renesas_an_blue–

User’s Manual

Target Device

RH850 Family

Embedded Target for RH850 Virtual Platform

V1.00.00

RH850 Model-Based Development Tool

User’s Manual

**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。**

Rev.1.00 2014.04

All information contained in these materials, including products and product specifications,

represents information on the product at the time of publication and is subject to change by

Renesas Electronics Corp. without notice. Please review the latest information published by

Renesas Electronics Corp. through various means, including the Renesas Electronics Corp.

website (http://www.renesas.com).

Rev.1.00 October 2022

Renesas Electronics

<http://www.renesas.com>

**Notice**

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.

5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depend on the product’s quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.

6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

|  |  |  |
| --- | --- | --- |
| **Corporate Headquarters** |  | **Contact information** |
| TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan  [www.renesas.com](https://www.renesas.com/) |  | For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: [www.renesas.com/contact/](https://www.renesas.com/contact/). |
| **Trademarks** |  |  |
| Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners. |  |  |

**How to Use This Manual**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Readers |  | This manual is intended for users who wish to understand the functions of the MATLAB/Simulink and use them to develop application systems. | | |
| Purpose |  | This manual is intended to educate users about the functions of the model-based development tools, Embedded Target for RH850 Virtual Platform, to be used as reference for developing software systems. | | |
| Composition |  | The contents of this manual are as follows:  Chapter 1 Overview  Chapter 2 Installation  Chapter 3 Functional Operation Procedure  Chapter 4 Points for Caution  Chapter 5 Error Messages | | |
| How to Read This Manual |  | Readers of this manual are assumed to have general knowledge of electricity, logic circuits, and microcontrollers. | | |
| Conventions |  | Note:  Caution:  Remark:  Numeric Representation: |  | Footnote for items marked with Note in the text  Information requiring attention  Supplementary information  Decimal ... XXXX  Hexadecimal ... XXXXH or 0xXXXX | |
| Related Documents |  |  | | |

All trademarks or registered trademarks in this document are the property of their respective owners.

Microsoft and Windows are trademarks or registered trademarks of Microsoft Corporation.

MathWorks, MATLAB, MATLAB Coder, Simulink, Simulink Coder, Embedded Coder, and Stateflow are trademarks or registered trademarks of The MathWorks, Inc.

table of contents

[**1** **overview** 5](#_Toc122608728)

[1.1 features 7](#_Toc122608729)

[1.2 operating environment 7](#_Toc122608730)

[1.3 license types and functions 9](#_Toc122608731)

[**2** **installation** 10](#_Toc122608732)

[2.1 installation 12](#_Toc122608733)

[2.2 uninstallation 14](#_Toc122608734)

[**3** **functional operation procedure** 16](#_Toc122608735)

[3.1 overview 16](#_Toc122608736)

[3.2 s-function block of peripherals 18](#_Toc122608737)

[**3.2.1 adc peripheral** 18](#_Toc122608738)

[**3.2.2 port peripheral** 20](#_Toc122608739)

[**3.2.3 rs-canfd peripheral** 22](#_Toc122608740)

[**3.2.4. rlin3n peripheral** 26](#_Toc122608741)

[**3.2.5. taud peripheral** 27](#_Toc122608742)

[3.3 executing simulator processor in the loop simulation 30](#_Toc122608743)

[**3.3.1 embedded sample model** 30](#_Toc122608744)

[**3.3.2 setting configuration parameters** 32](#_Toc122608745)

[**3.3.3 generating a vhils environment** 37](#_Toc122608746)

[**3.3.4 compiling generated source code** 43](#_Toc122608747)

[**3.3.5 executing vhils** 44](#_Toc122608748)

[3.4 time measurement 46](#_Toc122608749)

[**3.4.1 structure of simulink model for measurement** 46](#_Toc122608750)

[**3.4.2 input file for measurement** 47](#_Toc122608751)

[**3.4.3 how to change normal subsystem to atomic subsystem** 47](#_Toc122608752)

[**3.4.4 graph viewer** 48](#_Toc122608753)

[**4** **points for caution** 53](#_Toc122608754)

[4.1 features 53](#_Toc122608755)

[4.2 simulink models 56](#_Toc122608756)

[**4.2.1 available strings for paths and block names** 56](#_Toc122608757)

[**4.2.2 models handling complex number data** 56](#_Toc122608758)

[4.3 construction and simulation 56](#_Toc122608759)

[**4.3.1 length of path to code generation folder** 56](#_Toc122608760)

[**4.3.2 notes on power management** 56](#_Toc122608761)

[**4.3.3 length of script file name** 56](#_Toc122608762)

[**4.3.4 install drive and work drive** 56](#_Toc122608763)

[**5** **error messages** 57](#_Toc122608764)

[5.1 overview 57](#_Toc122608765)

[5.2 errors detected in configuration parameters dialog boxes 58](#_Toc122608766)

[5.3 errors during vhils execution 60](#_Toc122608767)

[index 61](#_Toc122608768)

[revision history 62](#_Toc122608769)

# 

# **Overview**

This section provides an overview of functions of Embedded Target for RH850 Virtual Platform (hereafter referrerd to as ET-VPF).

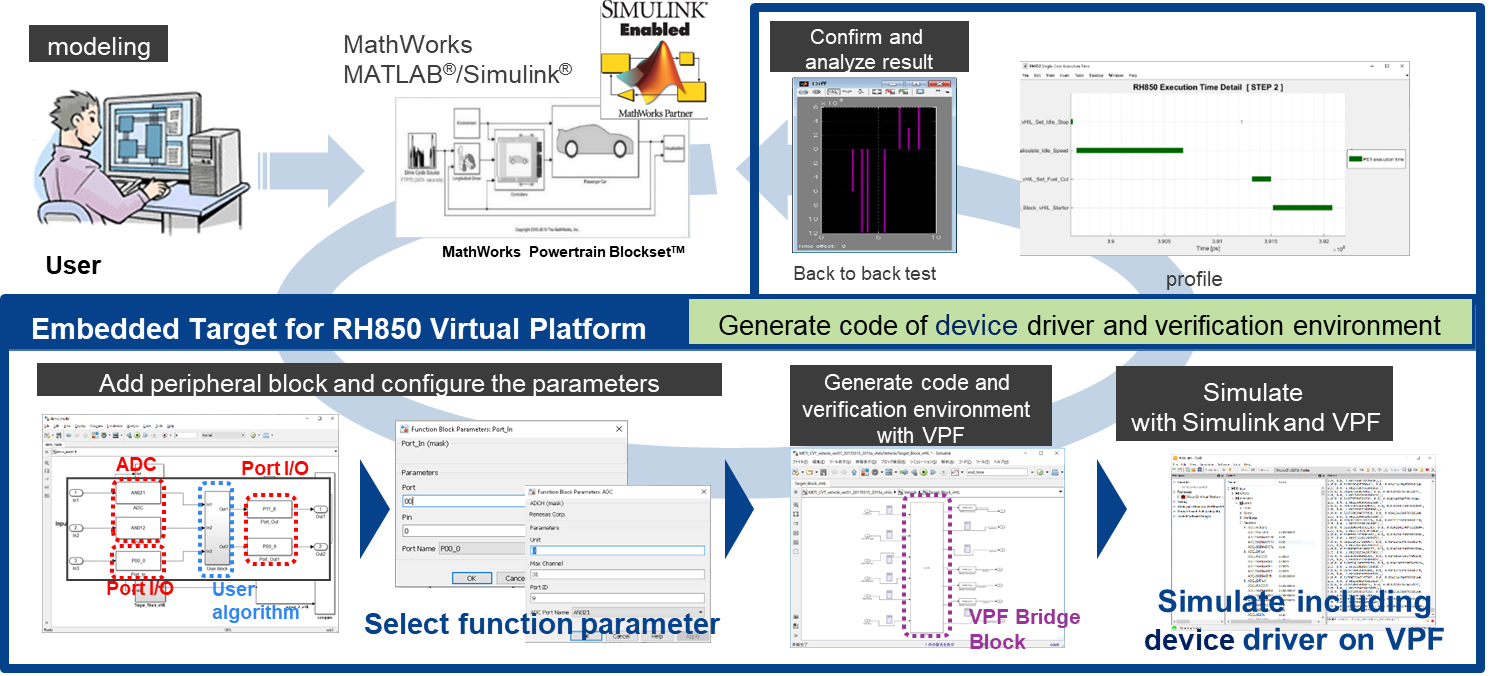
FMILS

At the stage of examining algorithms in Model in the Loop Simulation (MILS) used in model-based development, there is no environment to measure the performance of the target device including its peripheral functions.

To estimate the processing time of the peripheral functions in an actual device, it is necessary to combine the code generated from the model with the code for controlling the peripheral functions (hereafter referrerd to as peripherals’ source code). This is a lot of work for users who do not know the details of the peripheral functions.

ET-VPF generates code for the user algorithm part and generates peripheral code to be implemented on the target from the Simulink model, then runs target code on a partner's RH850 virtual platform (hereafter referred to as VPF) and performs a linked simulation with a Simulink plant model. Renesas call this virtual Hardware in the Loop Simulation(vHILS).

This enables comparative verification of MILS and vHILS (back-to-back test) and performance verification in the early stages of development.



***Figure 1‑1 Overview of Embedded Target for RH850 Virtual Platform***

## 1.1 Features

Show below are the features of ET-VPF:

* Automatic construction of vHILSimulation verification environment
  + In vHILSimulation in which MATLAB/Simulink, CS+ and VLAB are interlinked, a load module generated from Simulink models can be executed on a target device.
  + The following devices are supported.

***Table 1‑1 Supported devices***

|  |  |
| --- | --- |
| **Series** | **Devices** |
| RH850/F1x | RH850/F1KM-S1, RH850/F1KM-S4 |
| RH850/U2x | RH850/U2C |

* Graphical display of the execution state in Subsystem blocks of Simulink models
* Displaying the execution time for each subsystem during simulation
* A processing margin of the control cycle for the worst -case execution time during simulation time can be checked.

