

# Embedded Target for RH850 Virtual Platform

## V1.00.00

### RH850 Model-Based Development Tool

User's Manual

Target Device

RH850/F1KM

RH850/U2B

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>).

## 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 described 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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
  5. 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.
  6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. **RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.**
  8. 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.
  9. 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.
  10. 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.
  11. 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.
  12. 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.
  13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
  14. 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.5.0-1 October 2020)

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,  
Koto-ku, Tokyo 135-0061, Japan

[www.renesas.com](http://www.renesas.com)

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

## Contact information

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/](http://www.renesas.com/contact/)

# 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	<p>The contents of this manual are as follows:</p> <ul style="list-style-type: none"><li>Chapter 1 Overview</li><li>Chapter 2 Installation</li><li>Chapter 3 Functional Operation Procedure</li><li>Chapter 4 Points for Caution</li><li>Chapter 5 Error Messages</li></ul>	
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
1.1	Features .....	.6
1.2	Operating Environment .....	.6
1.3	License Types and Functions .....	.8
2	Installation .....	.9
2.1	Installation .....	.11
2.2	Uninstallation .....	.13
3	Functional Operation Procedure .....	.15
3.1	Overview .....	.15
3.2	S-Function block of peripherals .....	.17
3.2.1	ADC peripheral .....	.17
3.2.2	Port peripheral .....	.19
3.2.3	RS-CANFD peripheral .....	.21
3.2.4.	RLIN3n peripheral .....	.25
3.2.5.	TAUD peripheral .....	.26
3.3	Executing virtual Hardware in the Loop Simulation .....	.29
3.3.1	Embedded sample model .....	.29
3.3.2	Setting configuration parameters .....	.31
3.3.3	Generating a vHILS environment .....	.37
3.3.4	Compiling generated source code .....	.44
3.3.5	Executing vHILS .....	.46
3.4	Time measurement .....	.48
3.4.1	Structure of Simulink Model for measurement .....	.48
3.4.2	Input file for Measurement .....	.49
3.4.3	How to change Normal Subsystem to Atomic Subsystem .....	.49
3.4.4	Graph Viewer .....	.50
4	Points for Caution .....	.55
4.1	Features .....	.55
4.2	Simulink Models .....	.57
4.2.1	Available Strings for Paths and Block Names .....	.57
4.2.2	Models Handling Complex Number Data .....	.57
4.2.3	Constant input of peripheral S-function block .....	.57
4.3	Construction and Simulation .....	.57
4.3.1	Length of Path to Code Generation Folder .....	.57
4.3.2	Notes on Power Management .....	.57
4.3.3	Length of Script File Name .....	.57
4.3.4	Install Drive and Work Drive .....	.58
4.3.5	Build the CS+ project when using the “Generate the CS+ project and automatically build” .....	.58
5	Error Messages .....	.59
5.1	Overview .....	.59
5.2	Errors Detected in Configuration Parameters Dialog Boxes .....	.60
5.3	Errors during vHILS execution .....	.62
5.4	Errors during starting CS+ and building CS+ project .....	.63
5.5	Errors during building and generating the vHILS environment .....	.64
5.6	Errors during communication between MATLAB and VLAB .....	.64
Index .....	.65	
REVISION HISTORY .....	.66	

## 1 Overview

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

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 referred 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.

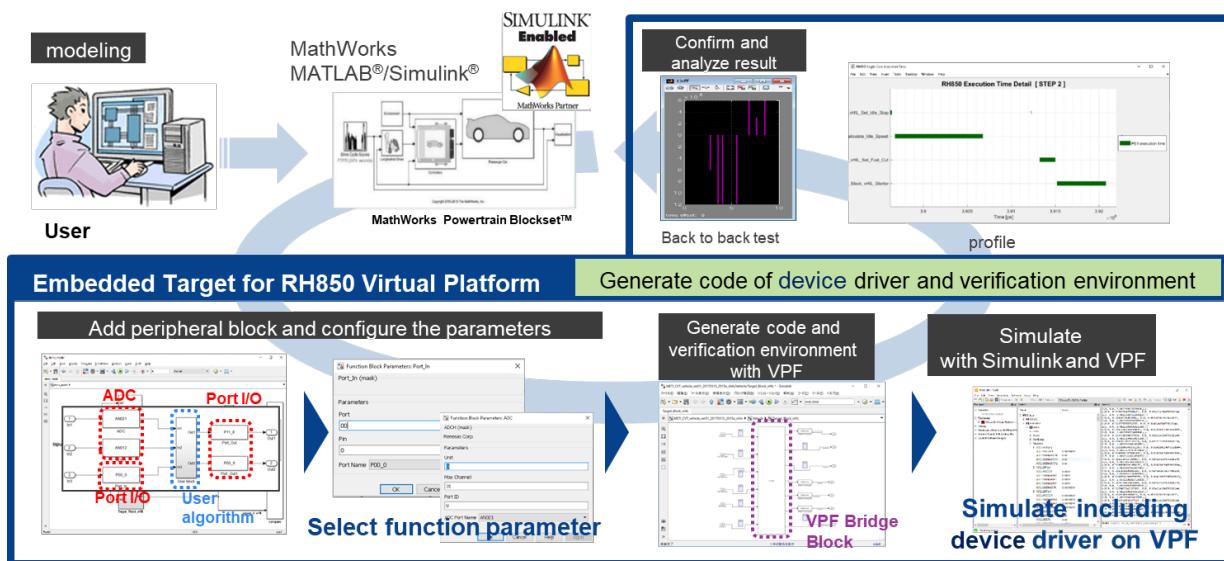


Figure 1-1 Overview of Embedded Target for RH850 Virtual Platform

## 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/U2B6, RH850/U2B10

- 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	V9.8 (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)

Note: MATLAB R2020a is for RH850/U2B device series, R2017b is for RH850/F1KM-S1 and RH850/F1KM-S4.

- 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++ 2015 compiler (from Microsoft Corporation)

Reference: System Requirements & Platform Availability

[https://www.mathworks.com/support/sysreq/previous\\_releases.html](https://www.mathworks.com/support/sysreq/previous_releases.html)

- MinGW V7.3.0

- VLAB (from The VLAB Works Pty Ltd)
  - VLAB V2.7.3 (win-vc140-x64) (For RH850/F1KM-S1, RH850/F1KM-S4)
  - V2.8.3 (win-vc140-x64) (For RH850/U2B device series)
- Toolboxes of VLAB
  - Toolbox RH850 Virtual Platform V3.2.1 (For RH850/F1KM-S1, RH850/F1KM-S4)
  - Toolbox CAN 2.5.0 (For supported peripheral RS-CANFD)
- Toolboxes of VLAB (included with VLAB V2.8.3)
  - Toolbox CAN 2.5.0 (For supported peripheral RS-CANFD)
  - Toolbox RH850 G4 Virtual Platform 1.22.1 (For supported devices RH850/U2B)
- Smart Configurator for RH850 (from Renesas Electronics)
  - SC V1.5.0 (For RH850/F1KM device series)
  - V1.8.0 (For RH850/U2B device series)
- Build Tool
  - CS+ V8.07.00 (from Renesas Electronics) (For RH850/F1KM device series)
  - CS+ V8.09.00 (from Renesas Electronics) (For RH850/U2B device series)
- Compiler
  - CC-RH V2.05.00 (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. Can use the upper software versions if the features used for ET-VPF be not changed but Renesas Electronics does not guarantee it.  
4. 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		Embedded Target for RH850/F1x Virtual Platform	Embedded Target for RH850/U2x Virtual Platform
Supported device	<b>RH850/F1KM-S1</b>	O	-
	<b>RH850/F1KM-S4</b>	O	-
	<b>RH850/U2B6</b>	-	O
	<b>RH850/U2B10</b>	-	O
Time measurement feature		O	O
Supported build tool	<b>GHS Compiler</b>	-	-
	<b>Renesas Compiler</b>	O	O

\* O : Can execute  
-: Cannot execute

## 2 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.

After the installation, 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\	... Contains the template of make file.
	ETVPF_include\Renesas\	... Contains the source code of each device which used for Renesas compiler.  <b>Note:</b> User can modify boot code files in “startup” folders but Renesas Electronics does not take responsibility for quality.
	ETVPF_include\XML_input\	... Contains the XML input files which provided for each RH850 device series.
Source code files		... Contains the list of source code files (include: *.p, *.tlc, make files) ... which used to execute the program of ET-VPF.

## (2) Folder structure under the ET-VPF package folder for RH850/U2B device series.

<ET-VPF installation folder>\<version information>\U2B\ETVPF_package\	ETVPF_include\ETVPF_S_function_block\	...	Contains the source code which used to execute the program of peripherals.
	ETVPF_include\make\	...	Contains the template of make file.
	ETVPF_include\Renesas\	...	Contains the source code of each device which used for Renesas compiler.
	ETVPF_include\XML_input\	...	<b>Note:</b> User can modify boot code files in "startup" folders but Renesas Electronics does not take responsibility for quality.
	Source code files	...	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.

## Remarks

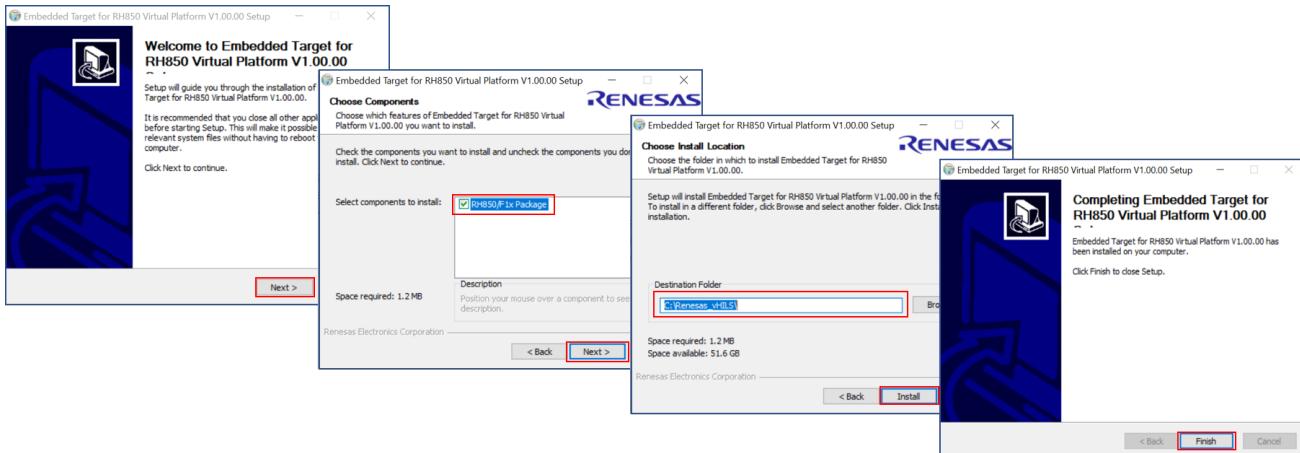
1. For <ET-VPF installation folder>, the user can change them when installing (the user needs to make sure that the installation locations are not limited permission). And the <ET-VPF installation folder> must be in the same <system drive> with the user's workspace to avoid the error occurring when loading the loaded module to RH850 virtual platform.
2. 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 or ETVF\_U2x\_<version information>\_Setup.7z) file.
- **Step 2:** Double click to (ETVPF\_<version information>\_Setup.exe or ETVF\_U2x\_<version information>\_Setup.exe) to start installing.
- **Step 3:** Execute step by step to install ET-VPF package as the following figure.

