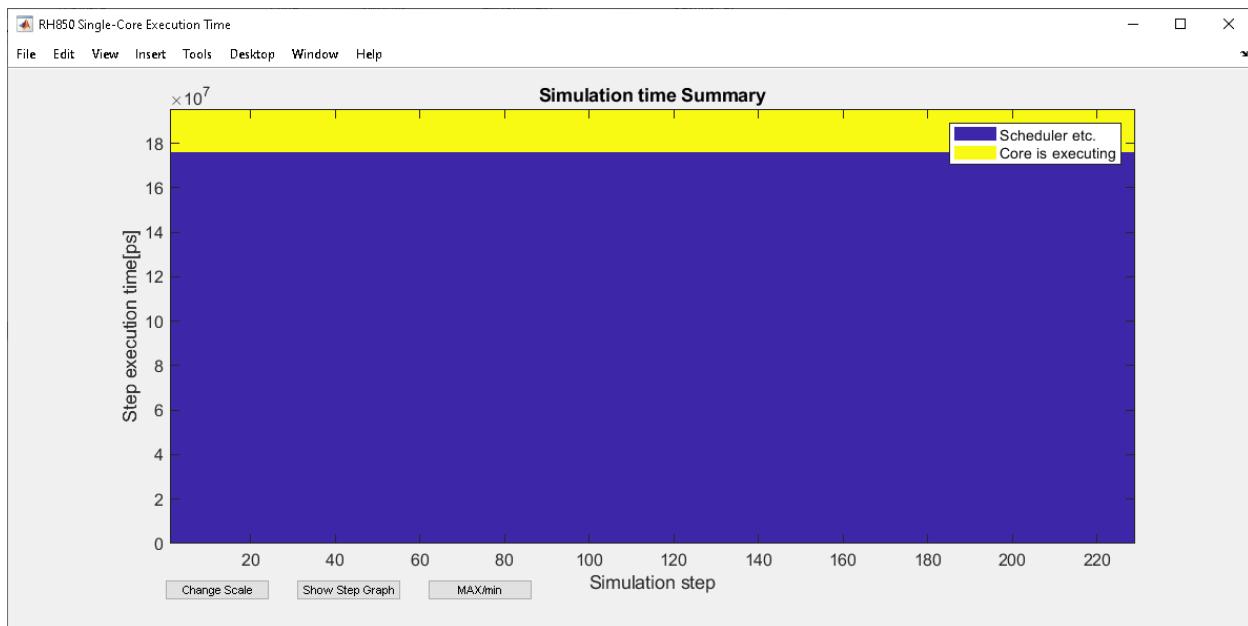


図 3-45 実行時間の表示例

補足 現状、Graph Viewer は周辺コードの実行時間は含んでいません。

ユーザは、Graph Viewer 下のボタンにより、以下の機能を利用可能です。

(1) 指定区間ステップ表示(詳細表示)

[Start Step]から[End Step]までの各サブシステムの実行時間を表示します。

[Show Step Graph]ボタンをクリックして、[Start Step]および[End Step]を設定して(例: Start Step = 1, End Step = 1)、表示するステップの区間を変更してください。