## 1.2 Operating Environment

The descriptions below are the system requirements for ET-VPF.

* + Hardware environment
* Operating system: Microsoft Windows® 10 (64-bit)
* Processor: 1 GHz or higher (supporting hyper-threading or Multi-Core CPU)
* Main memory: 4 GB or more is recommended
  + Software environment
* MATLAB and Simulink products (from The MathWorks, Inc.)

Windows 10

MATLAB V9.3 (R2017b)

Simulink V9.3 (R2017b)

Stateflow V9.0 (R2017b)

MATLAB Coder V9.0 (R2017b)

Simulink Coder V8.13 (R2017b)

Embedded Coder V6.13 (R2017b)

Vehicle Network Toolbox V3.4 (R2017b)

* MEX-file compiler  
  The MEX-file is the interface that invokes C library from MATLAB. The MEX-file compiler (used to compile MEX files) can be checked on the operating environment page provided by MathWorks. ET-VPF has been tested with the following compilers as the MEX file compiler.

Microsoft Visual C++ 2013, 2015, 2017 compiler (from Microsoft Corporation)

Reference: System Requirements & Platform Availability

<https://www.mathworks.com/support/sysreq/previous_releases.html>

* VLAB (from The VLAB Works Pty Ltd)

VLAB V2.7.2 (win-vc140-x64)

* Toolboxes of VLAB

Toolbox RH850 Virtual Platform V3.1.8 (For test supported devices (RH850/F1M-S1, RH850/F1M-S4))

Toolbox CAN 2.5.0 (For test supported peripheral RS-CANFD)

Toolbox RH850 G4 Virtual Platform 1.16.0 (For test supported devices RH850/U2C)

* Smart Configurator for RH850 (from Renesas Electronics)

SC V1.5.0

* Build Tool

CC-RH Included with CS+ V8.07.00/ E8.07.00j2 (from Renesas Electronics)

* Cygwin (from Cygwin Authors)

Cygwin V2.11.2

Remarks 1. For the MATLAB and Simulink products, an environment is constructed by using option products corresponding to the versions of MATLAB and Simulink being used.

2. When installing MATLAB, it is recommended that the installation folder is changed to other than the folder for UAC (user account control). Depending on the version of MATLAB in use, if the installation folder is the folder for UAC such as “<system drive>:\Program Files” or “<system drive>:\Program Files (x86)”, a problem such that MEX cannot be built, or the MATLAB path cannot be saved may occur.

3. MATLAB R2017b version is used because when preparing the prototype of vHILS, the VLAB Co-Simulink support R2017b version. For other versions, they have not been tested.

4. Can use the upper software versions if the features used for ET-VPF be not changed but Renesas Electronics does not guarantee it.

5. For CS+ E8.07.00j2, it is only used for RH850/U2C device series. For CS+ V8.07.00, it is only used for RH850/F1KM-S1, RH850/F1KM-S4 device series.

6. The Smart Configurator for RH850 is unavailable for RH850/U2C device series.

7. The installation path of ET-VPF package, CS+, Cygwin, Smart Configurator and VLAB cannot contain some special characters (refer to ***Table 3‑7 The supported special characters*** for more detail).

## 1.3 License Types and Functions

This section explains the license types.

ET-VPF offers various features to verify algorithm of embedded models. Some features require specific license, which was registered with Renesas Electronics. This section describes use cases of these features.

Below table show available operation when you own each license type.

***Table 1‑2 License definition of ET-VPF***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **License name** | **Supported device** | | | **Time measurement feature** | **Supported build tool** | |
| **RH850/F1KM-S1** | **RH850/F1KM-S4** | **RH850/U2C** | **GHS Compiler** | **Renesas Compiler** |
| Embedded Target for RH850 Virtual Platform | O | O | O | O | - | O |

*\** O *: Can execute  
 -: Cannot execute*

# 

# **Installation**

Start the installer, specify the ET-VPF installation folder, and then execute the installation. For the ET-VPF installation folder, a folder that is targeted for User Account Control (UAC), such as “<system drive>:\Program Files” or “<system drive>:\Program Files (x86)”, cannot be specified.

Remark If installed in a folder targeted for UAC, the ET-VPF cannot be used because, for example, MATLAB path settings cannot be saved.

By extracting the zipped package, the programs, samples, and libraries required for communication between MATLAB and VLAB (hereafter called “MATLAB communication library”) are stored in the following structures.

The following describes the structure of ET-VPF package after installed successfully.

1. Folder structure under the ET-VPF package folder for RH850/F1x device series.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
| <ET-VPF installation folder>\<version information>\F1x\  ETVPF\_package\ | ETVPF\_include\ETVPF\_S\_function\_block\ | … | Contains the source code which used to execute the program of peripherals. |
|  | ETVPF\_include\make\  ETVPF\_include\Renesas\  ETVPF\_include\XML\_input\  Source code files | …  …  …  … | Contains the template of make file.  Contains the source code of each device which used for Renesas compiler.  ***Note:*** User can modify boot code files in “startup” folders but Renesas Electronics does not take responsibility for quality.  Contains the XML input files which provided for each RH850 device series.  Contains the list of source code files (include: \*.p, \*.tlc, make files) which used to execute the program of ET-VPF. | |

1. Folder structure under the ET-VPF package folder for RH850/U2x device series.

|  |  |  |  |
| --- | --- | --- | --- |
| <ET-VPF installation folder>\<version information>\U2x\  ETVPF\_package\ | ETVPF\_include\ETVPF\_S\_function\_block\ | … | Contains the source code which used to execute the program of peripherals. |
|  | ETVPF\_include\make\  ETVPF\_include\Renesas\  Source code files | …  …  … | Contains the template of make file.  Contains the source code of each device which used for Renesas compiler.  ***Note:*** User can modify boot code files in “startup” folders but Renesas Electronics does not take responsibility for quality.  Contains the list of source code files (include: \*.p, \*.tlc, make files) which used to execute the program of ET-VPF. | |

Remarks 1. The current <version information> is V1.00.00.

2. For <ET-VPF installation folder>, the user can change them when installing (the user needs to make sure that the installation locations are not limited permission). And the <ET-VPF installation folder> must be in the same <system drive> with the user’s workspace to avoid the error occurring when loading the loaded module to RH850 virtual platform.

3. Currently, ET-VPF V1.00.00 installer only support to install ET-VPF package of RH850/F1x device series.

4. To avoid the warning when downloading the setup file from the website, a zip file for ET-VPF installer (ETVPF\_<version information>\_Setup.7z) will be prepared.

5. The <ET-VPF installation folder> must not contain the special characters (refer to ***Table 3‑7 The supported special characters***). An error will occur when executing ET-VPF if <ET-VPF installation folder> has the special characters.

## 2.1 Installation

The following describes the procedure for installing ET-VPF.

1. Install the ET-VPF package by the below steps:

* **Step 1:** Extract ETVPF\_<version information>\_Setup.7z file.
* **Step 2:** Double click to ETVPF\_<version information>\_Setup.exe to start installing.
* **Step 3:** Execute step by step to install ET-VPF package as the following figure.

Graphical user interface, application, Teams

Description automatically generated

**Figure 2‑1 Installation Procedure**

1. After installed the ET-VPF package, user must set path for ET-VPF package. Start MATLAB, and then add the folder of ET-VPF on the [Set Path] dialog box.

There are two necessary paths which need to set to MATLAB:

* + **ET-VPF include:** …\ETVPF\_package\ETVPF\_include
  + **ET-VPF S-Function:** …\ETVPF\_package\ETVPF\_include\ETVPF\_S\_function\_block

Graphical user interface, text, application

Description automatically generated

**Figure 2‑2 Adding the ET-VPF folder**

1. Register MATLAB Automation Server.

Execute the following command from the MATLAB command window to specify the current in-use MATLAB version as the Automation Server.

Here ">>" denotes the command prompt and "[Enter]" denotes entry of the Enter key.

>> regmatlabserver [Enter]

Remarks 1. Open MATLAB under administration privilege when executing this command.

2. If you change the in-use MATLAB version, execute this command again.

1. Register the license.

Add the ET-VPF license by license manager included with CS+ (refer License Manager [V2.05.00 User's manual](https://www.renesas.com/document/mat/license-manager-v20500-users-manual)).

1. Setup the MATLAB Simulink Library Browser

Execute the following command from the MATLAB command window to open ETVPF package on the MATLAB Simulink Library Browser to choose the expected S-function blocks of peripherals.

Here “>>” denotes the command prompt and “[Enter]” denotes entry of the Enter key.

>> setup\_etvpf\_lib [Enter]

## 2.2 Uninstallation

The following describes the procedure for uninstalling ET-VPF.

1. Start MATLAB, and then remove the folder of ET-VPF (include both two paths described in section **2.1 Installation**) on the [Set Path] dialog box.

Graphical user interface, text, application

Description automatically generated

**Figure 2‑3 Removing the ET-VPF folder**

1. Execute Uninst\_ETVPF\_<version information>.exe in the ET-VPF installation folder as below steps:

* **Step 1:** Double click to Uninst\_ETVPF\_<version information>.exe to start uninstalling ET-VPF package.
* **Step 2:** Execute step by step to uninstall ET-VPF package as the following figures.

Graphical user interface, text, application

Description automatically generated

**Figure 2‑4 Uninstallation Confirmation Message**

A screenshot of a computer

Description automatically generated

**Figure 2‑5 Uninstallation Procedure**

Remark If (2) is performed prior to (1) above, a warning will be displayed the next time MATLAB is started.

1. Delete the license.  
    Delete the ET-VPF license by license manager included with CS+.
2. Uninstall the ET-VPF category in Simulink Library Browser window as below step:

Execute the following command from the MATLAB command window to remove ETVPF category in the MATLAB Simulink Library Browser window.

Here “>>” denotes the command prompt and “[Enter]” denotes entry of the Enter key.

>> refresh(LibraryBrowser.LibraryBrowser2) [Enter]

Remark If (4) is performed prior to (1) above, a warning will be displayed the next time MATLAB is started.