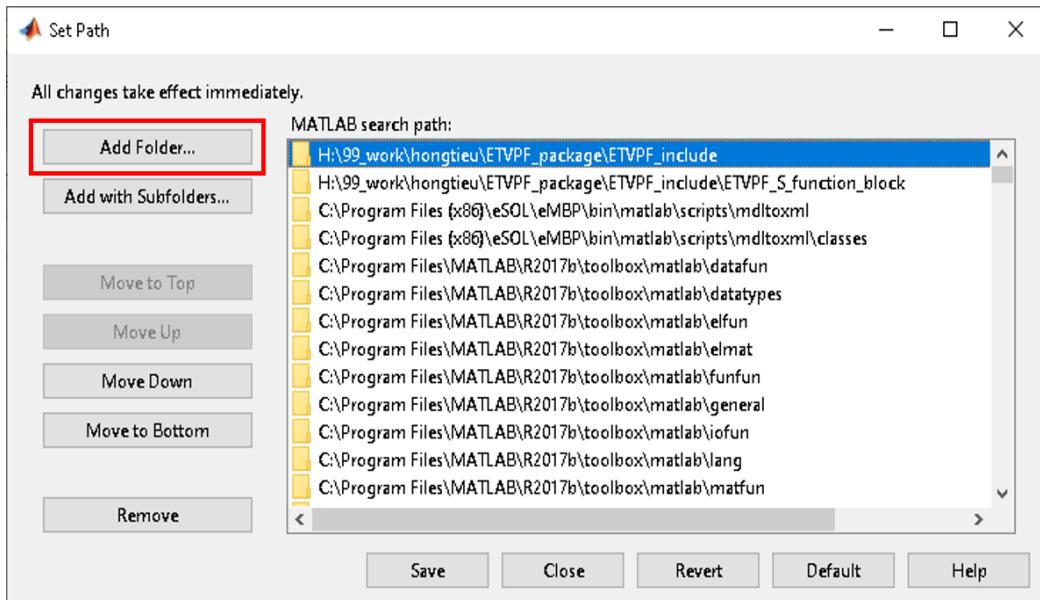


**Figure 2-1 Installation Procedure**

(2) 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



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

(3) 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.

(4) Register the license.

Add the ET-VPF license by license manager included with CS+ (refer License Manager [V2.05.00 User's manual](#)).

(5) 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.

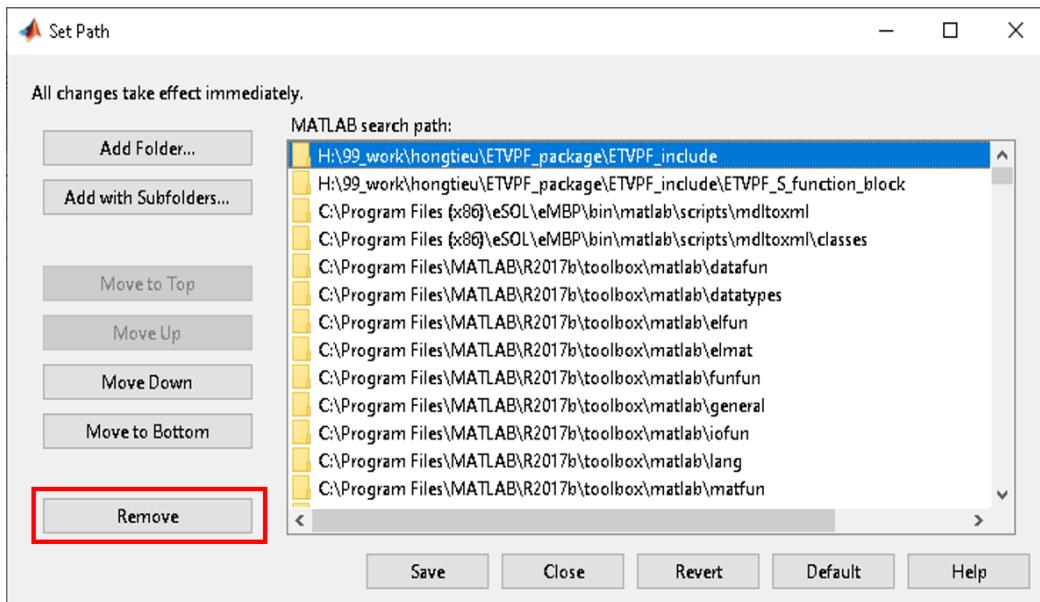
Here ">>" denotes the command prompt and "[Enter]" denotes entry of the Enter key.

```
>> setup_etvpf_lib [Enter]
```

## 2.2 Uninstallation

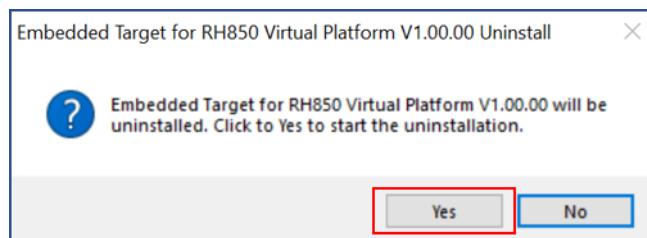
The following describes the procedure for uninstalling ET-VPF.

- (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.

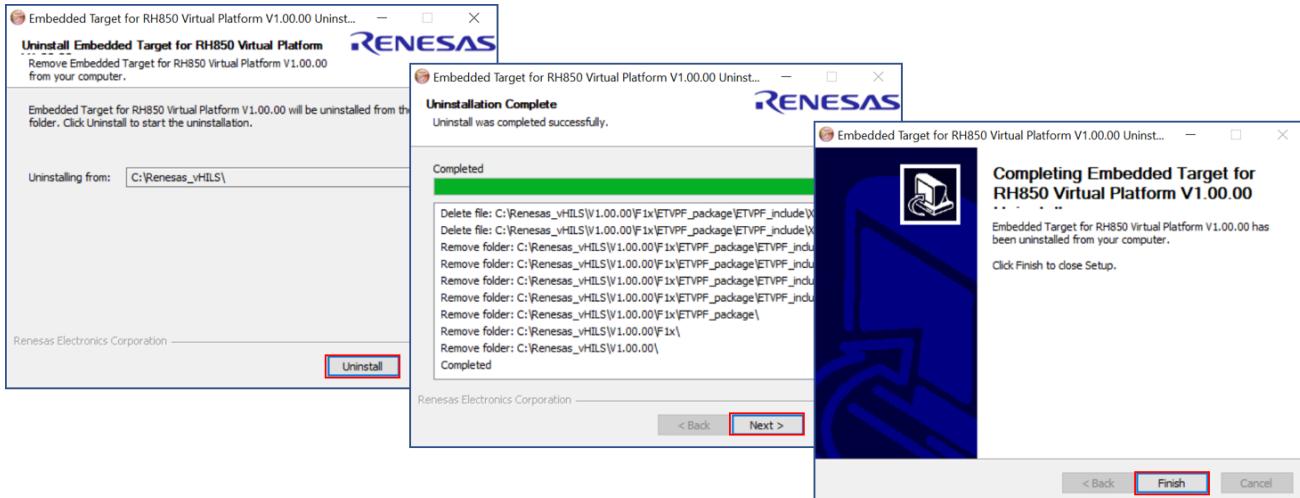


**Figure 2-3 Removing the ET-VPF folder**

- (2) Execute (Uninst\_ETVPF\_<version information>.exe or Uninst\_ETVPF\_U2x\_<version information>.exe) in the ET-VPF installation folder as below steps:
  - **Step 1:** Double click to (Uninst\_ETVPF\_<version information>.exe or Uninst\_ETVPF\_U2x\_<version information>.exe) to start uninstalling ET-VPF package.
  - **Step 2:** Execute step by step to uninstall ET-VPF package as the following figures.



**Figure 2-4 Uninstallation Confirmation Message**



**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.

(3) Delete the license.

Delete the ET-VPF license by license manager included with CS+.

(4) 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** After uninstalling ET-VPF, if you refer ET-VPF peripheral blocks in the Simulink library browser without restarting MATLAB, a warning will be displayed. If you restart MATLAB, the above steps are not necessary.

### 3 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:

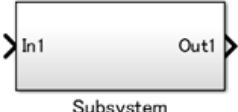
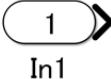
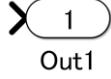
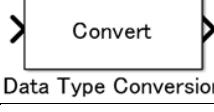
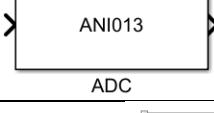
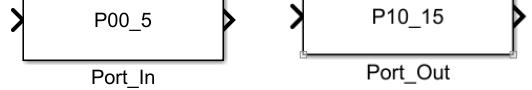
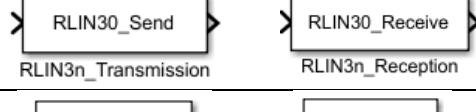
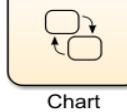
- Set configuration parameters
- Clean existed files, objects.
- Generate source code files.
  - Generate C source files of peripherals (Port, ADC, RLIN3n, TAUD) (using SC to generate them).
  - Generate source files of target device.
  - Generate C source files.
  - Generate make file and config file.
  - 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.
- For communication peripherals, data will be sent or received at every sampling time

About S-Function block of peripherals (hereafter referred 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 referred 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 referred to as usable blocks).

**Table 3-1 List of usable blocks**

No	Usable blocks	Configurable blocks
1	Subsystem	
2	Import (*1)	
3	Outport	
4	Mux	
5	Demux	
6	Data Type Conversion	
7	ADC	
8	PORT	
9	TAUD	
10	RLIN3	
11	RS-CANFD	
12	Chart	

(\*1) This block can be created by right clicking and then dragging the original Import block and then selecting “Duplicate Import” as the Import shadow. This causes confusion and Renesas Electronics does not recommend user to use the Import 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.
    - a) For ADC peripheral: support for Scan Group 1, Multicycle scan mode.
    - b) For Port peripheral: Only can use normal port (PORT) (e.g., PORT0, PORT1... defined in SC).
    - c) 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 U2B: Support use CANFD unit 0 (RSCFD0) with 8 channel, Sending/receiving Data frame, Standard ID, the remote frame, extended ID, receiving filter, and Vehicle Network Toolbox.
    - d) For TAUD peripheral:
 

For F1KM: 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.

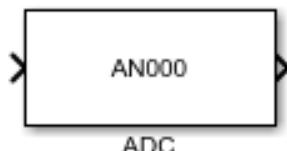
For U2B: Only can use PWM Output, master channel, pulse cycle and duty ratio value settings. About slave channels, and clock setting please set it in SC.

For U2B: support use TAUD 4 units (unit 0, 1, 2, 3) with 16 channel per unit.
    - e) For U2B: support use RLIN3n 7 units (unit 0, 1, 2, 3, 4, 5, 23) with 1 channel per unit.
  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/U2B, only use the Port, ADC, CAN, TAUD, RLIN3n peripherals
  5. ETVPF is implemented for device series with the max pin, if using devices series with fewer pins must check the hardware specification to setting the correct peripherals.
  6. To avoid waiting for the UART send function to finish transferring 8-bytes data (one byte at a time), the user can update the buffer setting.

#### 3.2.1 ADC peripheral

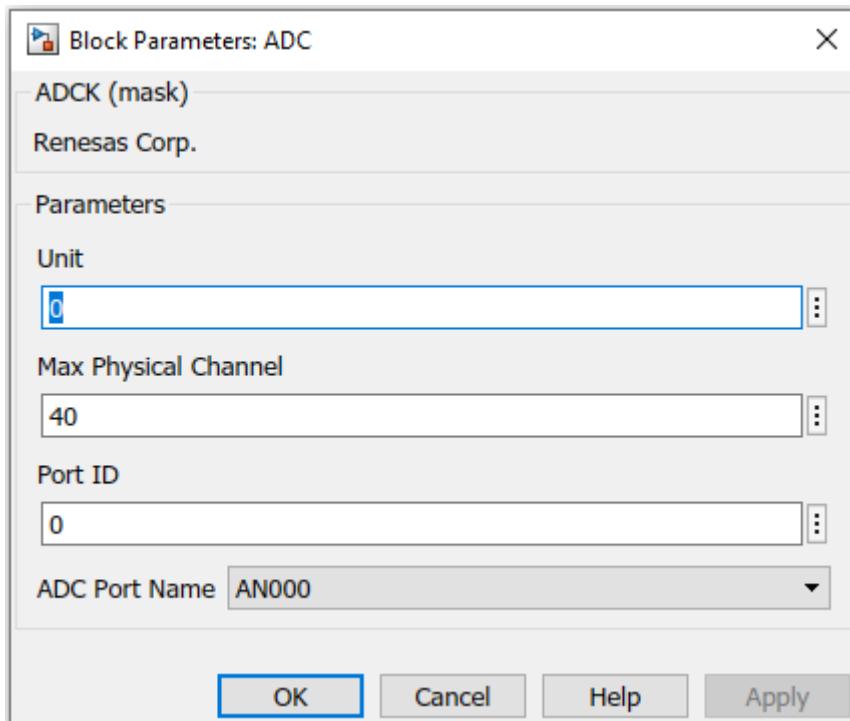
The following describes about the main features of ADC peripheral.