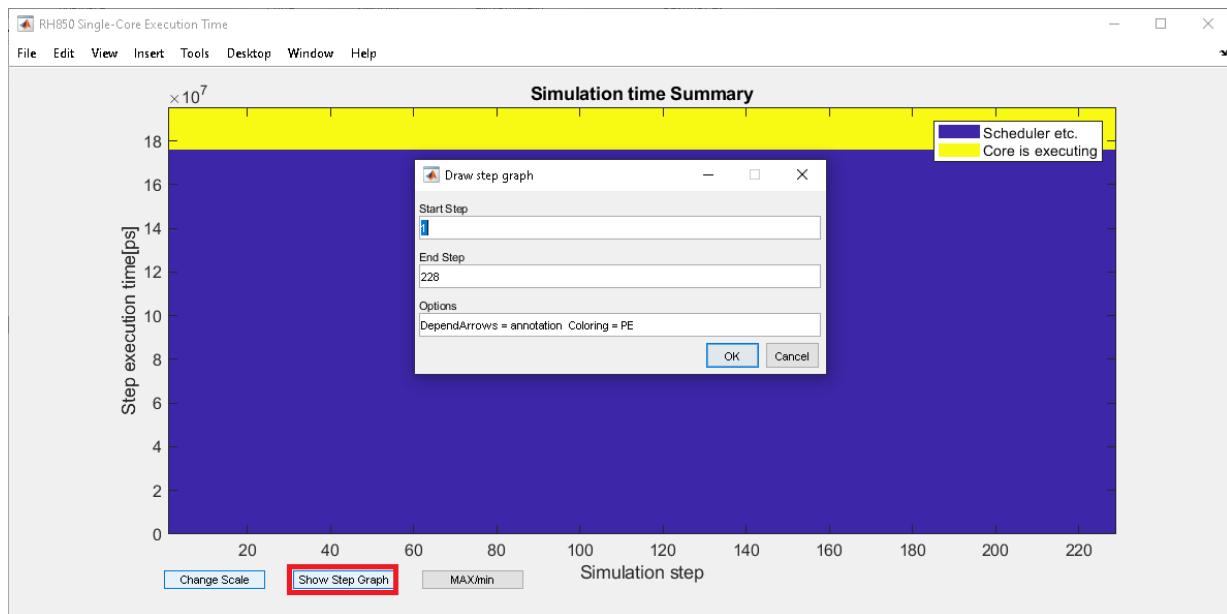


図 3-46 指定区間の各サブシステムの実行時間表示

[OK]をクリックすると、指定した区間のステップの各サブシステムの実行時間が表示されます。

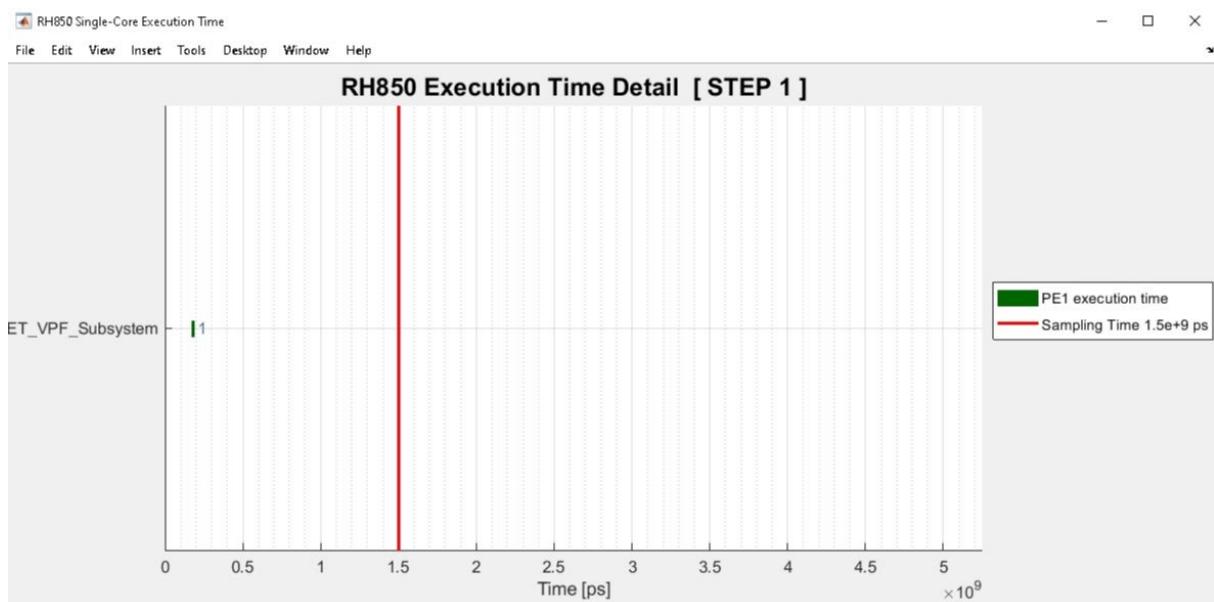


図 3-47 Step1 のサブシステム単位の実行時間表示例

(2) 指定区間ステップ表示(要約表示)