# 

# **Functional Operation Procedure**

This chapter describes the functions provided by ET-VPF.

## 3.1 Overview

ET-VPF provides functions to generate a vHILS environment and to verify algorithms. ET-VPF generates a vHILS environment in cooperation with Embedded Coder.

The following describes the main procedure of ET-VPF:

* Set configuration parameters
* Clean existed files, objects.
* Generate source code files.
* Generate C source files of peripherals (Port, ADC, RLIN3n, TAUD) (using SC to generate them).
* Generate source files of target device.
* Generate C source files.
* Generate make file and config file.
* Generate application make file.
* Generate VLAB execution file.
* Generate python files.
* Generate Define.h, OSTM\_define.h.
* Call Cygwin to build source code.
* Start-up VLAB then run the target python script.
* Wait until the co-sim finish and verify the result.

About S-Function block of peripherals (hereafter referrerd to as peripheral block), there are Port, ADC, RS-CANFD, RLIN3n, TAUD blocks included in “etvpf\_lib.slx” file.

ET-VPF will support generate a vHILS environment and verify algorithms automatically. But the settings of peripherals are dependent on each use case. Therefore, ET-VPF also support user do it by manual: during generation of vHILS environment, user can configure settings, generate source code for peripherals (Port, ADC, RLIN3n, TAUD) easier, more detail via Smart Configurator (hereafter referrerd to as SC). To do this step, user can refer to section **3.3.3.1 Generating the peripherals’ source code by SC**.

The following table shows the list of the blocks that can be placed in the same layer as the measurement target block (hereafter referrerd to as usable blocks).

***Table 3‑1 List of usable blocks***

|  |  |  |
| --- | --- | --- |
| **No** | **Usable blocks** | **Configurable blocks** |
| 1 | Subsystem | Graphical user interface, text, application  Description automatically generated with medium confidence |
| 2 | Inport (\*1) | Diagram  Description automatically generated |
| 3 | Outport | Diagram  Description automatically generated |
| 4 | Mux | Icon  Description automatically generated |
| 5 | Demux | A black letter on a white background  Description automatically generated with low confidence |
| 6 | Data Type Conversion | Text  Description automatically generated with medium confidence |
| 7 | ADC | A picture containing text  Description automatically generated |
| 8 | PORT | A picture containing table  Description automatically generatedA picture containing shape  Description automatically generated |
| 9 | TAUD | Text, whiteboard  Description automatically generated |
| 10 | RLIN3 | Diagram  Description automatically generated with medium confidenceDiagram  Description automatically generated with medium confidence |
| 11 | RS-CANFD | Text  Description automatically generatedText  Description automatically generated with medium confidence |
| 12 | Chart | Text, icon  Description automatically generated |

(\*1) This block can be created by right clicking and then dragging the original Inport block and then selecting “Duplicate Inport” as the Inport shadow. This causes confusion and Renesas Electronics does not recommend user to use the Inport shadow block.

## 3.2 S-Function block of peripherals

The following describes about the S-Function blocks of peripherals (Port, ADC, RS-CANFD, RLIN3n, TAUD), how to create, setting and generate C code for these S-Function blocks.

In addition, the MATLAB Simulink Library Browser supports ETVPF package, which contained the S-function blocks of peripherals to add into the model.

Remarks 1. The S-Function blocks of peripherals are different with each device series. Therefore, user must select the correct peripherals’ settings with the current device series (peripherals’ settings on model must be the same with the settings on SC – refer to section **3.3.3.1 Generating the peripherals’ source code by SC** to know the configuration method). The sample S-Function blocks of each device series also include inside the “etvpf\_lib.slx” file.

2. About the peripherals’ settings used for RH850/F1KM, only use the following settings.

1. For ADC peripheral: support for Scan Group 1, Multicycle scan mode.
2. For Port peripheral: Only can use normal port (PORT) (e.g., PORT0, PORT1… defined in SC).
3. For RS-CANFD peripheral:

For F1KM: Only can use CANFD unit 0 (RCFDC0), Sending/receiving Data frame/Remote frame, Standard ID/Expansion ID, Reception filter function, Support CAN Port I/O and Vehicle Network Toolbox.

For U2C: Only can use CANFD unit 0 (RCFDC0), Sending/receiving Data frame/Remote frame, Standard ID/Expansion ID, Reception filter function, and Vehicle Network Toolbox.

1. For TAUD peripheral: Only can use PWM Output and Triangle PWM Output, master channel, pulse cycle and duty ratio value settings. About slave channels, and clock setting please set it in SC

3. The S-Function blocks of peripherals are prepared for the largest device. If using the smaller device, please select the function, port name which is implemented for this device.

4. About the peripherals used for RH850/U2C, only use the Port, ADC and CAN peripherals.

### **3.2.1 ADC peripheral**

The following describes about the main features of ADC peripheral.

* Get the data from MATLAB to ADC port on model.
* Return the same value from MATLAB to user algorithms.

A picture containing text

Description automatically generated

***Figure 3‑1 S-Function block of ADC***

The following describes about the User Interface of ADC S-Function block that support user select and change ADC ports easier during model setting.

Graphical user interface, text, application, email

Description automatically generated

***Figure 3‑2 User Interface of ADC S-Function block***

The User Interface of ADC S-Function block includes “ADC Port Name”, “Unit”, “Max Channel” and “Port ID” parameters.

The purpose of these parameters is specified the target port name of ADC S-Function block. When building the model, these parameters will generate at the same time with the input data.

After changing ADC port name, the parameters (Unit, Max Channel, Port ID) and displaying name of S-Function block will be changed automatically based on the current ADC port name.

Graphical user interface, text, application

Description automatically generated

***Figure 3‑3 Changing port name of ADC S-Function block***

Graphical user interface, text, application, Teams

Description automatically generated

***Figure 3‑4 The changing result of ADC S-Function block***

### **3.2.2 Port peripheral**

The following describes about the main features of Port peripheral.

* **For Port\_In block:**
* Get data from MATLAB and send to Port in VLAB.
* Send the received value to the user algorithms.
* **For Port\_Out block:**
* Set the value from user algorithms to the target port name.
* Send the value of port to the MATLAB (output of the MATLAB Co-simulation).

A picture containing table

Description automatically generated A picture containing table

Description automatically generated

***Figure 3‑5 S-Function block of Port***

The following describes about the User Interface of Port S-Function block that support user select and change ports easier during model setting. The User Interface also have two types be corresponding to two types of Port S-Function block (Port\_In and Port\_Out).

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

***Figure 3‑6 User Interface of Port S-Function block***

The User Interface of Port S-Function block includes “Port Name”, “Port” and “Pin” parameters.

The purpose of these parameters is specified the target port name of Port S-Function block. When building the model, these parameters will generate at the same time with the input data.

After changing port name, the parameters (Port, Pin) and displaying name of S-Function block will be changed automatically based on the current port name.

A picture containing graphical user interface

Description automatically generated

***Figure 3‑7 Changing port name of Port S-Function block***

Graphical user interface, text, application, chat or text message, Teams

Description automatically generated

***Figure 3‑8 The changing result of Port S-Function block***

### **3.2.3 RS-CANFD peripheral**

The following describes about the main features of RS-CANFD peripheral.

* **For CAN\_Transmission block:**
* Get array bytes (dynamic length) from User algorithm, transmit message through RS-CANFD port, then output to MATLAB (bytes array data).
* **For CAN\_Reception block:**
* Get array bytes (dynamic length) from MATLAB, transmit through RS-CANFD port, output to user algorithm (bytes array data).

Table

Description automatically generatedTable

Description automatically generated

**Figure 3‑9 S-Function block of RS-CANFD**

The following describes about the User Interface of RS-CANFD S-Function block that support user select and change RS-CANFD ports easier during model setting. The User Interface also have two types be corresponding to two RS-CANFD S-Function blocks (CAN\_Transmission and CAN\_Reception).

Graphical user interface, application

Description automatically generated

**Figure 3‑10 User Interface of RS-CANFD S-Function block**

The User Interface of RS-CANFD S-Function block includes “RS-CANFD unit”, “RS-CANFD channel”, “Port Name” parameters. The “RS-CANFD unit” and “RS-CANFD channel” will be generate automatically after changing the “Port Name”.

The purpose of these parameters is specified the target port name of RS-CANFD S-Function block. When building the model, these parameters will generate at the same time with the input data.

- For the message ID: ET-VPF prepared the CAN ID array (with the default ID message value as CAN unit number) for each CAN unit. Users can change it as the demand. ET-VPF support get the ID and extension ID from the CAN Pack and CAN Unpack block for each unit and set to the CAN ID array when generating.