- This block receives the data (double) as input from MATLAB, then it transfers data to User Algorithm (double value).



**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.

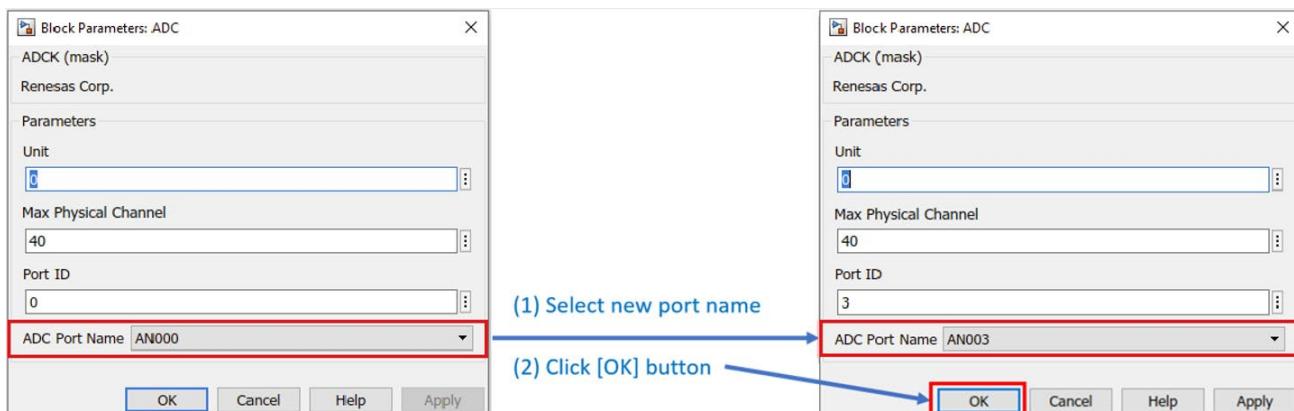


**Figure 3-2 User Interface of ADC S-Function block**

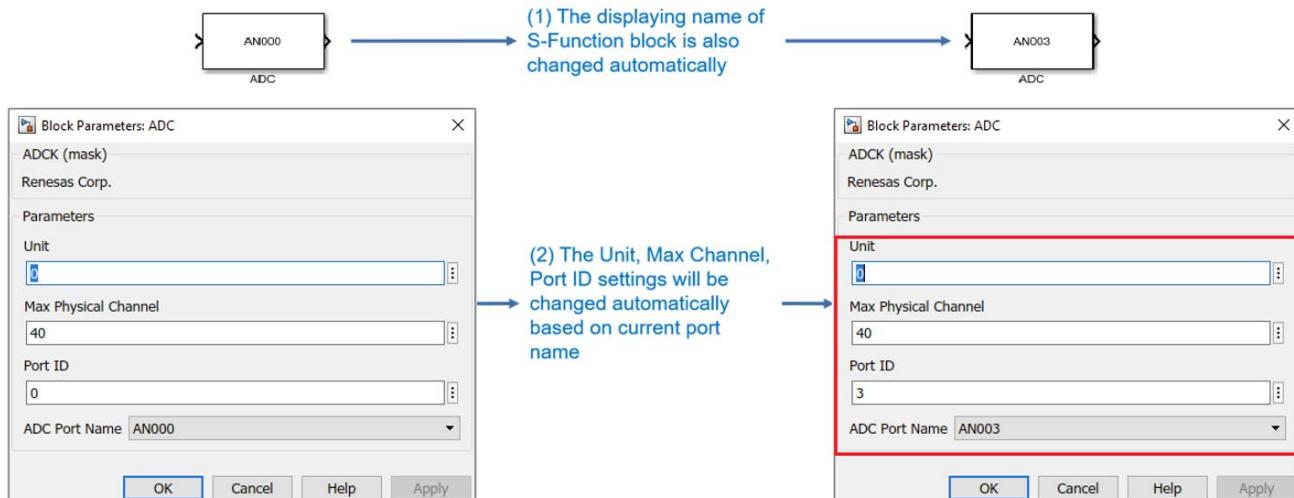
The User Interface of ADC S-Function block includes “ADC Port Name”, “Unit”, “Max Channel” and “Port ID” parameters.

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.



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



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

### 3.2.2 Port peripheral

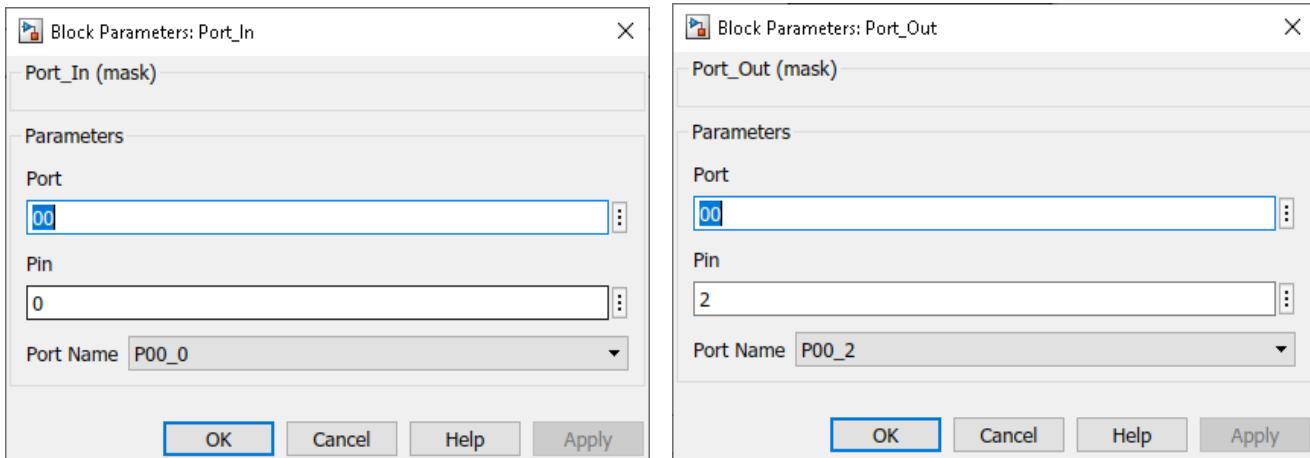
The following describes about the main features of Port peripheral.

- **For Port\_In block:**
  - This block receives the data (boolean) as input from MATLAB, then it transfers data to User Algorithm (boolean value).
- **For Port\_Out block:**
  - This block receives the data (boolean) as input from User Algorithm, then it transfers data to MATLAB (boolean value).



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

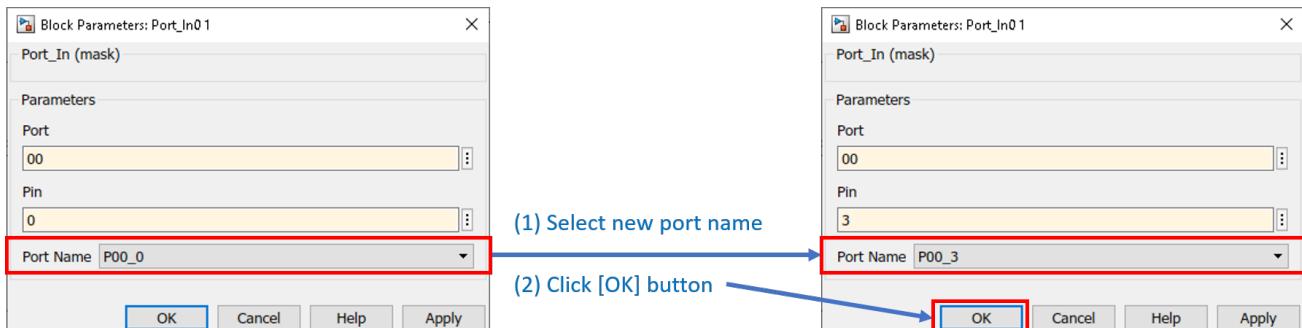
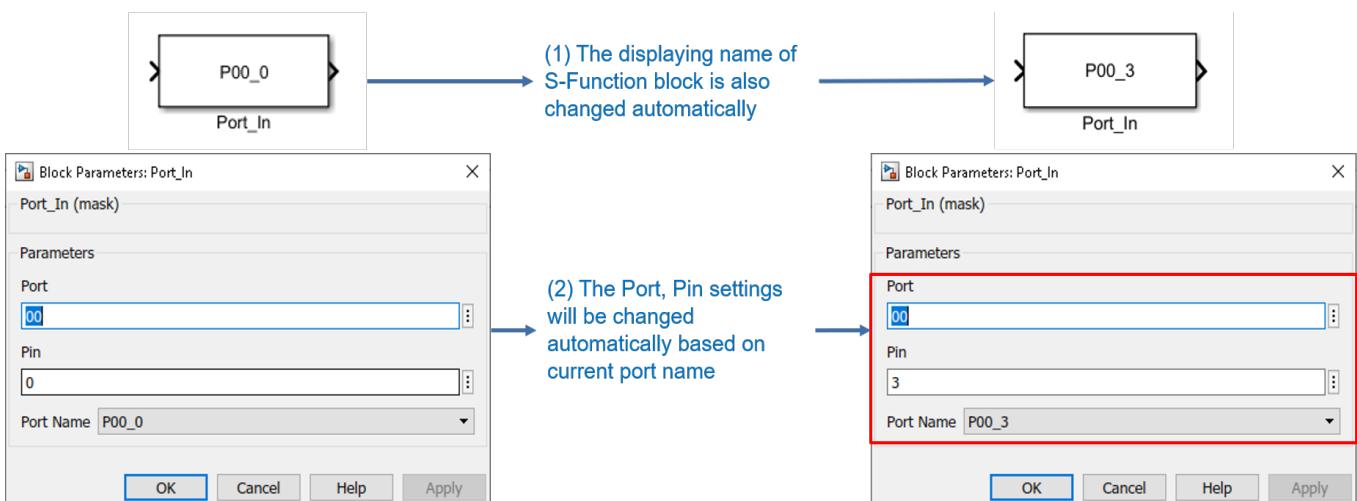
The following describes about the User Interface of Port S-Function block that support user select and change ports easier during model setting. The User Interface also have two types be corresponding to two types of Port S-Function block (Port\_In and Port\_Out).

**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.

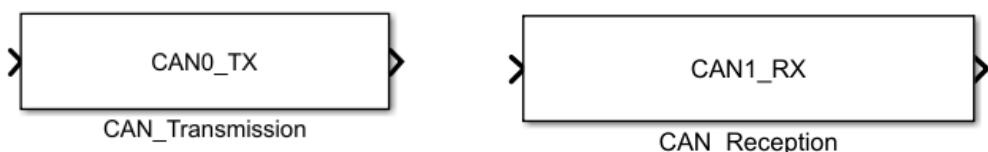
After changing port name, the parameters (Port, Pin) and displaying name of S-Function block will be changed automatically based on the current port name.