[Start Step]から[End Step]までのシミュレーション時間の要約を表示します。

[Change Scale]ボタンをクリックし、[Start Step]および「End Step」を設定して（例：Start Step = 1, End Step = 228）、表示するステップの区間を変更してください。

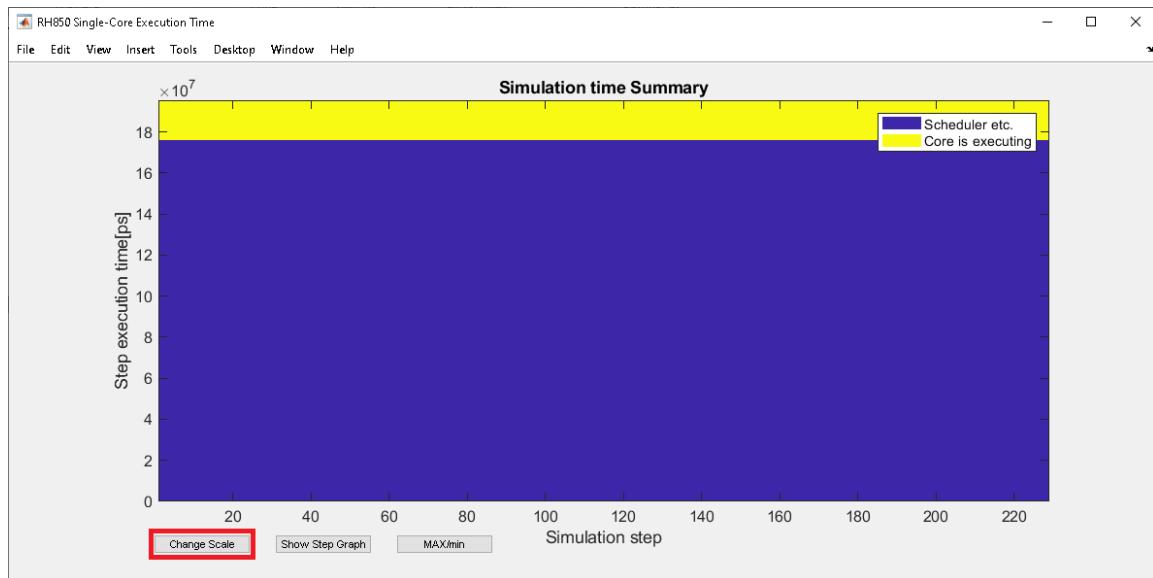


図 3-48 指定区間ステップ表示

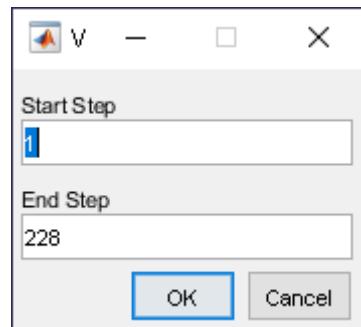


図 3-49 開始および終了ステップの値の設定

[OK]をクリックすると、指定した区間のステップが表示されます。

(3) 最悪実行時間表示または最小実行時間表示