Graphical user interface, application

Description automatically generatedGraphical user interface, application, email

Description automatically generated

**Figure 3‑11 Information from CAN Pack and CAN Unpack**

- Model structure between MATLAB and VLAB:

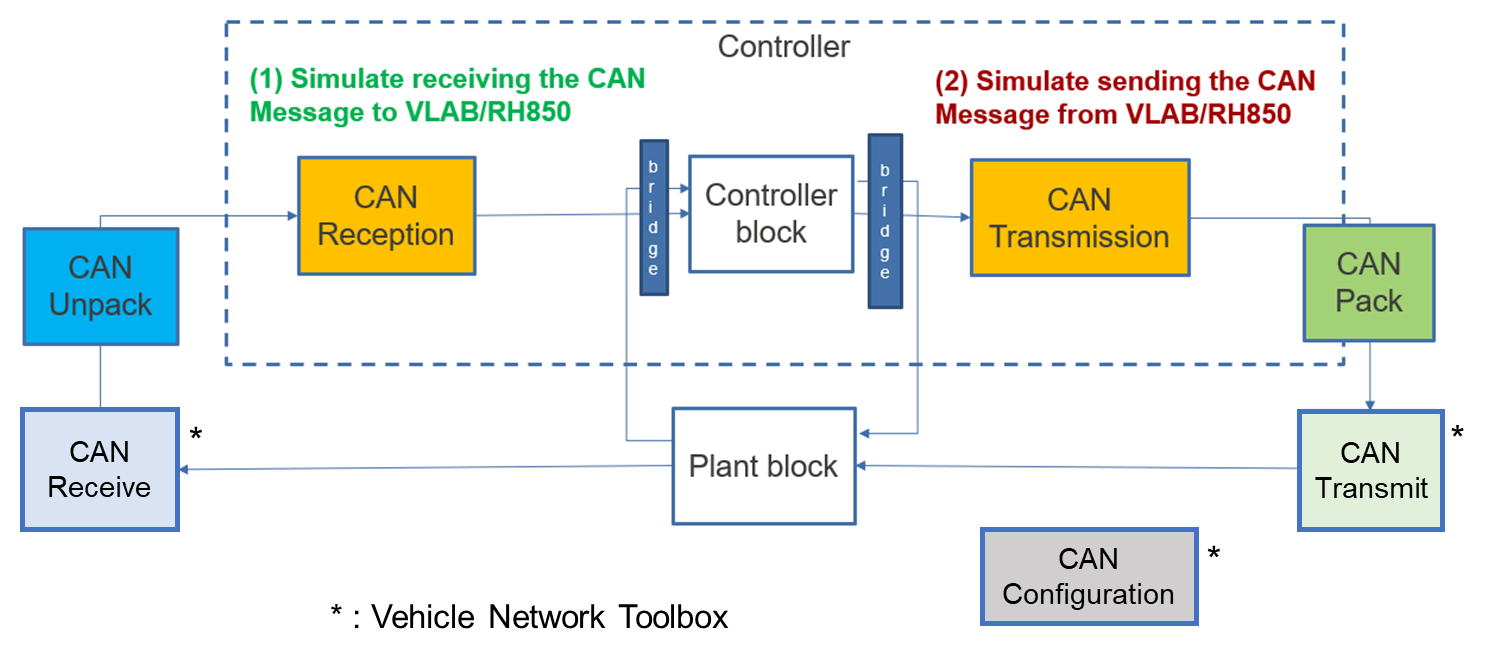
Diagram

Description automatically generated

**Figure 3‑12 Model structure for using RS-CANFD**

User must connect CAN Reception block and CAN transmission block and CAN Pack/Unpack same as above structure for conduct ETVPF with RS-CANFD.

- ETVPF support Vehicle Network Toolbox for integrating with CAN module of ETVPF.



**Figure 3‑13 Model structure for using RS-CANFD**

- Port name of CAN (F1KM-S1 and F1KM-S4 are same)

CAN0 RX (P0\_1)

CAN0 TX (P0\_0)

CAN1 RX (P0\_2)

CAN1 TX (P0\_3)

CAN2 RX (P0\_5)

CAN2 TX (P0\_4)

CAN3 RX (P0\_7)

CAN3 TX (P0\_8)

CAN4 RX (P0\_9)

CAN4 TX (P0\_10)

CAN5 RX (P0\_13)

CAN5 TX (P0\_14)

- Port name of CAN (for U2C)

CAN0 RX (T.B.D)

CAN0 TX (T.B.D)

CAN1 RX (T.B.D)

CAN1 TX (T.B.D)

CAN2 RX T.B.D)

CAN2 TX (T.B.D)

CAN3 RX (T.B.D)

CAN3 TX (T.B.D)

CAN4 RX (T.B.D)

CAN4 TX (T.B.D)

CAN5 RX (T.B.D)

CAN5 TX (T.B.D)

CAN6 RX (T.B.D)

CAN6 TX (T.B.D)

CAN7 RX (T.B.D)

CAN7 TX (T.B.D)

**Remark:** Currently, in HWM of U2C, the target port has not decided yet.

### **3.2.4. RLIN3n peripheral**

The following descriptions describe about the main features of RLIN3 peripheral.

* **For RLIN3n Transmission block:**

- This block receives the double data type from User Algorithm and transmit 8 bits through RLIN3 port. After that, it still receives data until it gets full data of 64 bits, then it transfers data to MATLAB (double value).

* **For RLIN3n Reception block:**

- This function receives the double data from MATLAB and transmit 8 bits through RLIN3 port. After that, it still receives data until it gets full data of 64 bits, then it transfers data to User Algorithm (double value).

Text

Description automatically generated with medium confidenceDiagram

Description automatically generated with medium confidence

**Figure 3‑14 S-Function block of RLIN3n**

The following descriptions describe how the User Interface of RLIN3n S-function block is used to support user selection and change RLIN3n ports easier during the setting model. The User Interface includes two types, which are corresponding to two RLIN3n S-function blocks (RLIN3n\_Send and RLIN3n\_Receive).

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application, email

Description automatically generated

**Figure 3‑15 User Interface of RLIN3n S-Function block**

The User Interface of Port S-Function block includes “RLIN3 unit” and “Select baud rate” parameters.

The purpose of these parameters is to specify the target port name of RLIN3n S-Function block. When building the model, these parameters will be generated at the same time as the input data.

After changing the unit of RLIN3n, the displaying name of S-function block will be changed automatically based on the current selecting RLIN3n unit.

Graphical user interface

Description automatically generated

**Figure 3‑16 The changing result of RLIN3n S-Function block**

### **3.2.5. TAU****D peripheral**

The following descriptions describe about the main features of TAUD peripheral.

* This block receives three input Start/Stop (boolean), pulse cycle (double), duty value (array of double value) from User algorithm to control ouput of PWM signals.
* Ouput of this block is an array of data signals (master channel and slave channels). Slave channels select by User in SC.
* PWM output: generate PWM data signal then send data to MATLAB (boolean).
* **Diagram

  Description automatically generated**Triangle PWM output: generate Triangle PWM data signal then send data to MATLAB (boolean).

**Figure 3‑17 S-Function block of TAUD**

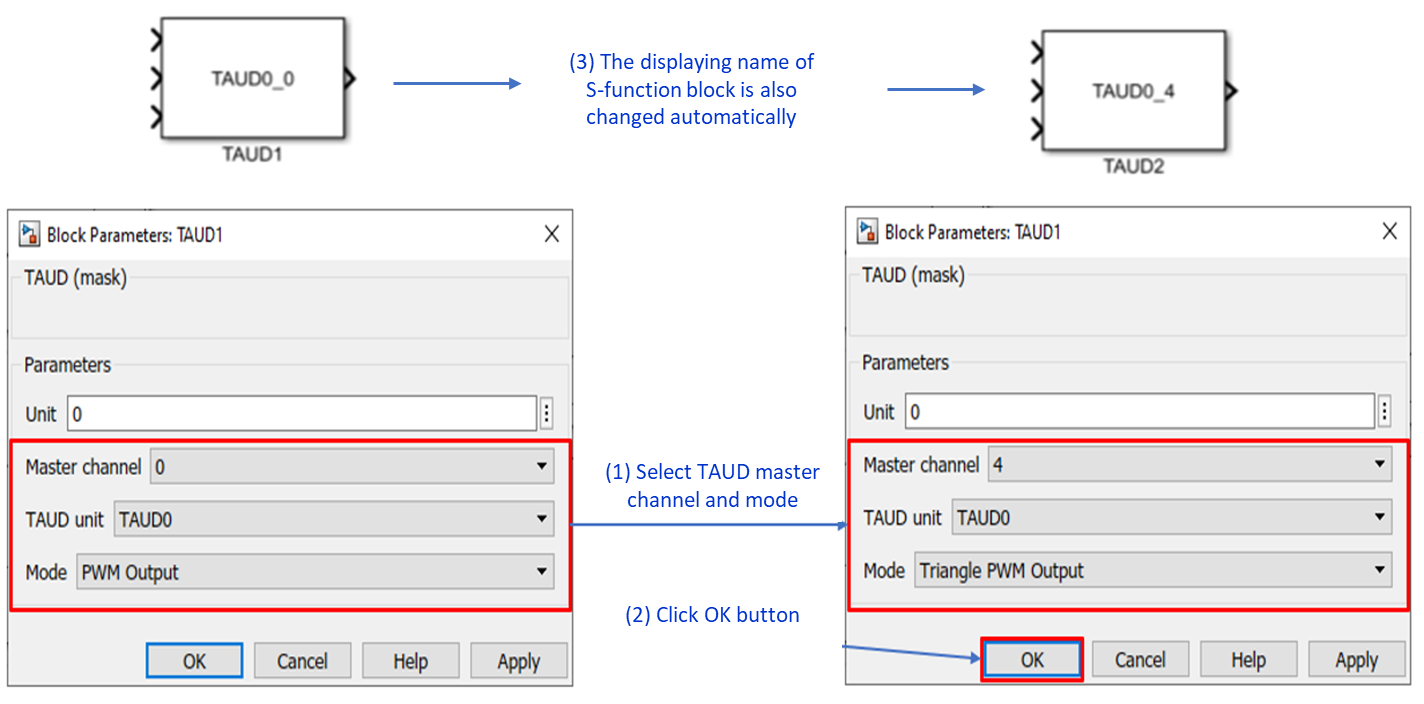
Graphical user interface, text, application, email

Description automatically generatedThe following descriptions describe how the User Interface of TAUD S-function block is used to support user selection and change TAUD ports easier during the setting model.

**Figure 3‑18 User Interface of TAUD S-Function block**

The User Interface of TAUD S-Function block includes “TAUD unit”, “Mode”, “Master channel” parameters.

The purpose of these parameters is to specify the target port name of TAUD S-Function block. When building the model, these parameters will generate at the same time as the input data.

After changing the master channel of TAUD, the displaying name of S-function block will be changed automatically based on the current selecting TAUD master channel.

**Figure 3‑19 Changing port name of TAUD S-Function block**

* **Chart, waterfall chart

  Description automatically generatedModel structure of connection between User block with TAUD S-Function block:**

**Figure 3‑20 Model structure for using TAUD**

User must connect TAUD block same as above structure for conduct ETVPF with TAUD. To support setting for multiple channels and connection between MATLAB and VLAB, use Mux and Demux to connect from User algorithm to TAUD block.