**Figure 3-7 Changing port name of Port S-Function block****Figure 3-8 The changing result of Port S-Function block**

### 3.2.3 RS-CANFD peripheral

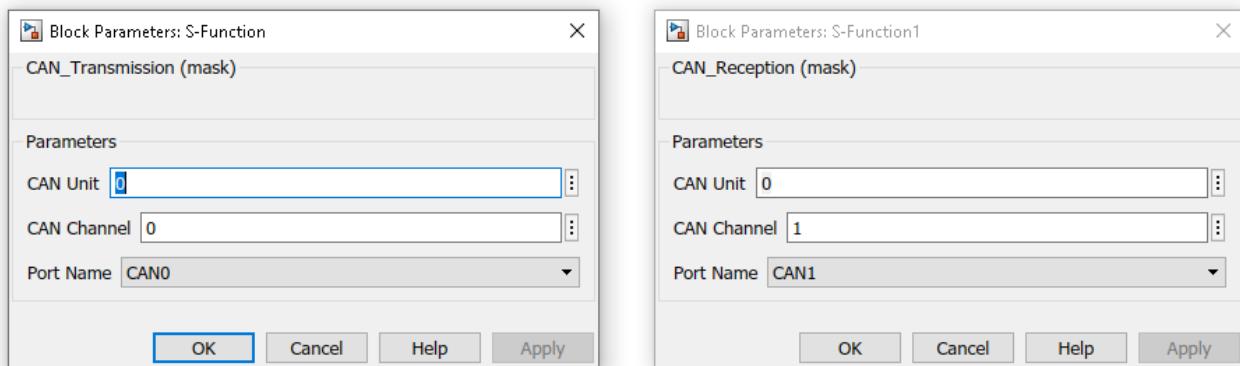
The following describes about the main features of RS-CANFD peripheral.

- **For CAN\_Transmission block:**
  - Get array bytes (dynamic length, 8-bit unsigned integer) from User algorithm, transmit message through RS-CANFD port, then output to MATLAB (bytes array data, 8-bit unsigned integer).
- **For CAN\_Reception block:**
  - Get array bytes (dynamic length, 8-bit unsigned integer) from MATLAB, transmit through RS-CANFD port, output to user algorithm (bytes array data, 8-bit unsigned integer).



**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).

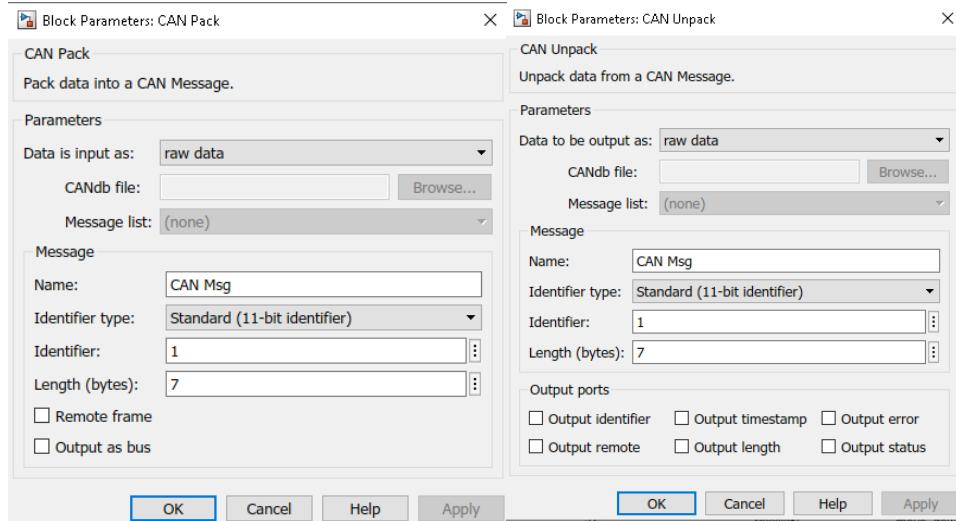


**Figure 3-10 User Interface of RS-CANFD S-Function block**

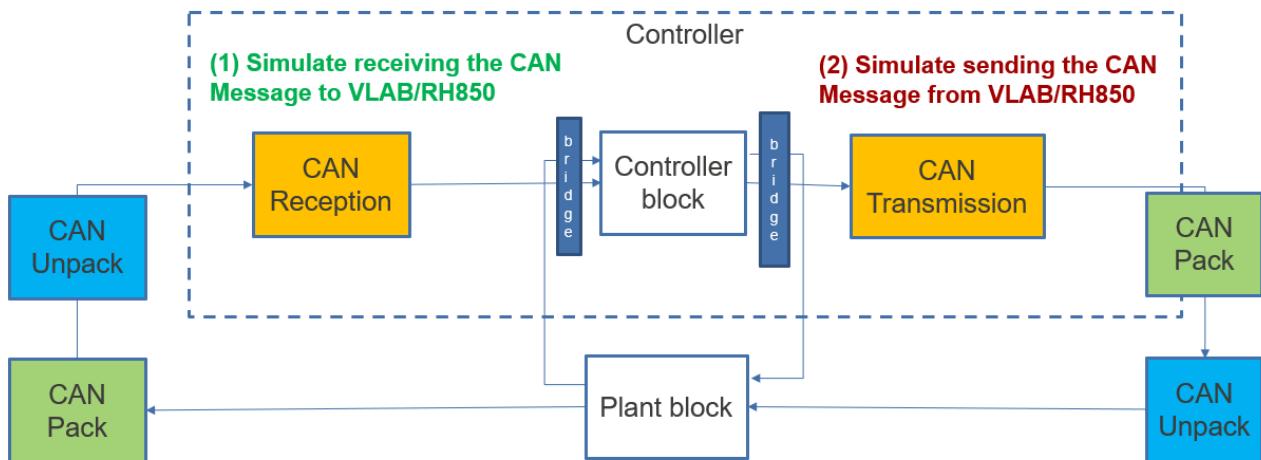
The User Interface of RS-CANFD S-Function block includes “RS-CANFD unit”, “RS-CANFD channel”, “Port Name” parameters. The “RS-CANFD unit” and “RS-CANFD channel” will be generate automatically after changing the “Port Name”.

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.

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

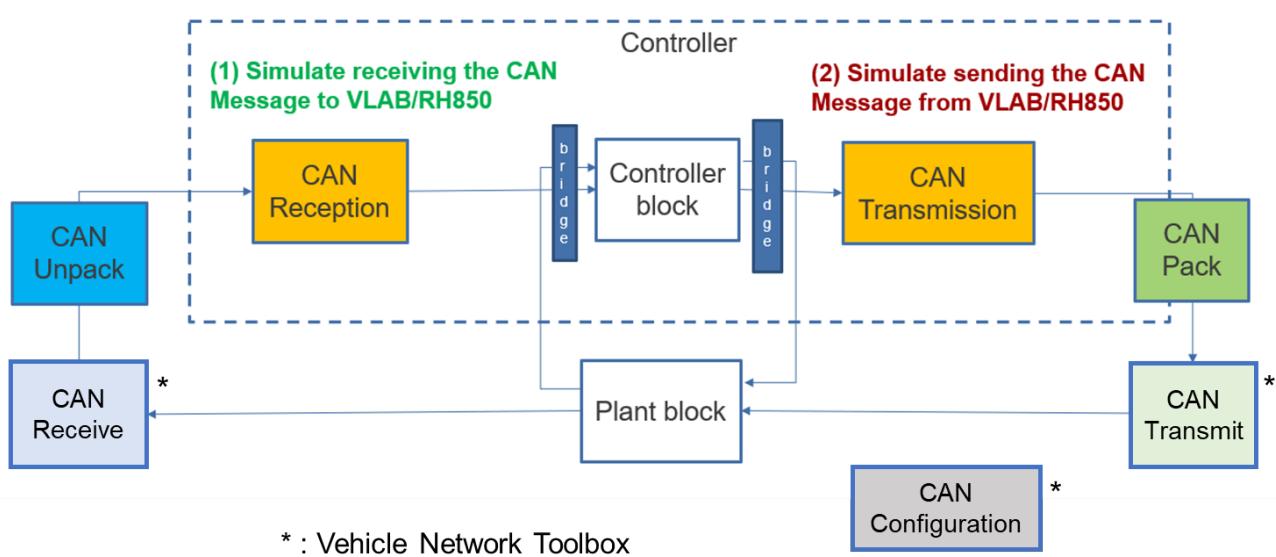
- Model structure between MATLAB and VLAB:

**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.

The CAN Pack/Unpack must be placed on the upper layer of the code generation target.

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



**Figure 3-13 Model structure for using RS-CANFD with Vehicle Network Toolbox**

- Time out for receiving CAN message:

ETVPF prepares a loop to wait for the CAN message during reception. If time out duration is insufficient, CAN message may be missed. User can define number of loop times in below files:

[For RH850/F1KM-S1 Device Series]

ETVPF\_package/ETVPF\_include/Renesas/Common\_files\_F1KM\_S1/include/Define\_Macro\_loop.h

[For RH850/F1KM-S4 Device Series]

ETVPF\_package/ETVPF\_include/Renesas/Common\_files\_F1KM\_S4/include/Define\_Macro\_loop.h

[For RH850/U2B Device Series]

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

To determine the loop times, use the algorithm below:

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

Where:

**Time out:** the wait time for receiving CAN message (s)

**CPUCLK:** the CPUCLK clock (Hz)

**Number of loop instructions:** number of instructions in 1 loop. Default: 15 instructions.

**Table 3-2 The Port Name is used for RS-CANFD**

Device Series	RS-CANFD	Port Name
RH850/F1KM-S1 RH850/F1KM-S4	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
RH850/U2B	CAN0 RX	P02_2
	CAN0 TX	P02_0
	CAN1 RX	P00_1
	CAN1 TX	P00_0
	CAN2 RX	P02_4
	CAN2 TX	P02_1
	CAN3 RX	P00_3
	CAN3 TX	P00_2
	CAN4 RX	P00_5
	CAN4 TX	P00_4
	CAN5 RX	P14_0
	CAN5 TX	P14_2
	CAN6 RX	P00_10
	CAN6 TX	P00_11
	CAN7 RX	P20_4
	CAN7 TX	P20_5

### 3.2.4. RLIN3n peripheral

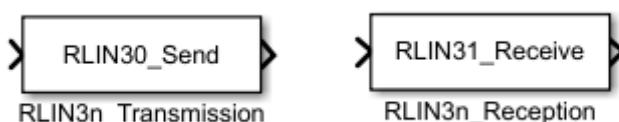
The following descriptions describe about the main features of RLIN3 peripheral.

- **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).

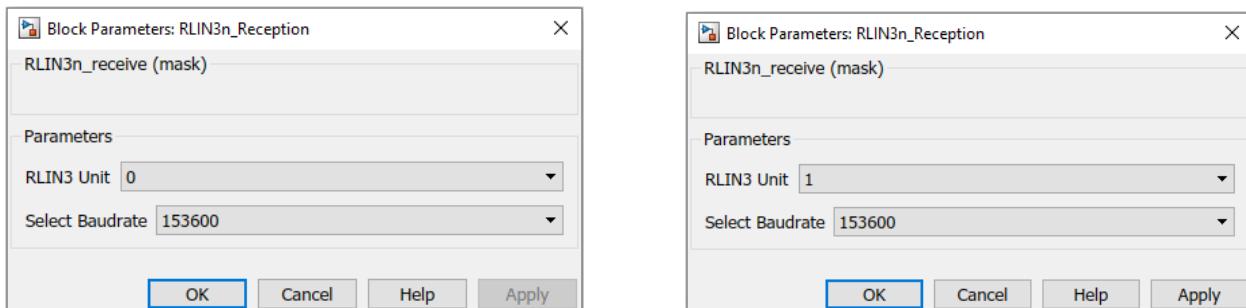
- **For RLIN3n Reception block:**

- 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).



**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 RLIN3 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).

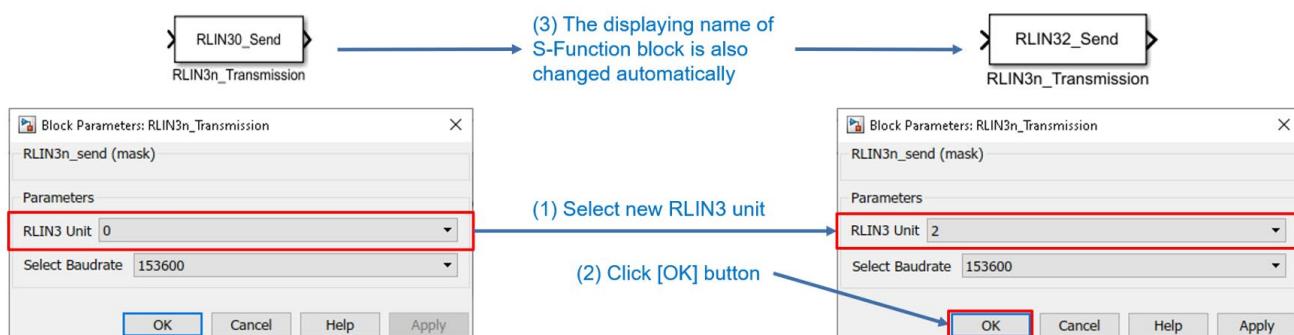


**Figure 3-15 User Interface of RLIN3n S-Function block**

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.