最悪実行時間または最小実行時間となるステップを表示します。

[MAX/min] ボタンをクリックすると [Max, min] ダイアログに、最悪実行時間、および最小実行時間となるステップ値と実行時間が表示されます。

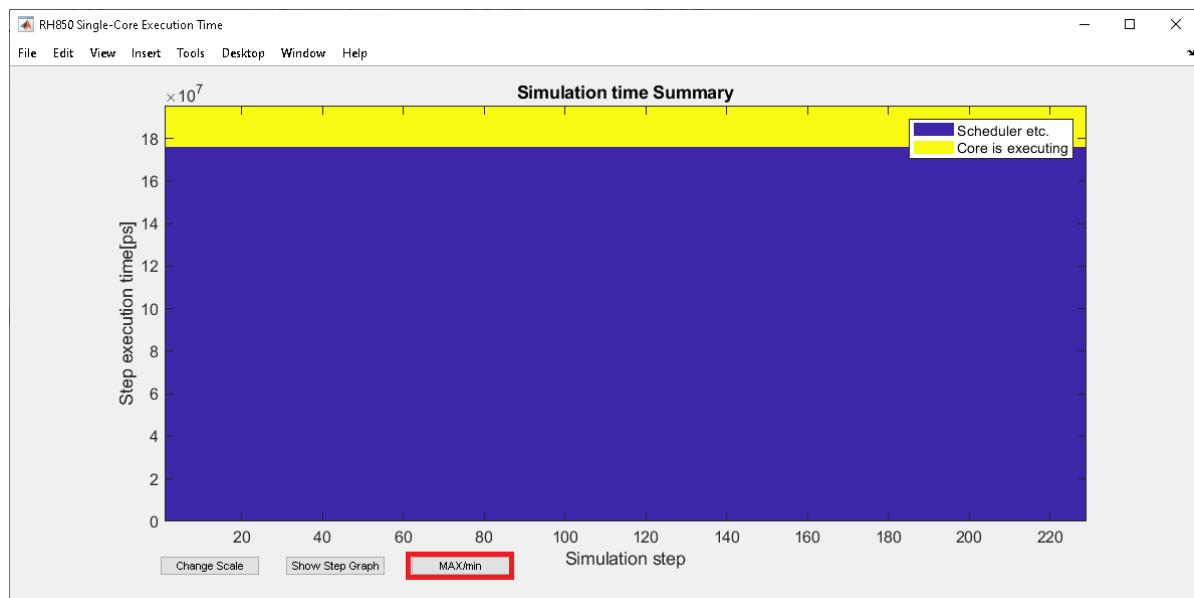


図 3-50 最悪または最小実行時間の表示

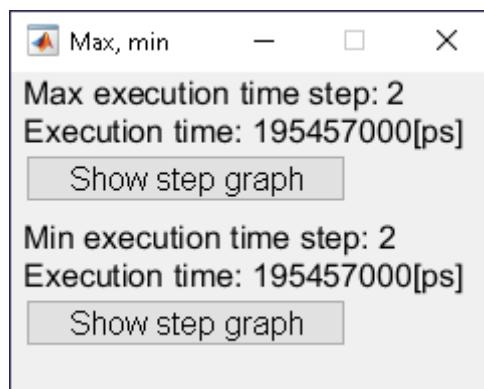


図 3-51 [Max, min] ダイアログ

それぞれの [Show step graph] ボタンのクリック後は、上記ステップ(1)と同様の操作を行ってください。最悪実行時間または最小実行時間となるステップの実行時間の詳細を表示します。