To keep the dimension of input and output are same, Input duty array must include a dummy value for master at the end of input array.

If user want to NOT change the input value after started, user can connect inputs to constant values. The constant value is same as initial setting value.

Example: connect Duty and Pulse to constant number

Output of TAUD is an array of data signals. To get data from each channel, user must extract data from array to single data signal by using Demux to get each data.

Output arrays include signal of master channel and slave channels. Input for slave is also array.

**- Support Model in the Loop Simulation (MILS) for TAUD block:**

ETVPF create TAUD block for executing MILS To use this block, user must replace the S-function TAUD to MILS TAUD block then execute MILS.

Input of MILS TAUD includes duty, pulse, and start/stop trigger. When running the model, the S-function “mils\_taud” inside TAUD MILS will set the duty input for “PulseWidth” and pulse input for “Period” of the pulse generator. Please match the input of MILS TAUD and the initial value of the pulse generator (“PulseWidth” and “Period”).

In MILS, user don’t need to use Mux and Demux block.

Chart

Description automatically generatedEach block support for each TAUD channel. To use multiple channels, user must connect to multiple blocks.

**Figure 3‑21 Model structure for using TAUD MILS**

## 3.3 Executing Simulator Processor in the Loop Simulation

The following describes how to generate a vHILS environment necessary for Simulator Processor in the Loop Simulation (hereafter referrerd to as vHILS) by ET-VPF.

### **3.3.1 Embedded sample model**

The user opens ETVPF package of the MATLAB Simulink Library Browser to choose the expected S-function blocks of peripherals.

* The ETVPF package is in MATLAB Simulink Library Browser, which contained the S-function blocks of peripherals.

Graphical user interface, application

Description automatically generated

**Figure 3‑22 The S-function blocks of peripherals are in ETVPF package**

The sample model is Power Window model (slexPowerWindowExample.slx) that used for the following explanation with ET-VPF. The feature of peripherals (Port, ADC, RS-CANFD, RLIN3n, TAUD) will be supported by S-Function blocks that added under the Code generation target.

Remarks 1. All blocks in the first layer under the Code generation target must be wrapped to a Subsystem.

2. The S-Function blocks of peripherals (Port, ADC, RS-CANFD, RLIN3n, TAUD) must be in the layers under the Code generation target. If they are in outside the Code generation target, error can occur.

Graphical user interface

Description automatically generated

***Figure 3‑23 The Code generation target and S-Function blocks of peripherals***

The following tables show the information about the components of sample model.

***Table 3‑2 Code generation target of sample model***

|  |  |  |
| --- | --- | --- |
| **Sample model name** | **Code generation target** | **Block type** |
| slexPowerWindowExample.slx | Target\_Block\_ET\_VPF | Subsystem block |

***Table 3‑3 Peripherals under Code generation target***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Code generation target** | **Peripheral block** | **Peripheral port name** | **Block name** | **Block type** |
| Target\_Block\_ET\_VPF | ADC | ANI<Unit><Port ID>\*1  AN<Unit><Port ID>\*2 | ADC\_<Number> | S-Function block |
| Port | P<Port>\_<Pin> | Port\_In\_<Number> |
| Port\_Out\_<Number> |
| RS-CANFD | CAN<unit>\_TX | CAN\_Transmission\_<Number> |
| CAN<unit>\_RX | CAN\_Reception\_<Number> |
| RLIN3n | RLIN3<RLIN3 unit>\_Send | RLIN3n\_Transmission\_<Number> |
| RLIN3<RLIN3 unit>\_Receive | RLIN3n\_Reception\_<Number> |
| TAUD | TAUD<Unit>\_<master channel> | TAUD\_<Number> |

\*1… Use for RH850/F1KM-S1 and RH850/F1KM-S4 device series.

\*2… Use for RH850/U2C device series.

### **3.3.2 Setting configuration parameters**

ET-VPF implements execution of vHILS environment generation by interworking with Embedded Coder. Therefore, it is necessary to check/set Embedded Coder options when using the vHILS environment generation functions provided by ET-VPF.

After placing peripheral blocks in the model, set the configuration parameters as follows:

1. Open MATLAB R2017b.
2. Select [Current Folder] is location which contains Power Window model. Open model, set model variables, select port name for S-Function of peripherals.
3. Open the [Model Configuration Parameters] dialog box to setting for the Power Window model.

* Graphical user interface, text, application, email

  Description automatically generated**Step 1:** Select [Solver] -> [Type] is “Fixed-step”.

***Figure 3‑24 [Solver] settings***

* **Step 2:** Setting for [Code Generation].
* Select [System target file] is “etvpf.tlc”.
* Graphical user interface, text, application, email

  Description automatically generatedSelect [Generate code only].

***Figure 3‑25 [Code Generation] settings***

* **Step 3:** In the [ET-VPF options], select necessary settings that described in the **Table 3‑4 ET-VPF Options**.

Graphical user interface, text, application, email

Description automatically generated

***Figure 3‑26 [ET-VPF options] settings***

The following table shows the items in the [ET-VPF options] pane.

***Table 3‑4 ET-VPF Options***

|  |  |  |
| --- | --- | --- |
| **Item name** | **Description** | |
| Cygwin Installation Directory \*1 \*14 | Specifies the folder where Cygwin has been installed (the folder where bin/bash.exe is stored) as an absolute path. | |
| [Use default Cygwin Installation Directory] checkbox | Specifies the default folder where Cygwin has been installed. It is “C:/cygwin64”. | |
| [Select Cygwin Installation Directory] button \*1 \*2 | Clicking this button displays the dialog box for selecting the absolute path of the folder where the Cygwin is installed. Folder specifications made in the dialog box that is opened by this button are reflected in the [Cygwin Installation Directory] field. | |
| VLAB Installation Directory \*3 \*14 | Specifies the folder where VLAB has been installed (the folder where vlab-ide.exe is stored) as an absolute path. | |
| [Use default VLAB Installation Directory] checkbox | Specifies the default folder where VLAB has been installed. It is “C:/Program Files/Vlab”. | |
| [Select VLAB Installation Directory] button \*3 \*4 | Clicking this button displays the dialog box for selecting the absolute path of the folder where the VLAB is installed. Folder specifications made in the dialog box that is opened by this button are reflected in the [VLAB Installation Directory] field. | |
| SC Installation Directory \*5 \*14 | Specifies the folder where Smart Configurator has been installed (the folder where SmartConfigurator.exe is stored) as an absolute path. | |
| [Use default Smart Configurator Installation Directory] checkbox | Specifies the default folder where Smart Configurator has been installed. It is “C:/Program Files (x86)/Renesas Electronics/SmartConfigurator/RH850/eclipse”. | |
| [Select Smart Configurator Installation Directory] button \*5 \*6 | Clicking this button displays the dialog box for selecting the absolute path of the folder where the Smart Configurator is installed. Folder specifications made in the dialog box that is opened by this button are reflected in the [SC Installation Directory] field. | |
| IDE Mode | Selects the type of project file that is loaded when the VLAB starts and whether or not a series of processing including download of a load module is performed after the VLAB start-up. | |
| Create Project (default) | The default project file provided by ET-VPF is loaded. |
| Build Tool \*7 | Selects the Build tool for the generated project, this indicates the compiler will be used to generate the load module. | |
| Renesas Compiler \*8 | Selects any of Renesas compilers, which will be determined by CS+ is reflected in the [CS+ Installation Directory] field. |
| CS+ Installation Directory \*9 \*14 | Specifies the folder where CS+ has been installed (the folder where CubeSuiteW+.exe is stored) as an absolute path. It's used for the path for Renesas compiler. | |
| [Use default CS+ Installation Directory] checkbox | Specifies the default folder where CS+ has been installed. | |
| [Select CS+ Installation Directory] button (\*9 \*10) | Clicking this button displays the dialog box for selecting the absolute path of the folder where the CS+ is installed. Folder specifications made in the dialog box that is opened by this button are reflected in the [CS+ Installation Directory] field. | |

***Table 3‑5 ET-VPF Options (Cont.)***

|  |  |  |
| --- | --- | --- |
| **Item name** | **Description** | |
| Device Series \*11 | Selects the series name of the microcontroller being used. | |
| <Device Series Name> \*12 | The supported Device Series described in **Table 1‑1 Supported devices** |
| N/A | Default value of Device Series. |
| OSTM PCLK \*13 | The OS Timer value corresponding to each device series. | |
| [Check Available License] button | Displays list of requiring licenses in ET-VPF System. | |
| [About] button | Displays version information and copyright information of ET-VPF. | |

\*1… When Cygwin has not been installed in the folder specified with the dialog box (bin/bash.exe file does not exist in the specified folder), an error is output, and the information of the specified folder is not reflected in [Cygwin Installation Directory].

This setting is only valid if “Create Project” is selected for [IDE Mode], the [Use default Cygwin Installation Directory] checkbox is not checked.

\*2… When the [Use default Cygwin Installation Directory] checkbox is checked, if the [Select Cygwin Installation Directory] button is clicked, an error message displays.

\*3… When VLAB has not been installed in the folder specified with the dialog box (vlab-ide.exe file does not exist in the specified folder), an error is output, and the information of the specified folder is not reflected in [VLAB Installation Directory].

This setting is only valid if “Create Project” is selected for [IDE Mode], the [Use default VLAB Installation Directory] checkbox is not checked.

\*4… When the [Use default VLAB Installation Directory] checkbox is checked, if the [Select VLAB Installation Directory] button is clicked, an error message displays.

\*5… When Smart Configurator has not been installed in the folder specified with the dialog box (SmartConfigurator.exe file does not exist in the specified folder), an error is output, and the information of the specified folder is not reflected in [SC Installation Directory].

This setting is only valid if “Create Project” is selected for [IDE Mode], the [Use default Smart Configurator Installation Directory] checkbox is not checked.

Remark This setting is unavailable if using RH850/U2C device series.