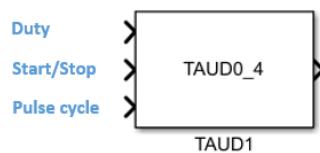
**Figure 3-16 The changing result of RLIN3n S-Function block**

### 3.2.5. TAUD peripheral

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

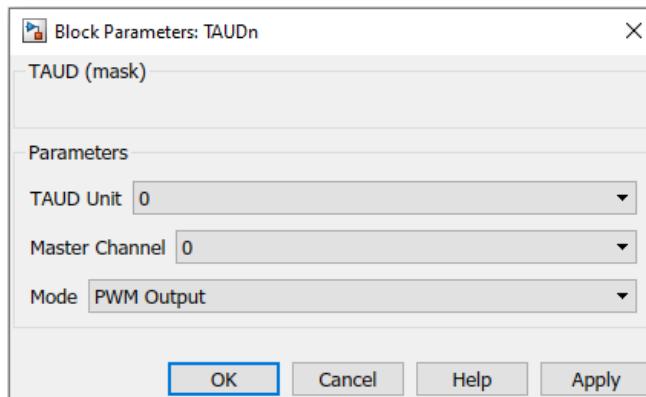
- This block receives three inputs from User algorithm to control output of PWM.
  1. Start/Stop (boolean) corresponds to Start (1) or Stop (0) state of TAUD.
  2. Pulse cycle (double) is required in nanosecond (example: 0.3ns = 0.3 as input).
  3. Duty value (array of double value) is required a decimal number corresponding to percentage (example: 50% -> 0.5 as input).
- Output of this block is an array of data signals (master channel corresponds to the first output of array signal and other outputs corresponds to each slave channel was selected).
- Slave channels select by User in SC.
- PWM output: generate PWM data signal then send data to MATLAB (boolean).
- Triangle PWM output: generate Triangle PWM data signal then send data to MATLAB (boolean).

- Remarks
1. For RH850/U2B does not support for Triangle PWM Output.
  2. For RH850/U2B, due to the TAUD module using the common clock with OSTM (CLKC\_HSB) and having a default value of 20 MHz, if a different clock value is used, the pulse cycle needs to be configured to obtain correct results. The conversion method is described below:
    - Freq(MHz) = <freq\_value> -> pulse cycle(ms) = <(pulse\_value\*20)/freq\_value>
    - For example, for a desired pulse cycle of 200 (ms):
      - Freq = 20MHz -> pulse cycle = (200\*20)/20 = 200ms
      - Freq = 80MHz -> pulse cycle = (200\*20)/80 = 50ms



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

The 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.

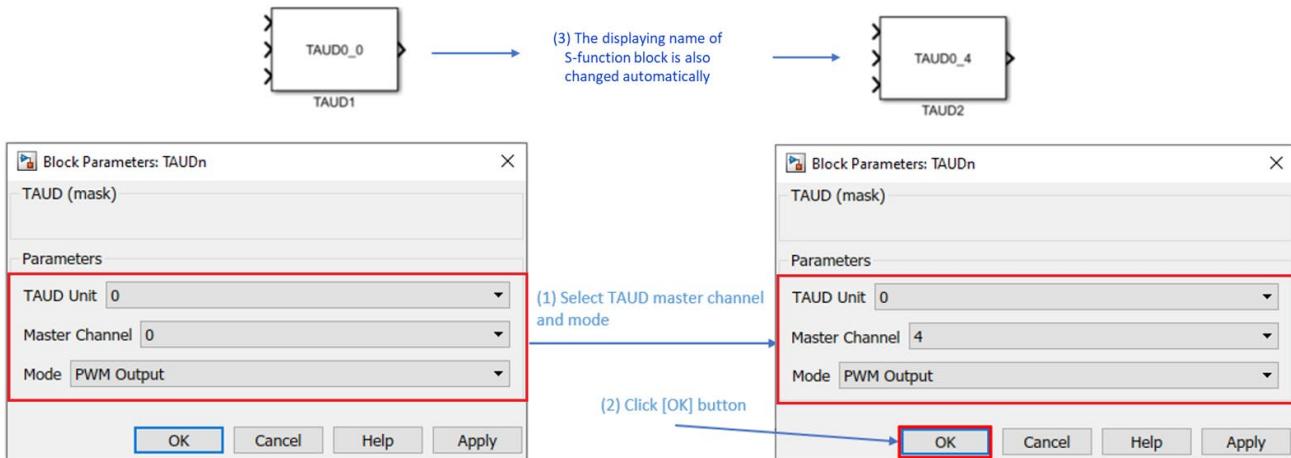


**Figure 3-18 User Interface of TAUD S-Function block**

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

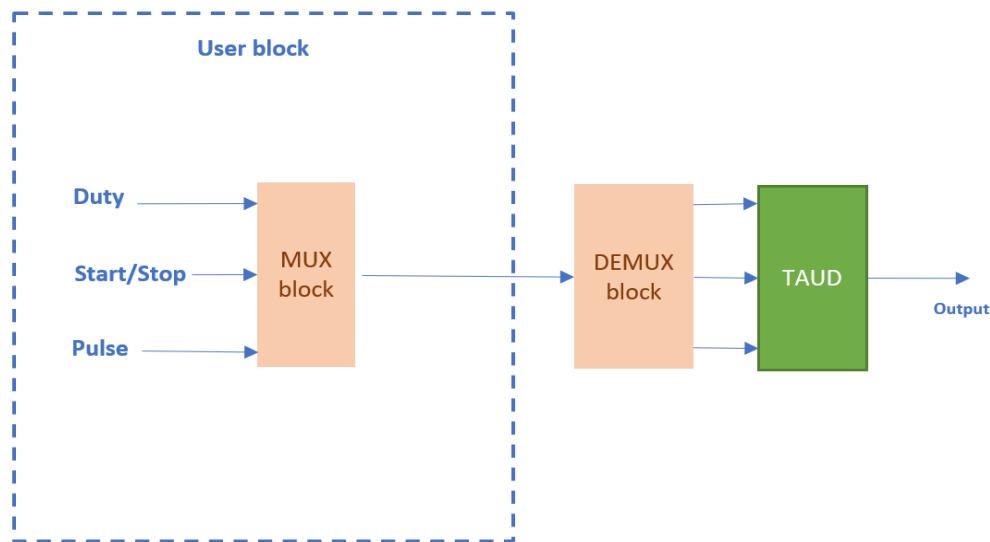
The purpose of these parameters is to specify the target port name of TAUD S-Function block. When building the model, these parameters will generate at the same time as the input data.

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**

- **Model 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.

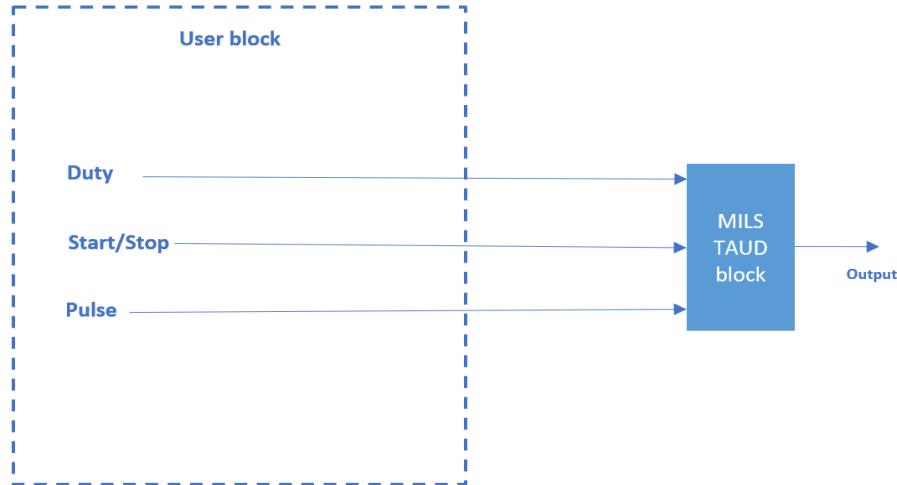
- **Support Model in the Loop Simulation (MILS) for TAUD block:**

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.

Each 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**

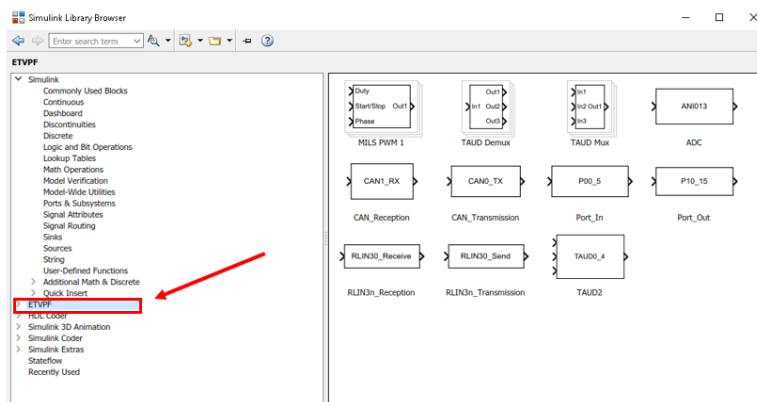
### 3.3 Executing virtual Hardware in the Loop Simulation

The following describes how to generate a vHILS environment necessary for virtual Hardware in the Loop Simulation (hereafter referred 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.

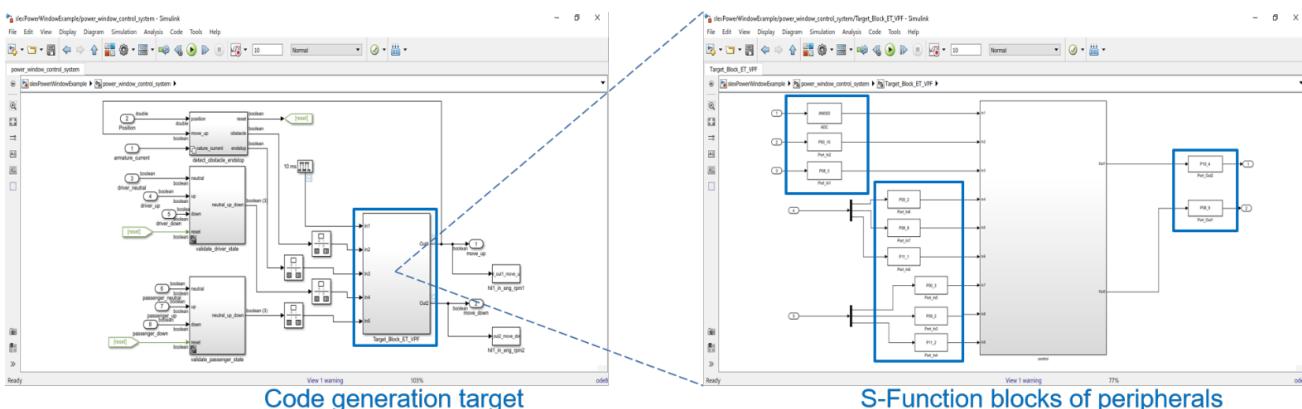
- The ETVPF package is in MATLAB Simulink Library Browser, which contained the S-function blocks of peripherals.



**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.



**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 3-3 Code generation target of sample model**

Sample model name	Code generation target	Block type
slexPowerWindowExample.slx	Target_Block_ET_VPF	Subsystem block

**Table 3-4 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<unit>_RX	CAN_Transmission_<Number> CAN_Reception_<Number>	
	RLIN3n	RLIN3<RLIN3 unit>_Send RLIN3<RLIN3 unit>_Receive	RLIN3n_Transmission_<Number> 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/U2B device series.