4 注意事項

本章では、ET-VPF を使用する上での注意事項について説明します。

4.1 仕様に関する注意事項

- (1) SC V1.5.0 では、RH850/F1KM-S4 の 272 ピンの品種はサポートされていないため、ご利用いただけません。
- (2) 時間計測対象ブロックの名前はユニークでなければなりません。
ET-VPF は時間計測対象に指定されたブロックに応じた関数名を使用しています。そのため異なる階層にある同じ名前のブロックを区別できません。
- (3) ET-VPF により生成された周辺コードのうち、以下の関数は未使用で、動作保証していません。
 - R_Config_<CAN Channel>_Transmission_Stop(): ET-VPF では CAN のチャネルストップモードへの移行は行わないため、この関数は使用しません。
 - CAN_Common_Reception_Init(): ET-VPF は CAN の送受信の初期化には CAN_Common_Init() を使うため、この関数は使用しません。
- (4) ADC 周辺ブロックでは、RH850/F1KM の場合、ユニット 0(ADCA0)、スキャン・グループ 1、マルチサイクル・スキャン・モードのみをサポートします。RH850/U2B の場合、スキャン・グループ 0、マルチサイクル・スキャン・モードのみをサポートします。
- (5) RESETOUT 端子が割り当てられているポートは Port 周辺ブロックで使用できないため、他のポートを選択してください。RH850/F1KM の場合は P8-6、RH850/U2B の場合は P27_0 が該当します。
- (6) MATLAB と VLAB 間の通信を同期させるには、サンプリング時間を各ステップのアルゴリズムの実行時間よりも長くする必要があります。各ステップの実行時間がサンプリング時間よりも長い場合、実行結果は保証されません。実行時間については、”<Code generation target>_etvpf” フォルダに生成される execution_data.csv ファイル内の” Time” 列を参照してください。
- (7) TAUD 周辺ブロックのデューティタイムおよびパルス周期はサンプリング時間より長くなければなりません。ET-VPF では PWM 出力信号はサンプリング時間毎に MATLAB に送信します。デューティタイムやパルス周期をサンプリング時間よりも短くすると、VPF から MATLAB への PWM 出力信号の出力時に、一部の信号が欠落することがあります。
- (8) VPF の PWM 出力信号が少し遅れます。VPF では、TAUD クロックは初期化時にいくらかの遅延があり、実際のデバイスと似た動作を行います(TAUD を起動した後、すぐに高い値になりますが理想的な動作ではありません)。回避策として、Unit Delay ブロック(初期値 0)を追加した MILS 用 PWM ブロックを用意しました。このブロックと TAUD 周辺ブロックの動作を比較すると、MILS と vHILS の波形は同じになります(動作が不定な最初のステップを除く)。
- (9) MILS 用 PWM ブロックは三角波 PWM 出力には対応していません。

(10) VLAB の ISS モデルには精度ありの CA モードがあります。以下の手順で使用可能ですが、動作保証はしてません。

手順 1: 一度 ET-VPF を実行し、実行環境を保存しておきます。

手順 2: ETVPF_testbench_sample.py ファイル中の以下の記述を修正します。

- vlab.read_register("RH850.CPUSS.PE1_FastIss.r31")
↓
vlab.read_register("RH850.CPUSS.PE1_CAIss.r31")
- vlab.load('rh850.f1km', ['--device-type=S1', '-t', './sim.py'])
↓
vlab.load('rh850.f1km', ['--device-type=S1', '-t', './sim.py', '-iss=ca'])

手順 3: 現在の Simulink モデルを手動で再実行します。

手順 4: ETVPF_testbench_sample.py を手動で再実行します。

(11) CS+、Cygwin、Smart Configurator、および VLAB のインストール・パスには、一部の特殊文字を含めることはできません（詳細は「表 3.7 使用可能な特殊文字」を参照してください）。

(12) VLAB ツール・ボックスの問題により、RH850/U2B の Port 周辺ブロックでは Port_Out 31、ADC 周辺ブロックでは、AN050、AN051 に対応していません。

(13) CAN Pack および CAN Unpack パラメーターについて、MATLAB 変数を使用した設定は ET-VPF ではサポートしていないため、即値を入力してください。

(14) サンプリング時間が短い場合、処理が間に合わず演算結果を期待通りに出力する事が場合があります。その場合は、サンプリング時間を長くしてください。

(15) 周辺ブロックのボーレートを高くまたは低く設定すると、通信時間に影響を与える場合があります。

4.2 Simulink モデルに関する注意事項

4.2.1 パス名やブロック名に使用できる文字列

Simulink モデルが保存されているパスや、コード生成対象のブロック名には、2 バイト文字（日本語等）、スペース、スラッシュ、改行、ハイフンは使わないでください。コード生成対象のブロック名に 2 バイト文字（日本語等）を使った場合は MATLAB が置き換えた文字列に、ハイフン、スペースを使った場合はそれ以降が省略された文字列になり、ET-VPF の一連の動作は可能ですが、動作は保証しません。