\*6… When the [Use default Smart Configurator Installation Directory] checkbox is checked, if the [Select Smart Configurator Installation Directory] button is clicked, an error message displays.

Remark This setting is unavailable if using RH850/U2C device series.

\*7… This setting is only valid if “Create Project” is selected for [IDE Mode].

\*8… When [Build Tool] is set to “Renesas Compiler”, the build tool is decided by CS+ at the source code compiling time.

\*9… When CS+ has not been installed in the folder specified with the dialog box (CubeSuiteW+.exe file does not exist in the specified folder), an error is output, and the information of the specified folder is not reflected in [CS+ Installation Directory].

This setting is only valid if “Create Project” is selected for [IDE Mode], the [Use default CS+ Installation Directory] checkbox is not checked.

\*10… When the [Use default IDE Install Directory] checkbox is checked, if the [Select IDE Install Directory] is clicked, an error message displays.

\*11… This setting is only valid if “Create Project” is selected for [IDE Mode], the “Embedded Target for RH850 Virtual Platform” license is valid to use.

\*12… The list of supported devices in the current ET-VPF version (refer to ***Table 1‑1 Supported devices***). **Table *1‑1 Supported devices***

\*13… Set following [OSTM PCLK] value Depending on the selected [Device Series]. These values are set as default values. These values are set as default values.

***Table 3‑6 OSTM PCLK Settings***

|  |  |
| --- | --- |
| **Device Series** | **OSTM PCLK [Hz]** |
| RH850/F1KM-S1 | 4000000 |
| RH850/F1KM-S4 | 2000000 |
| RH850/U2C | 80000000 |

\*14… The path of CS+, Cygwin, Smart Configurator and VLAB is only support the special characters that described in the following table.

***Table 3‑7 The supported special characters***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Special Characters** | **CS+ and Cygwin** | **Smart Configurator** | **VLAB** | **<ET-VPF installation folder>** |
| ! | O | O | O | O |
| @ | O | O | O | O |
| # | - | - | O | - |
| $ | - | O | O | - |
| % | - | - | - | O |
| ^ | - | - | O | O |
| & | - | - | O | O |
| ~ | O | O | O | O |
| ` | - | O | O | O |
| - | O | O | O | O |
| \_ | O | O | O | O |
| + | O | O | O | O |
| = | - | O | O | O |
| ( | - | O | O | - |
| ) | - | O | O | - |
| [ | O | O | O | O |
| ] | O | O | O | O |
| { | O | O | O | O |
| } | O | O | O | O |
| , | - | - | - | - |
| . | O | O | O | - |
| ‘ | - | - | - | - |
| ‘’ | - | O | O | - |
| ; | - | - | - | - |
| space | - | O | O | - |

*\** O *: Can use  
 -: Cannot use*

* **Step 4:** Click [Apply] button then save model.
* **Step 5:** Click [OK] or [X] button to close the [Model Configuration Parameters] dialog box.

### **3.3.3 Generating a vHILS environment**

This section explains how to execute generation of the vHILS environment required for vHILS.

ET-VPF provides the following command, which can be used in the MATLAB command window. This command automatically executes a series of operations for generation of a vHILS environment.

***Table 3‑8 Provided command***

|  |  |
| --- | --- |
| **Command name** | **Description** |
| run\_vlab | Generates a vHILS environment and executes vHILS automatically |

1. Select the Code generation target on model.
2. Execute generation of vHILS environment by entering the provided command in the MATLAB Command Window, using the following syntax.

Here ">>" denotes the command prompt and "[Enter]" denotes entry of the Enter key.

>> run\_vlab [Enter]

1. After executing “**run\_vlab**” command, the SC\_project folder will be created. And a message box is displayed to notify for user about the full path of SC\_project.

Remark If using RH850/U2C device series, this message box is not displayed and do not execute the feature which described in section **3.3.3.1 Generating the peripherals’ source code by SC**.

Graphical user interface, application, Teams

Description automatically generated

***Figure 3‑27 Message box notify the full path of SC\_project***

#### 3.3.3.1 Generating the peripherals’ source code by SC

1. After that, the SC Window will be displayed. User can configure detail settings and generate peripherals’ source code as below:

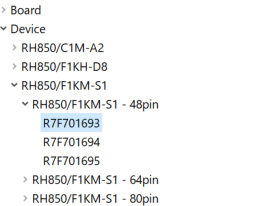
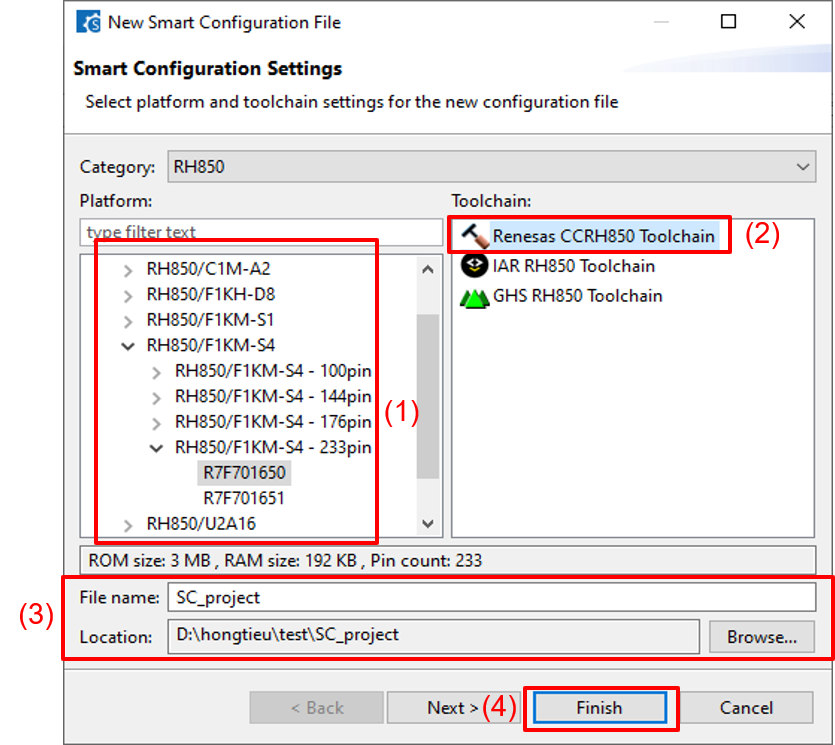
* **Step 1:** Select [New Configuration File] or [File] -> [New…] to create new SC project.

Graphical user interface, application, Word

Description automatically generated

***Figure 3‑28 Creating new SC project***

* **Step 2:** Select platform and toolchain settings:
* Select Device name in [Device] setting **(1)** (***Note:*** Currently, only RH850/F1KM device series are supported).
* Select Toolchain type in [Toolchain] setting **(2)** (***Note:*** Currently, only Renesas toolchains are supported).
* Fill [File name] and select [Location] settings **(3)** (***Note:*** The [Location] must be the full path of SC project displayed in message box).
* Click [Finish] button **(4)**.



***Figure 3‑29 Selecting SC settings***

* **Step 3:** Import available XML file to SC. (Skip this step if model contains only CAN)
* The XML file of each device name is provided to support user importing the components easier. This XML file (with name is “newxml.xml”) will be generated to the working directory.
* To import available XML file: Select [Component] **(1)**, select [Import Configuration] **(2)** settings.

Graphical user interface, application

Description automatically generated

***Figure 3‑30 Importing Configuration***

* Select [Import File] **(1)** and select the available XML file (newxml.xml).
* Click [Select All] button **(2)** if all configurations are necessary (***Note:*** Only select the necessary configurations).
* Click [Next >] button **(3)**.

Graphical user interface, text, application

Description automatically generated

***Figure 3‑31 Importing available XML file***

* The configuration of all peripherals is prepared. Click [Finish] button to complete importing configuration.

A computer screen capture

Description automatically generated with low confidence

***Figure 3‑32 Completed importing configuration***

Remarks 1. ET-VPF also support executes configuration functions of peripherals automatically. To do this, the configuration function name must be a determined format. Therefore, when generate peripherals’ source code by SC, the configuration name must be default name.

2. For each RLIN3n unit, it only has one configuration function, which corresponds to UART resource (e.g., RLIN30 corresponds to “UART0” resource). If we use two RLIN3n units with the same UART resource, the error will occur.

3. For TAUD, user must select the clock source and slave channels in SC GUI.

* Graphical user interface, application

  Description automatically generated**Step 4:** Change detail settings (skip this action if model contains only CAN) and generate source code for peripherals.

***Figure 3‑33* Setting slave channels and clock source for TAUD**

Graphical user interface, application

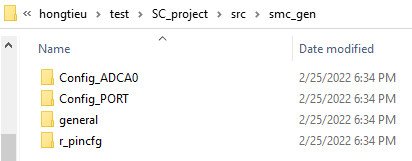
Description automatically generated

***Figure 3‑34 Selecting detail settings and generating peripherals’ source code***

Remark When change detail settings, these settings must be the same with the peripherals’ settings on model. If these settings are different with the peripherals’ settings on model, it causes vHILS execution incorrect.

* **Step 5:** Close SC Window.