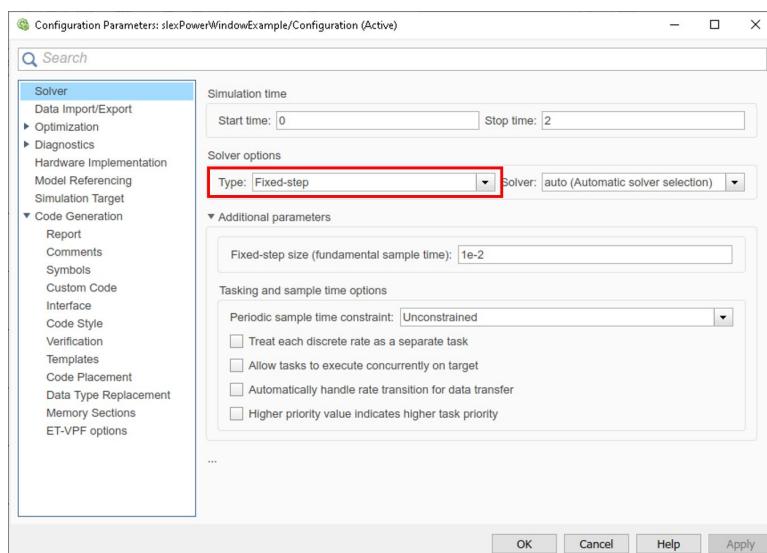
### 3.3.2 Setting configuration parameters

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 or R2020a.
- (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.

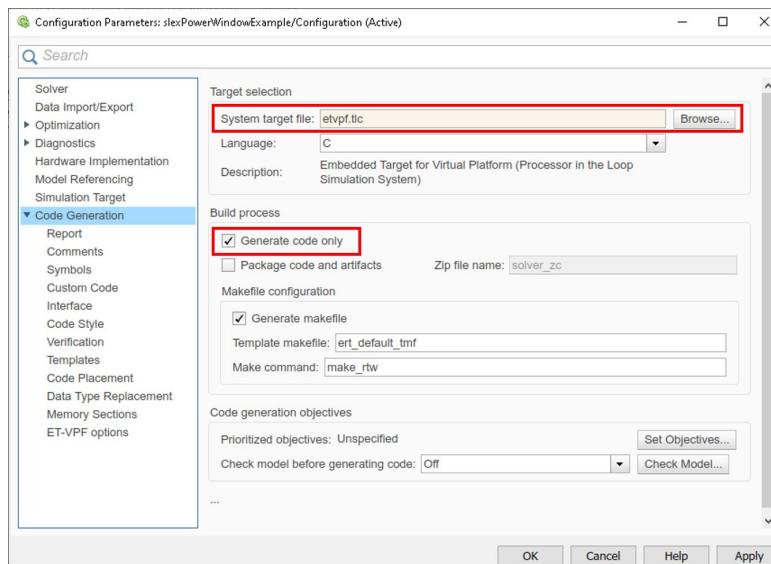
➤ **Step 1: Select [Solver] -> [Type] is “Fixed-step”.**



**Figure 3-24 [Solver] settings**

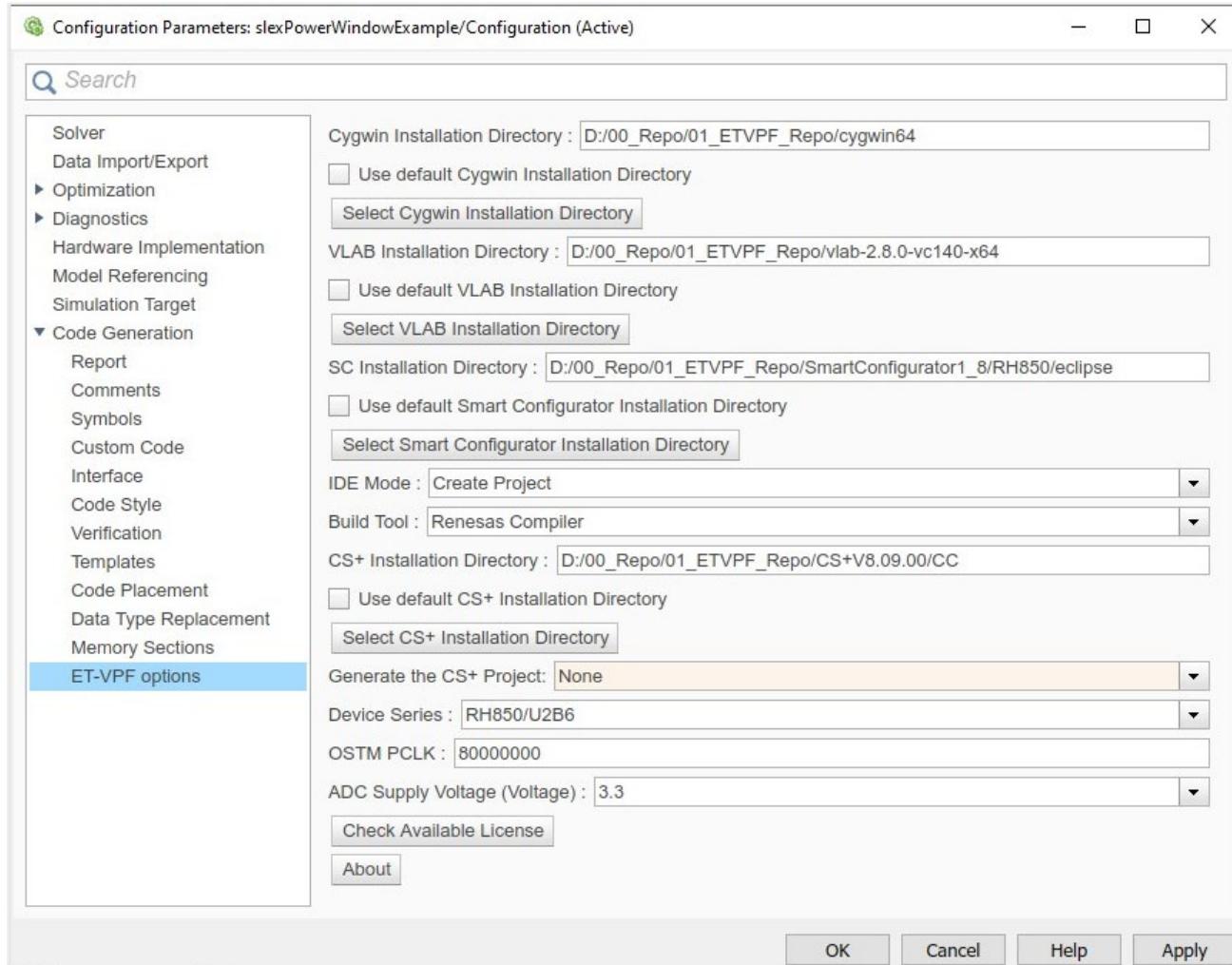
➤ **Step 2: Setting for [Code Generation].**

- Select [System target file] is “etvpf.tlc”.
- Select [Generate code only].



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

- **Step 3:** In the [ET-VPF options], select necessary settings that described in the **Table 3-5 ET-VPF Options**.



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

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

**Table 3-5 ET-VPF Options**

Item name	Description	
Cygwin Installation Directory *1 *15	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 *15	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 *15	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 *15	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. It is "C:/Program Files (x86)/Renesas Electronics/CS+/CC".	
[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 3-6 ET-VPF Options (Cont.)**

Item name	Description	
Generate the CS+ Project	Specifies the generating the CS+ project for standalone build.	
	None	The default value of Generate the CS+ project. When it is selected, the CS+ project files will not be generated. The source code is compiled by makefile.
	Generate the CS+ project only	The CS+ project files will be generated by ET-VPF and the source code is compiled by CS+.
Device Series *11	Generate the CS+ project and automatically build	The CS+ project files will be generated by ET-VPF and the source code is compiled by CS+. Then, the CS+ project files will be automatically built.
	Selects the series name of the microcontroller being used.	
	<Device Series Name> *12	The supported Device Series described in <b>Table 1-1 Supported devices</b>
OSTM PCLK *13	N/A	Default value of Device Series.
	The OS Timer value corresponding to each device series.	
	Select the supply voltage for Analog/Digital-Converter (Volt)	
ADC Supply Voltage (Voltage)	3.3	The default value (3.3 Volt) of ADC Supply Voltage.
	5.0	The supported 5 Volt value for ADC Supply Voltage.
[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.

\*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.

- \*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... For RH850/F1KM, the OSTM PCLK is derived from the CPUCLK\_L clock.

User can modify the CPUCLK\_L clock by referring to section 3.3.3.1 Generating the peripherals' source code by SC. However, please assure that the update is same as [OSTM PCLK] value in ETVPF Options GUI.

The definition of the CPUCLK\_L clock depends on the Device Series. For details regarding this CPUCLK\_L clock, please refer to the clock controller configurator in the Hardware Manual.

\*14... For RH850/U2B, the OSTM PCLK is derived from the CLKC\_HSB clock.

User can modify the CLKC\_HSB clock by referring to section 3.3.3.1 Generating the peripherals' source code by SC. However, please assure that the update is same as [OSTM PCLK] value in ETVPF Options GUI.

The definition of the CLKC\_HSB clock depends on the Device Series. For details regarding this CLKC\_HSB clock, please refer to the clock controller configurator in the Hardware Manual.

\*15... 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	O	-
'	-	-	-	-
"	-	O	O	-
;	-	-	-	-
space	O	O	O	-

\* O : Can use

-: Cannot use

\*16... When the "Generate the CS+ project only" or "Generate the CS+ project and automatically build" is selected, the "Renesas Compiler" will be set and cannot be selected.

➤ **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.

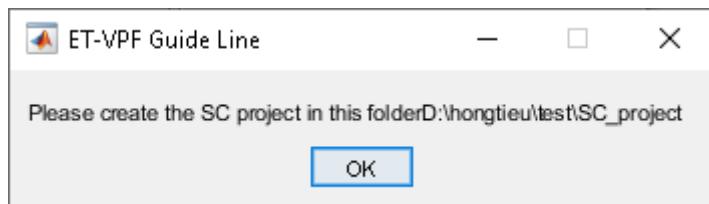
**Table 3-8 Provided command**

Command name	Description
run_vlab	Generates a vHILS environment and executes vHILS automatically