4.2.2 複素数データを扱うモデル

複素数データを扱う MATLAB/Simulink モデル(サブシステム)からのコード生成に対応していません。

4.2.3 周辺ブロックへの Constant ブロック入力

MATLAB/Simulink モデル(サブシステム)からのコード生成に定数出力が含まれる場合、周辺ブロックのソースが最適化されるため、MATLAB は予期しないソースを生成します。

この問題を回避するには、コード生成対象ブロック内にて、Constant ブロックの出力に、Repeating Sequence Stair ブロック(出力値のベクトルは 1 に設定)を乗算し、その出力を周辺ブロックに接続します。これにより、Constant ブロックの出力は、すべてのステップで同じ値で利用できるようになります。

4.3 実行環境構築時やシミュレーション中の注意事項

4.3.1 コード生成フォルダのパスの長さ

Windows のプラットフォームには、ファイル、フォルダへのパスの長さが最大 260 文字という制約があります。Simulink モデルによって生成されたソースコードを含むコード生成フォルダへのパスが長い場合、ビルドエラーになります。

4.3.2 電源管理

ET-VPF の作業途中で PC をスリープまたは休止状態にすると、復帰後の操作でエラーが発生する場合があります。

4.3.3 スクリプト・ファイル名の長さ

MATLAB にはスクリプト・ファイル名の長さの制限があります。ファイル名の長さが 64 文字以上の場合、スクリプト・ファイルは実行できず、エラーが発生します。

4.3.4 インストール・ドライブと作業ドライブ

ET-VPF をインストールしたドライブと、モデルファイルを格納する作業ドライブが異なる場合、エラーが発生する場合があります。

4.3.5 デフォルト設定での CS+プロジェクトのビルド

“Generate the CS+ project and automatically build” 選択時は、CS+のデフォルト設定で自動ビルドを行うため、ユーザによる設定変更は行えません。そのため、CS+のデフォルト設定がユースケースに適さない場合、エラーになる事があります。この問題を回避するには、“Generate the CS+ project only”を選択して CS+の必要な設定を手動で変更する、あるいは “None” を選択して makefile によりソースコードをコンパイル、ビルドしてください。

5 エラー・メッセージと対処方法

本章では、ET-VPF のエラー・メッセージとその対処方法を説明します。

5.1 概要

エラー・メッセージは、[コンフィギュレーション パラメーター] ダイアログで [ET-VPF options] オプションを設定している際、または ET-VPF が vHILS 実行中に“ユーザに通知すべき情報”を検出した際に表示されます。

補足 ET-VPF が出力するエラー・メッセージは、Windows と連携していません。このため、ET-VPF がエラー・メッセージを出力した際、F1 キーを押下しても該当ヘルプは起動しません。

5.2 コンフィギュレーション パラメーターのエラー

以下に、[コンフィギュレーション パラメーター]ダイアログで検出されるエラー・メッセージの一覧を示します。これらのエラー・メッセージは、ET-VPF エラー・ダイアログに出力します。

表 5.1 コンフィギュレーション パラメーターにおけるエラー・メッセージ

[メッセージ]	E0101 The cygwin directory setting is incorrect.
[説明]	[Cygwin Installation Directory]に正しいインストール先の情報が設定されていない場合に表示されます。
[対処方法]	1. Cygwin の正しいインストール先のパスを [Cygwin Installation Directory] に設定してください。bin/bash.exe があるフォルダを指定します。 2. Cygwin が正常にインストールされていることを確認してください。
[メッセージ]	E0102 The VLAB directory setting is incorrect.
[説明]	[VLAB Installation Directory]に正しいインストール先の情報が設定されていない場合に表示されます。
[対処方法]	1. VLAB の正しいインストール先のパスを [VLAB Installation Directory] に設定してください。vlab-ide.exe があるフォルダを指定します。 2. VLAB が正常にインストールされていることを確認してください。
[メッセージ]	E0103 The CS+ directory setting is incorrect.
[説明]	[CS+ Installation Directory]に正しいインストール先の情報が設定されていない場合に表示されます。
[対処方法]	1. CS+の正しいインストール先のパスを [CS+ Install Directory] に設定してください。CubeSuiteW+.exe があるフォルダを指定します。 2. CS+が正常にインストールされていることを確認してください。
[メッセージ]	E0104 The Smart Configurator directory setting is incorrect.
[説明]	[SC Installation Directory]に正しいインストール先の情報が設定されていない場合に表示されます。
[対処方法]	1. SC の正しいインストール先のパスを [SC Installation Directory] に設定してください。SmartConfigurator.exe があるフォルダを指定します。 2. SC が正常にインストールされていることを確認してください。