1. The source code of peripherals will be generated under SC\_project folder.



***Figure 3‑35 The peripherals’ source code generated under SC\_project***

#### 3.3.3.2 Generating the target vHILS environment

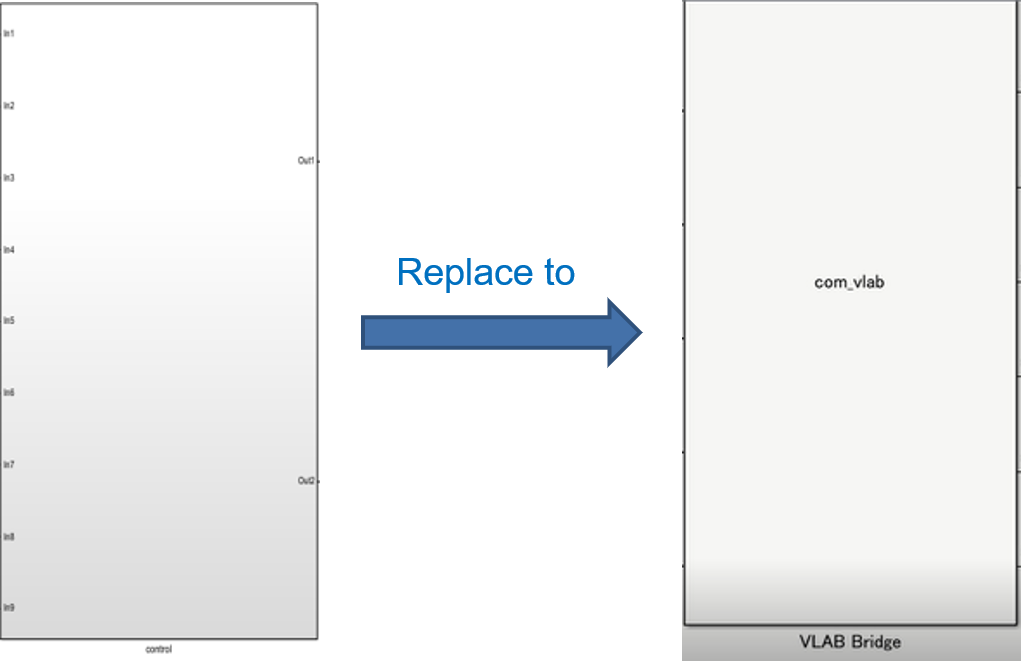
The target vHILS environment will be generated in the working directory (the location which contains target model).

The target vHILS environment includes “slprj” and “<Code generation target>\_etvpf” folders. And the necessary files will be generated in these folders.

* Source files of target device.
* C source files.
* Make file
* VLAB execution file
* Peripherals’ wrapper source files
* Python source files.
* Define.h, OSTM\_define.h, target.out.

The model file is copied (the destination model file has the same name as the original model file but "\_etvpf" suffix is added).

The Subsystem under the Code generation target block is replaced with the block for vHILS sequential execution (with block name is “VLAB Bridge”) for the model file to be copied.



***Figure 3‑36 Example of replacing to the block for PIL sequential execution***

### **3.3.4 Compiling generated source code**

1. After that, the C source files, and source files of target device (with the extension is \*.c, \*.850) will be compiled to object files (with the extension is \*.o, \*.ao) via Cygwin.

The information related to the compiling will be displayed on MATLAB Command Window.

Table

Description automatically generated

***Figure 3‑37 Compiling is displayed on MATLAB Command Window***

1. Then, the “target.out” file is also generated. After that, the object files and the “target.out” file are also stored in the working directory, under “<Code generation target>\_etvpf” folder.

### **3.3.5 Executing vHILS**

1. After the compiling completed, the VLAB is opened. The RH850 Virtual Platform of RH850/F1KM, RH850/U2C toolbox, “target.out” and source files in the working directory are loaded.

The vHILS execution start on both Simulink model and VLAB.

Graphical user interface, diagram

Description automatically generated

***Figure 3‑38 The vHILS execution***

1. The result of scope is below, MIL and ET-VPF results are the same.

Graphical user interface

Description automatically generated

***Figure 3‑39 The result of scope***

Remark Please ignore the first step. Because, at that time, MATLAB and VLAB is connect and dummy value is sent to MATLAB by the MATLAB Co-simulation.

## 3.4 Time measurement

This section describes time measurement method which used in ET-VPF

### **3.4.1 Structure of Simulink Model for measurement**

Graphical user interface

Description automatically generated

***Figure 3‑40 Structure of Simulink model for measurement***

Remarks (\*1): The measured block meet all requirement for measurement.

(\*2): User decision: Not measure this block. So, they don't need to transform it to Atomic Subsystem. Refer to section **3.4.3 How to change Normal Subsystem to Atomic Subsystem**.

(\*3): Although it is Atomic Subsystem, but user do not define in the input\_subsystem.txt file.

Below is the requirement for the measured block:

1. Only measure for the block inside the User Block.

2. Only measure for the block that define in the input\_subsystem.txt. Refer to section **3.4.2 Input file for Measurement**.

3. It must be the Atomic Subsystem.

### **3.4.2 Input file for Measurement**

In the model folder, to decide which measured block, preparing the input\_subsystem.txt with below format:

<Path of Atomic\_Subsystem\_name\_1>,<First core>

<Path of Atomic\_Subsystem\_name\_2>,<First core>

…

<Path of Atomic\_Subsystem\_name\_N>,<First core>

For example:

Text

Description automatically generated

***Figure 3‑41 Examples of input\_subsystem.txt for Time Measurement***

### **3.4.3 How to change Normal Subsystem to Atomic Subsystem**

To change the Normal Subsystem to Atomic Subsystem, do the following step:

1. Right click on the Normal Subsystem -> Select Property
2. In the Main tab, select the [treat as atomic unit] setting.
3. In the Code Generation tab, set [Function packaging] to “Reusable function”.

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application, email

Description automatically generated

***Figure 3‑42 Change Normal Subsystem to Atomic Subsystem***

### **3.4.4 Graph Viewer**

The following describes how to display the time measurement results using Graph Viewer.

#### 3.4.4.1 Input data

Graph Viewer has two input files:

Input 1: execution\_data.csv file refer to upper data structure.

* Its unit is picosecond.
* It contains the execution time of each step and start end of each subsystem.

Graphical user interface, application, table, Excel

Description automatically generated

Input 2: input\_subsystem.txt file that specifies the subsystem to be measured, refer to section **3.4.2 Input file for Measurement**.

#### 3.4.4.2 Output figure

The following is the output figure that will be displayed when executing command of Graph Viewer (ETVPF\_Launch\_GraphViewer.m) in the MATLAB Command Window, using the following syntax.

Here ">>" denotes the command prompt and "[Enter]" denotes entry of the Enter key.

>> ETVPF\_Launch\_GraphViewer [Enter]

Chart

Description automatically generated

***Figure 3‑43 Output figure of time measurement***

Remark Currently, in the Graph Viewer do not have the execution time of peripheral source code.

Users can perform some other functions as followings:

1. Press [Show Step Graph] to shows execution time of each Subsystem for one step or more steps belong to user choose in “Start Step” and “End Steps”.

Graphical user interface

Description automatically generated

***Figure 3‑44 Press [Show Step Graph] to show execution time each Subsystem***

Set “Start Step” and “End Start” (e.g., Start Step = 1, End Step = 1).

Chart

Description automatically generated with medium confidence

***Figure 3‑45 Example of Execution Time Detail at step 1***

1. Change scale of step by press [Change Scale] button then set “Start Step” and “End Start” (e.g., Start Step = 1, End Step = 228).

Chart

Description automatically generated

***Figure 3‑46 Press [Change Scale] to change scale of step***

Graphical user interface, application

Description automatically generated

***Figure 3‑47 Set “Start Step” and “End Step” value***

Then pressing [OK], the graph scale will change accordingly.

1. Press [MAX/min] to shows which steps has the maximum execution time and minimum execution time.

Chart

Description automatically generated with medium confidence

***Figure 3‑48 Show Max/Min execution time***

Then the Max Min Dialog show step has execution time maximum/minimum and the value of execution time of this step.

Press [Show step graph] in Maximum/Minimum Section to shows execution time detail of this step.

Graphical user interface, text, application

Description automatically generated

***Figure 3‑49 Max/Min Dialog***

# 

# **Points for Caution**

This section describes limitations and points to note when using ET-VPF.

## 4.1 Features

The descriptions of affected features are removed in ET-VPF User’s Manual.

1. In the current SC (version 1.5.0), do not have RH850/F1KM-S4 - 272 pins device series. Therefore, this device series cannot use in this development.
2. The name for measured blocks must be unique.

Current method to get the generated function for measured block is using the information in the comment. Its comment only has the Subsystem name. Then, cannot identify the target measured block by the Subsystem full path.

1. There are some functions of driver code which are not guaranteed:

* R\_Config\_<CAN Channel>\_Transmission\_Stop function: Because ET-VPF guideline is not for stopping, it is only for executing vHILS. Therefore, this function is just for user reference and not guaranteed.
* CAN\_Common\_Reception\_Init function: for executing vHILS, ET-VPF use the CAN\_Common\_Init function for initializing for both transmission and reception. Therefore, this function is just for user reference and not guaranteed.

1. The features for RH850/U2C are preliminary because it's under development.
2. For ADC in RH850/U2C this version only supports for Unit 1 (ADCK1), Scan group 1, Multicycle scan mode.

For ADC in RH850/F1KM this version only supports for Scan group 1, Multicycle scan mode.

1.  pin is assigned P8-6 (for RH850/F1KM-S1 and RH850/F1KM-S4) or P27\_0 (for RH850/U2C), so use other port.
2. To synchronize the communication between VLAB and MATLAB, sampling time must higher than the execution time of algorithm each step. When the execution time higher than sampling time, the output is not guaranteed. For specific execution time, it describes in the “Time” column of execution\_data.csv in “<Code generation target>\_etvpf” folder.
3. Duty time and pulse time of TAUD must be higher than sampling time.

As ETVPF spec, signal of PWM Output will be sent to MATLAB each sampling time

For TAUD, when using duty time and pulse time smaller than sampling time, during receive output of PWM from VPF to MATLAB, some signal may be missed. Thus, Duty time and pulse time of TAUD must be higher than sampling time.

1. TAUD output signal of VPF delays a little bit

In actual VPF, the TAUD clock has some delay at initialization and it may be the same as the actual chip (not ideal as spec, high value immediately when start TAUD).