- (4) Select the Code generation target on model.
- (5) 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]
```

- (6) 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.

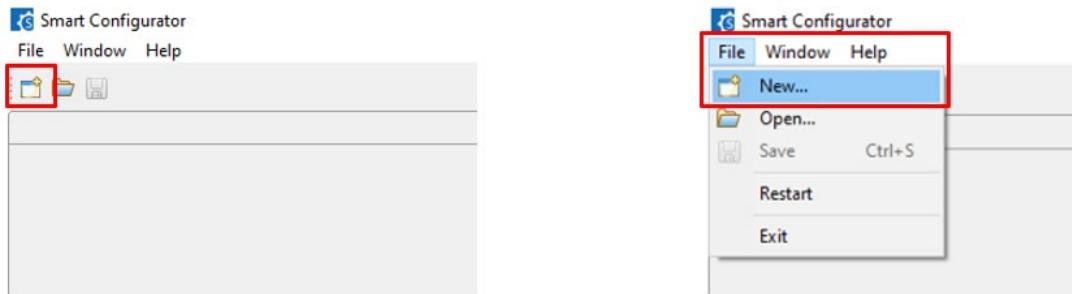


**Figure 3-27 Message box notify the full path of SC\_project**

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

(7) After that, the SC Window will be displayed. User can configure detail settings and generate peripherals' source code as below:

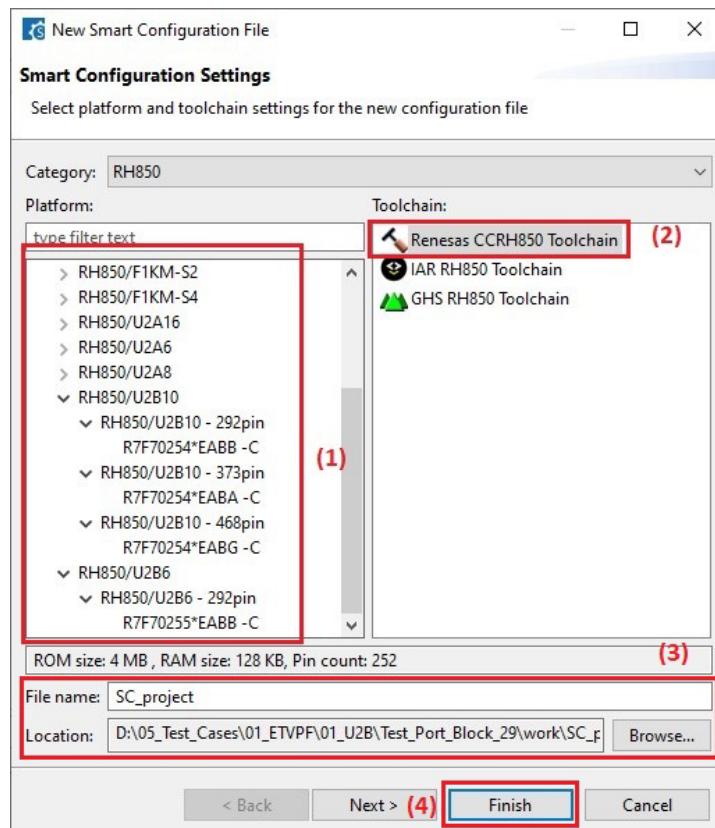
- **Step 1:** Select [New Configuration File] or [File] -> [New...] to create new SC project.



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

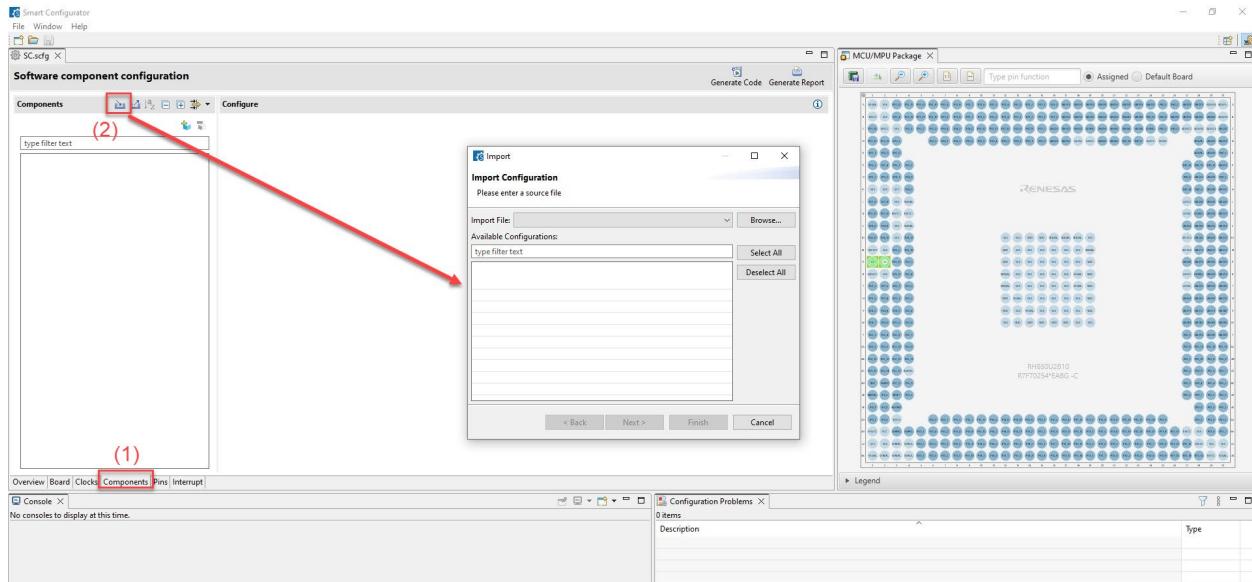
- **Step 2:** Select platform and toolchain settings:

- Select Device name in [Device] setting (1) (**Note:** Currently, only RH850/F1KM and RH850/U2B device series with any pin type is 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).



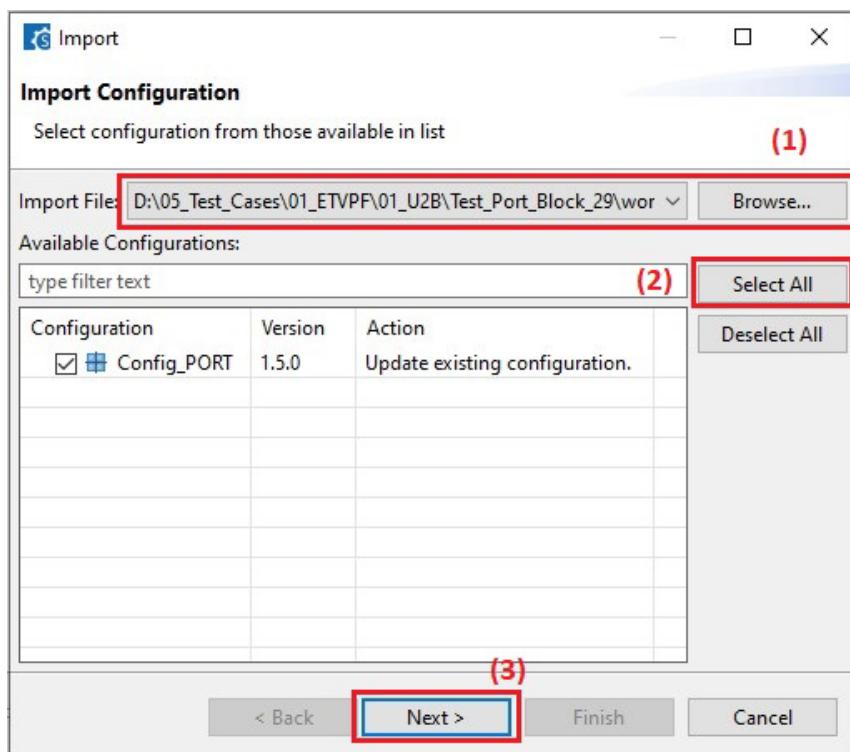
**Figure 3-29 Selecting SC settings**

- **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.



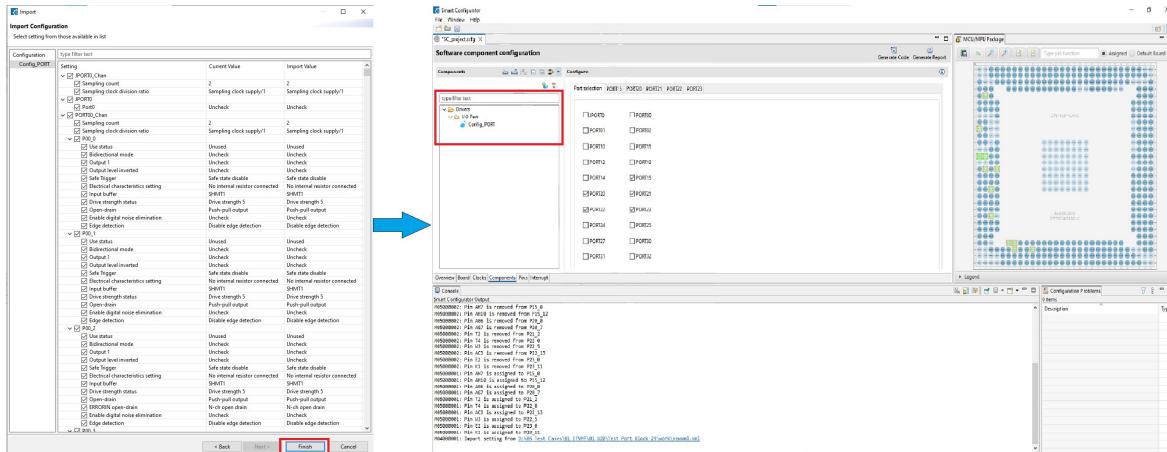
**Figure 3-30 Importing Configuration**

- 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).



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

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



**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.

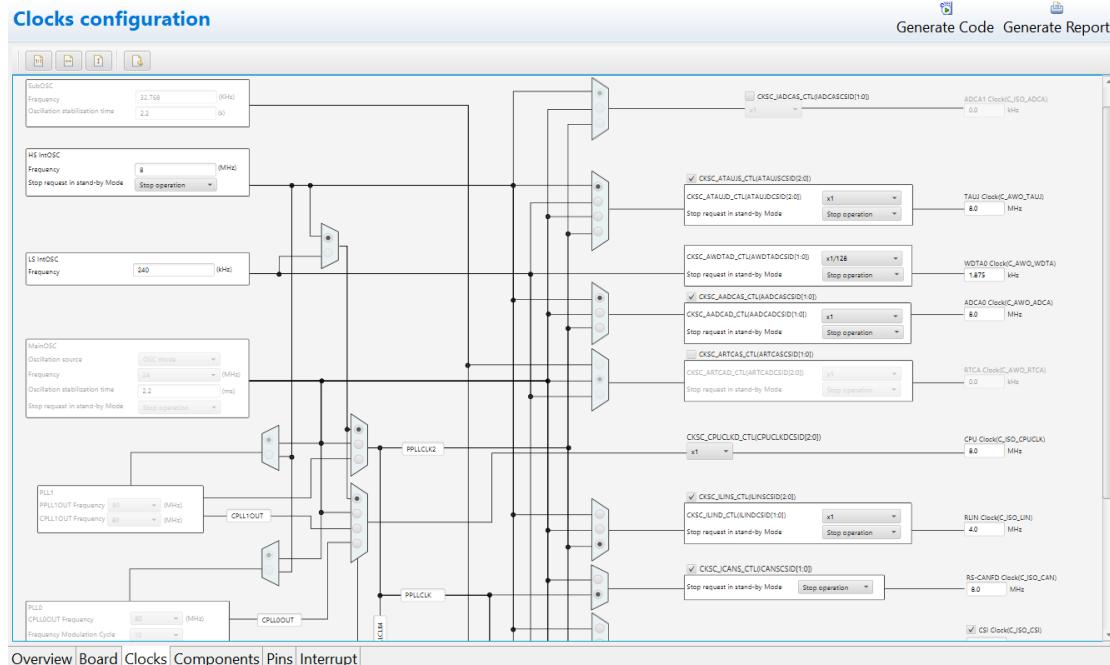
3. For TAUD, user must select the clock source and slave channels in SC GUI.

#### ➤ Step 3.5: Update clock (optional)

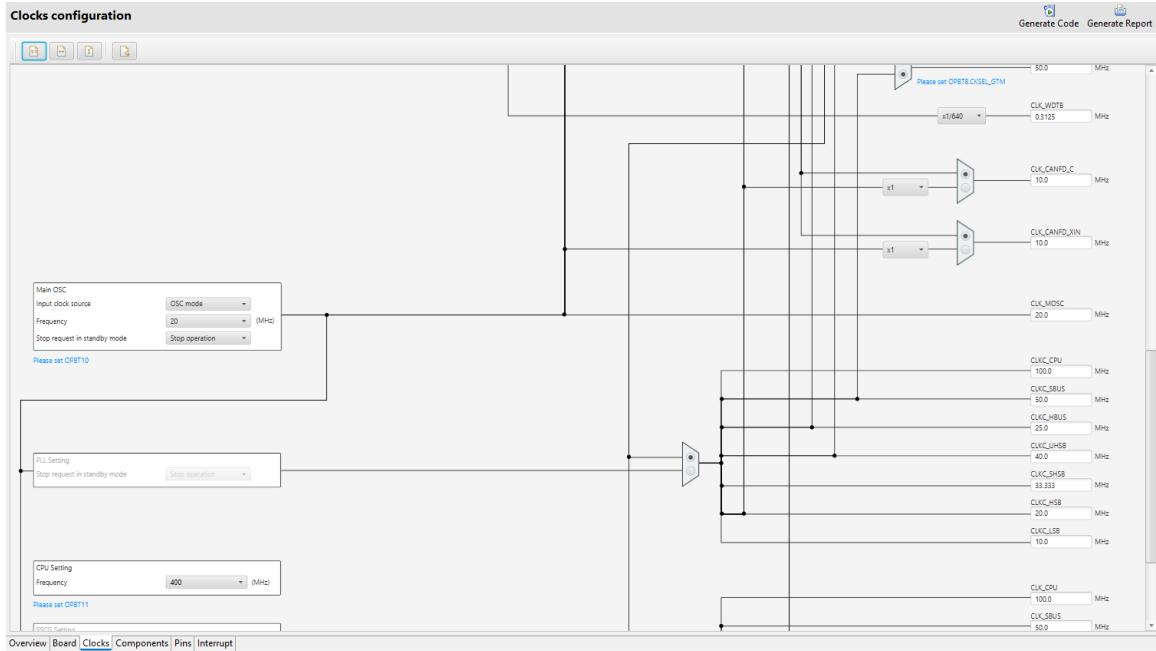
[For RH850/F1KM, RH850/U2B Device Series]

To update the clock, go to the Clocks tab in the SC GUI. Then, user can make modifications in the Clocks configuration panel.

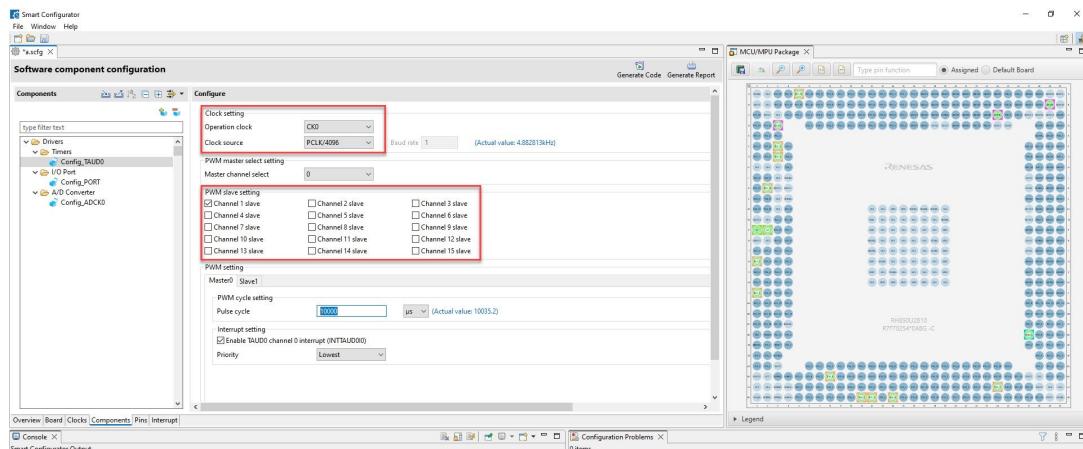
#### Clocks configuration

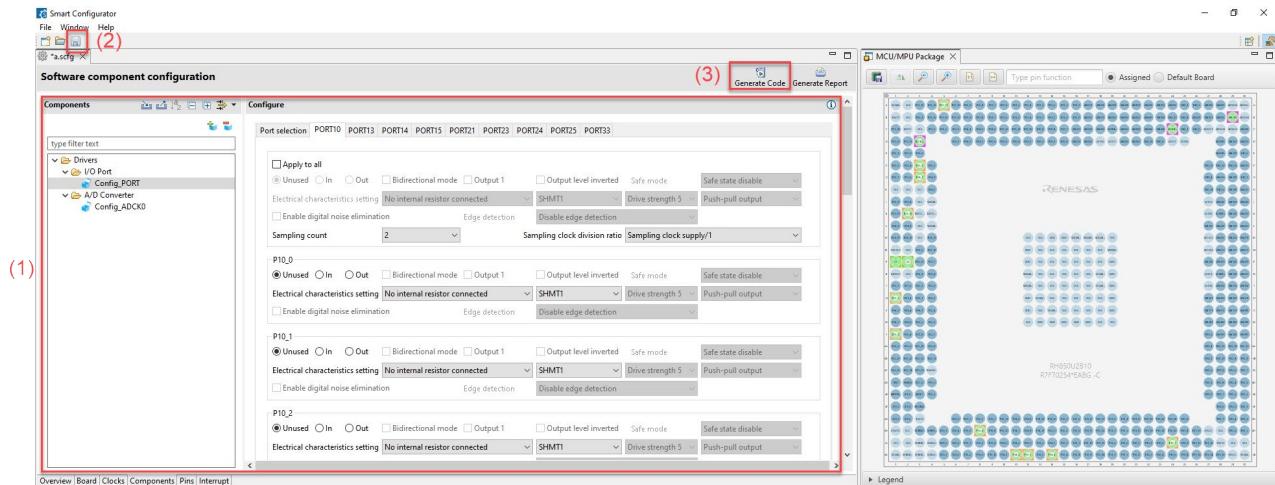


**Figure 3-33 Clocks configuration panel for F1KM**

**Figure 3-34 Clocks configuration panel for U2B**

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

**Figure 3-35 Setting slave channels and clock source for TAUD**



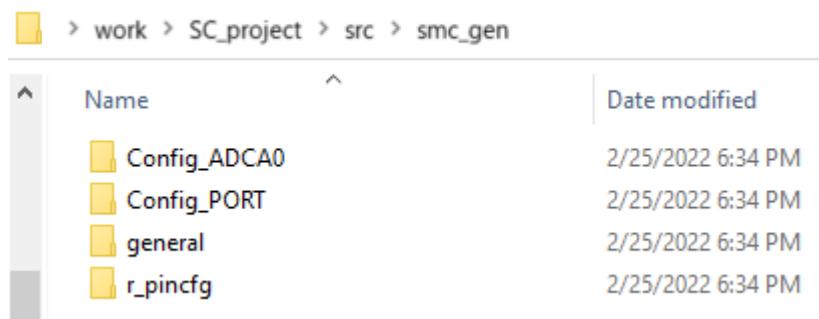
**Figure 3-36 Selecting detail settings and generating peripherals' source code**

#### 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.

(8) The source code of peripherals will be generated under SC\_project folder.



**Figure 3-37 The peripherals' source code generated under SC\_project**

### 3.3.3.2 Generating the target vHILS environment

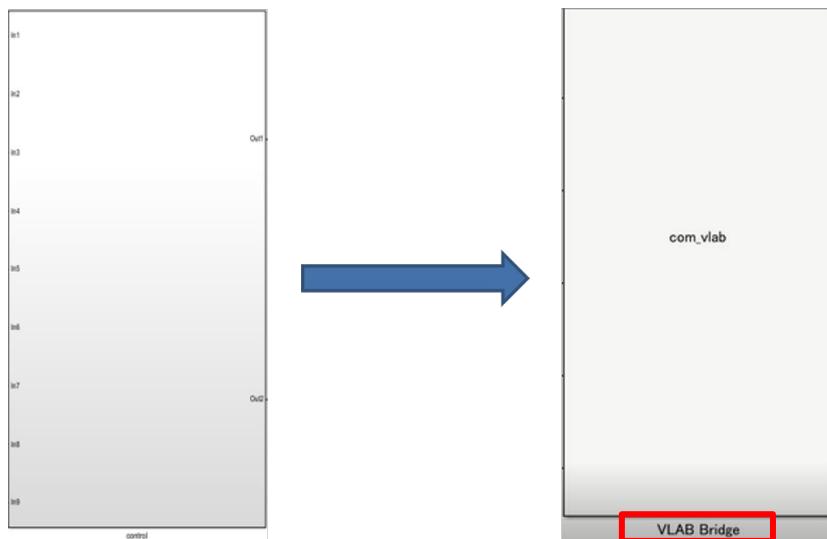
- (9) The target vHILS environment will be generated in the working directory (the location which contains target model).

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
- The CS+ project
- 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 3-38 Example of replacing to the block for PIL sequential execution**

### 3.3.4 Compiling generated source code

- (10-a) After that, when the [Generate the CS+ Project] is set to “None”, the C source files, and source files of the target device (with the extension is \*.c, \*.asm) will be compiled to object files (with the extension is \*.o) via Cygwin.

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