表 5.1 コンフィギュレーション パラメーターにおけるエラー・メッセージ（続き）

[メッセージ]	E0106 The default directory is selected as the current installation directory. Deselect the use of the default directory.
[説明]	[Use default <ツール名> Install Directory] チェックボックスにチェックが付いている状態で、[Select <ツール名> Install Directory] ボタンを押下した場合に表示されるエラーです。
[対処方法]	[Use default <ツール名> Install Directory] チェックボックスのチェックを外した状態で、[Select <ツール名> Install Directory] ボタンを押下してください。
[メッセージ]	E0107 The parameter <paramName> does not exist.
[説明]	パラメーター <paramName> が ET-VPF Options にない場合に表示されます。
[対処方法]	システム ターゲット ファイル が etvpf.tlc に設定されているかどうかを確認してください。 MATLAB パスに ET-VPF インストール・フォルダのパスが設定されているか確認してください
[メッセージ]	E0110 A license is not registered.
[説明]	利用可能なライセンスがない場合に表示されるエラーです。
[対処方法]	1. ライセンスがない場合、ルネサスへお問合せください。 2. ライセンスをお持ちの場合、ルネサス製ライセンス・マネージャにてライセンス追加を行ってください。 3. ライセンスを確認するため、[ET-VPF options] パネルの [Check Available Features] ボタンを押下してください。
[メッセージ]	E0111 <Device Series> is not available. Register a valid license.
[説明]	<Device Series> のライセンスが無効である場合に表示されるエラーです。
[対処方法]	1. ライセンスがない場合、ルネサスへお問合せください。 2. ライセンスをお持ちの場合、ルネサス製ライセンス・マネージャにてライセンス追加を行ってください。 3. ライセンスを確認するため、[ET-VPF options] パネルの [Check Available Features] ボタンを押下してください。

5.3 vHILS 中のエラー

vHILS 処理で検出されるエラー・メッセージについて説明します。vHILS 処理中のエラー・ダイアログは、MATLAB/Simulink から出力されます。

表 5.2 実行時間取得時のエラー・メッセージ

[メッセージ]	E0201 Wrong measurement data. Please check data input files.
[説明]	実行時間測定結果に誤りがあるか空です。
[対処方法]	<ol style="list-style-type: none">モデルのコンフィグレーション、SC のコンポーネントのコンフィグレーションが正しいか確認してください。上記が正しく設定されている場合は、vHILS を再度実行して実行時間測定結果ファイルを生成してください。