For workaround, we prepared a unit delay block at the output of MILS TAUD block. By applied this method, waveform of MILS and vHILS are the same, except the first step value we not sure (currently the initial value of unit delay is 0)

1. TAUD MILS for Triangle PWM Output is not supported.
2. For CA mode, user can enable it, but it is just the evaluation feature in this version

If user wants to use CA mode, please follow the below steps:

**Step 1**: Run ETVPF for the first time and keep the environment.

**Step 2**: Change some strings in ETVPF\_testbench\_sample.py to replace FastIss to CAIss.

* vlab.read\_register("RH850.CPUSS.PE1\_FastIss.r31") change to vlab.read\_register("RH850.CPUSS.PE1\_CAIss.r31")
* vlab.load('rh850.f1km', ['--device-type=S1', '-t', './sim.py']) change to vlab.load('rh850.f1km', ['--device-type=S1', '-t', './sim.py', '-iss=ca'])

**Step 3**: Rerun the current Simulink model manually

**Step 4**: Rerun the ETVPF\_testbench\_sample.py manually.

1. The installation path of ET-VPF package, CS+, Cygwin, Smart Configurator and VLAB cannot contain some special characters (refer to ***Table 3‑7 The supported special characters*** for more detail).

## 4.2 Simulink Models

### **4.2.1 Available Strings for Paths and Block Names**

Do not use 2-byte characters (Japanese, etc.), spaces, slashes, line feeds, or hyphens for the names of code generation target blocks or paths to folders where the Simulink models are saved. If 2-byte characters are used for code generation target block names, they will be replaced with strings given by MATLAB, and if hyphens or spaces are used, their subsequent strings will be omitted. A series of ET-VPF operations is possible but not guaranteed.

### **4.2.2 Models Handling Complex Number Data**

Code generation from the MATLAB/Simulink models (subsystems) handling the complex number data is not supported.

## 4.3 Construction and Simulation

### **4.3.1 Length of Path to Code Generation Folder**

The path length to files and directories is restricted in the Windows platform (limit of 260 characters). If the path length to the code generation folder containing source code generated by the Simulink model is too long, MATLAB will display an error message on MATLAB Command Window.

Error: Build failed because the build file name(s) exceed the Windows limit of 260 characters. Build from a working directory with a shorter path, to allow build files to be created with shorter filenames. “<invalid file name>”

### **4.3.2 Notes on Power Management**

If the PC goes to sleep or hibernate mode while working on ET-VPF, an error may occur during the operation after rebooting.

### **4.3.3 Length of Script File Name**

For the length of script file name, it is limited by MATLAB. If its name is 64 characters or more, MATLAB will not allow executing it and an error will occur.

**4.3.4 Install Drive and Work Drive**

Make sure the ET-VPF installation drive and the working drive that stores the model are the same. Otherwise, an error will occur.

# 

# **Error Messages**

This section explains the error messages output by ET-VPF.

## 5.1 Overview

Error messages are output to notify you of information that you should know about events that occur while you are setting [ET-VPF options] in the Configuration Parameters dialog box or while a vHILSimulation is running.

Remark Error messages output by ET-VPF are not linked to VLAB. Therefore, no help is displayed even if you press the F1 key after ET-VPF displays an error message.

## 5.2 Errors Detected in Configuration Parameters Dialog Boxes

The following table lists the messages that are output when an error is detected while settings are being made in the Configuration Parameters dialog box.

These error messages are output to the ET-VPF Error dialog box.

***Table 5‑1 Error messages display when setting in the Configuration Parameters dialog box***

| [Message] | E0101  The cygwin directory setting is incorrect. |
| --- | --- |
| [Explanation] | This error message is displayed when information on the correct installation destination has not been set in [Cygwin Path]. |
| [Action by User] | 1. Set the correct path of the installation destination of Cygwin to [Cygwin Path].  The folder containing “/bin/bash.exe” must be specified. 2. Check that Cygwin has normally been installed. |
| [Message] | E0102  The VLAB directory setting is incorrect. |
| [Explanation] | This error message is displayed when information on the correct installation destination has not been set in [VLAB Path]. |
| [Action by User] | 1. Set the correct path of the installation destination of VLAB to [VLAB Path].  The folder containing “vlab-ide.exe” must be specified. 2. Check that VLAB has normally been installed. |
| [Message] | E0103  The CS+ directory setting is incorrect. |
| [Explanation] | This error message is displayed when information on the correct installation destination has not been set in [CS+ Path]. |
| [Action by User] | 1. Set the correct path of the installation destination of CS+ to [CS+ Path].  The folder containing “CubeSuiteW+.exe” must be specified. 2. Check that CS+ has normally been installed. |
| [Message] | E0104  The Smart Configurator directory setting is incorrect. |
| [Explanation] | This error message is displayed when information on the correct installation destination has not been set in [SC Path]. |
| [Action by User] | 1. Set the correct path of the installation destination of Smart Configurator to [SC Path]. The folder containing “SmartConfigurator.exe” must be specified.   2. Check that Smart Configurator has normally been installed. |

***Table 5‑2 Error messages display when setting in the Configuration Parameters dialog box (cont.)***

| [Message] | E0106  The default directory is selected as the current installation directory.  Deselect the use of the default directory. |
| --- | --- |
| [Explanation] | 1. This error message displayed when [Select Cygwin Installation Directory] button is clicked, while the [Use default Cygwin Installation Directory] checkbox is checked. 2. This error message displayed when [Select VLAB Installation Directory] button is clicked, while the [Use default VLAB Installation Directory] checkbox is checked. 3. This error message displayed when [Select Smart Configurator Installation Directory] button is clicked, while the [Use default Smart Configurator Installation Directory] checkbox is checked.   4. This error message displayed when [Select CS+ Installation Directory] button is clicked, while the [Use default CS+ Installation Directory] checkbox is checked. |
| [Action by User] | 1. Uncheck [Use default Cygwin Installation Directory] checkbox. 2. Uncheck [Use default VLAB Installation Directory] checkbox. 3. Uncheck [Use default Smart Configurator Installation Directory] checkbox.   4. Uncheck [Use default CS+ Installation Directory] checkbox. |
| [Message] | E0110  A license is not registered. |
| [Explanation] | This message displays when no license or license was expired on your system. |
| [Action by User] | 1. Register ET-VPF License with Renesas Electronics Corporation. |
| [Message] | E0111  <Device Series> is not available. Register a valid license. |
| [Explanation] | This error message displayed when either the license for any RH850 device series is invalid. |
| [Action by User] | 1. If you don’t have the license for RH850 device series, please contact a Renesas Electronics sales office. 2. If you have got license RH850 device series, check if it is put in the ET-VPF installation. 3. To confirm the availability of the license, please [Check Available License] on [ET-VPF options] panel. |

## 5.3 Errors during vHILS execution

The following describes error messages detected when you get visual display of the execution time in algorithm verification after executed vHILS. Error dialog boxes during vHILS are output from MATLAB/Simulink

***Table 5‑3 Error messages display when getting the execution time***

| [Message] | E0201  Wrong measurement data. Please check data input files. |
| --- | --- |
| [Explanation] | The execution time measurement result is wrong or empty. |
| [Action by User] | 1. Check if the model settings, components are correctly. 2. If above information is correct, re-execute vHILS to generate execution time measurement result again. |

Index

Ａ

Atomic 44, 45

Ｅ

error messages 54

Ｇ

Graph Viewer 46

Ｉ

Installation 11

Ｌ

License types 8

Ｍ

maximum execution time 50

MILS 5, 27, 51

minimum execution time 50

Ｏ

Operating Environment 6

Ｐ

peripheral block 15

peripherals’ source cod 5

peripherals’ source code 36

**Points for Caution** 51

Ｓ

Simulink 6

Ｔ

time measurement method 44

Ｕ

Uninstallation 13

Ｖ

vHILS 5

vHILS environment 35

VLAB 6

VPF 5

| REVISION HISTORY | Embedded Target for RH850 Virtual Platform V1.00.00  RH850 Model-Based Development Tool  User’s Manual |
| --- | --- |

| Rev. | Date | Description | |
| --- | --- | --- | --- |
| Page | Summary |
| 0.01 | June 28, 2022 | 36 | Update the contents that related to F1KM-S4 |
| 0.01 | June 28. 2022 | 16, 18, 26, 26, 31, 40 | Update the contents that related to RLIN3n |
| 0.02 | July 01, 2022 | 7, 8, 9, 11, 12, 18, 31, 35, 37, 44, 53 | Update the contents of U2C alpha version  Update the contents of Port, ADC for U2C |
| 0.03 | July 18, 2022 | 16, 18, 27, 31, 62 | Update the contents that related to TAUD |
| 0.03 | July 19, 2022 | 18,22,23 | Update the contents that related to CAN |
| 0.04 | August 22, 2022 | 28, 53 | Update after implement coding phase |
| 0.05 | September 19, 2022 | 39, 40 | Update the contents after fixed REL’s comments |
| 0.06 | September 22, 2022 | 7, 30, 46 | Update the contents of U2C alpha version after received new toolbox |
| 0.07 | October 19, 2022 | 10, 11, 12, 14 | Update the contents of ET-VPF installer |
| 0.08 | October 31, 2022 | 12, 14, 15 | Update the installation procedure |
| 0.09 | November 03, 2022 | 18 | Update ADC unit 1 for RH850/F1KM-S4  Remove the remark related to ADC unit which is supported in RH850/F1KM-S4 |
| 0.10 | November 25, 2022 | 8, 17, 36, 53, 55 | Update for REL’s comments |
| 0.11 | November 25, 2022 | 13, 16, 18, 30 | Preparing peripheral S-function blocks to MATLAB Simulink Library Browser |
| 0.12 | Dec 02, 2022 |  | Update for supporting MATLAB R2021a |
| 0.13 | Nov 29, 2022 | 25 | Update for supporting CAN for U2C  Remove MATLAB R2021a information (not open to user) |

R20UT4486EJ0001

|  |
| --- |
| Embedded Target for RH850 Virtual Platform  V1.00.00  RH850 Model-Based Development Tool User's Manual  Publication Date: Rev.1.01 October 26, 2022  Published by: Renesas Electronics Corporation |

A4_j_logo5

Embedded Target for RH850 Virtual Platform V1.00.00

RH850 Model-Based Development Tool

R20UT5142EJ0101