5.4 CS+起動時、CS+でのビルド中のエラー

CS+起動時、CS+でのビルド中に検出されるエラー・メッセージについて説明します。

表 5.3 CS+にて表示されるエラー・メッセージ

[メッセージ]	E0301 Creating the CS+ project was not possible (project.Create error).
[説明]	CS+プロジェクト・ファイルが生成できませんでした。
[対処方法]	<ol style="list-style-type: none"> ET-VPF がサポートしているCS+バージョンであることを確認してください。 CS+ Python プラグインが有効になっていることを確認してください。
[メッセージ]	E0302 Adding the source file was not possible (project.File.Add error).
[説明]	CS+プロジェクト・ファイルにソースを登録できませんでした。
[対処方法]	ET-VPFがサポートしているMATLABバージョンであることを確認してください。
[メッセージ]	E0303 Removing the source file was not possible (project.File.Remove error).
[説明]	CS+プロジェクト・ファイルからソースを削除できませんでした。
[対処方法]	<ol style="list-style-type: none"> CS+プロジェクト・ファイルの削除権限があることを確認してください。 CS+プロジェクト・ファイルが存在していることを確認してください。
[メッセージ]	E0304 Setting the link options was not possible (link_option.sec error).
[説明]	CS+プロジェクトでセクション開始アドレスを設定するための値が誤っている場合エラーが発生します。
[対処方法]	<p>以下の手順を行ってください。</p> <ol style="list-style-type: none"> 値と構文を見直し、ET-VPF インストール・フォルダにある <code>renesas_link_RH850_option_single.sec</code> ファイルを修正してください。 検証環境を再度生成してください。 デバイスのメモリサイズが小さい場合は、メモリサイズの大きいデバイスの利用を検討してください。
[メッセージ]	E0305 Building was not possible (build.All error).
[説明]	ビルト時にエラーが発生しました。
[対処方法]	<p>以下の確認を行い、検証環境の作成からやり直してください。</p> <ol style="list-style-type: none"> CS+プロパティ設定を見直してください。 CS+出力パネルに表示されているエラー・メッセージを確認してください。 デバイスのメモリサイズが小さい場合は、メモリサイズの大きいデバイスの利用を検討してください。

5.5 vHILS 環境構築時のエラー

vHILS 環境構築時に検出されるエラー・メッセージについて説明します。

表 5.4 vHILS 環境構築時に表示されるエラー・メッセージ

[メッセージ]	E0310 The <Full File Name> file is not existed. Please check this file.
[説明]	入力ファイル <Full File Name> が見つからない場合に表示されるエラーです。
[対処方法]	<ol style="list-style-type: none"> 1. 入力ファイルがあることを確認してください。 2. 入力ファイルのフルパスが正しいことを確認してください。

5.6 MATLAB と VLAB 間の通信中のエラー

MATLAB と VLAB 間の通信中に検出されるエラー・メッセージについて説明します。

表 5.5 MATLAB と VLAB 間の通信中に表示されるエラー・メッセージ

[メッセージ]	E0401 An incorrect automation server is registered.
[説明]	MATLAB の異なるバージョンがオートメーション・サーバとして登録された場合に表示されるエラーです。
[対処方法]	regmatlabserver コマンドを使用して、使用中の MATLAB をオートメーション・サーバとして登録してください。

索引

A

Atomic サブシステム 47, 48

G

Graph Viewer 49

M

MILS 6, 30

S

Simulink 7

V

vHILS 6

vHILS 環境生成 38

VLAB 8

VPF 6

あ

アンインストール 14

い

インストール 12

インストール・パス 8, 37, 55

え

エラー・メッセージ 58

こ

コンフィギュレーション パラメーター 32

さ

最悪実行時間表示 53

最小実行時間表示 53

し

時間計測方法 47

周辺コード 6

周辺コード生成 39

周辺ブロック 16

ち

注意事項 54

と

動作環境 7

ら

ライセンス 9

改訂記録	Embedded Target for RH850 Virtual Platform V1.00.00 RH850 モデルベース開発環境 ユーザーズマニュアル
------	--

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2022.10.21	—	初版発行
1.01	2023.3.20	9	インストール・フォルダに関する注意事項の追加
		41	input_subsystem.txt のフォーマットの修正
		7, 32, 47, 48	注意事項の追加と更新
		—	サポートしていないデバイス関連の記述の削除
1.02	2024.1.16	—	RH850/U2B 関連の記述の追加

Embedded Target for RH850 Virtual Platform V1.00.00
RH850 モデルベース開発環境
ユーザーズマニュアル

発行年月日 2024 年 1 月 16 日 Rev. 1.02

発行 ルネサス エレクトロニクス株式会社
〒135-0061 東京都江東区豊洲 3-2-24 (豊洲フォレシア)

Embedded Target for RH850 Virtual Platform
V1.00.00

RH850 Model-Based Development Tool