```

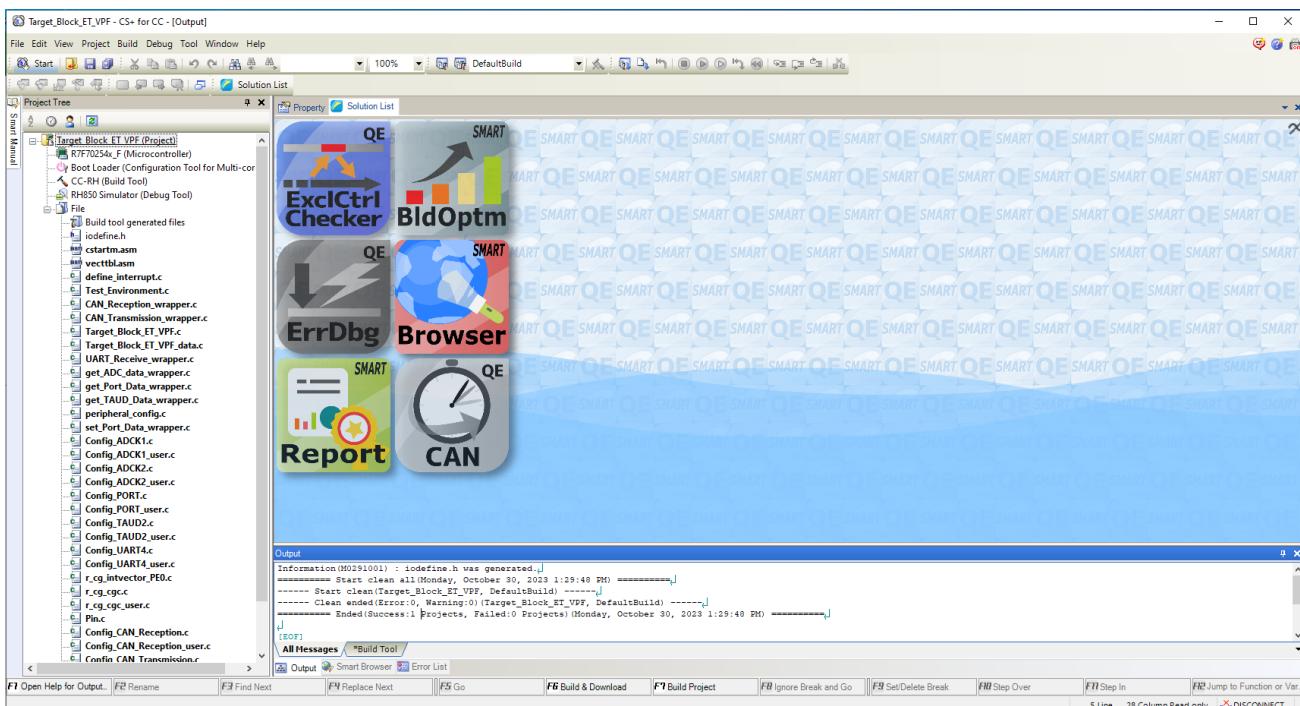
Command Window
*** Finished generating files ***
*** Build and run co-simulation ***
*****
make[1]: Entering directory '/cygdrive/d/SVN_repo/01_Working/03_Worksplace/ETVPF_2023/U5L_U2B/CD_U2B/ETVPF'
Makefile:109: D:/SVN_repo/01_Working/03_Workspace/ETVPF_2023/U5L_U2B/CD_U2B/ETVPF_package_2_13/Sample_mod...
*** Create target dependency definitions done. ***
*****
Compiling etvpf_main.c ...
Generating etvpf_main.o ...
D:/00_Repo/02_ETVPF_Toolbox/CS+V8.09.00/CC/CC-RH/V2.05.00/bin/ccrh.exe -Xcpu=g4mh -g -c -g_line -Xmsg_lan...
Done ...
*****
Compiling D:/SVN_repo/01_Working/03_Workspace/ETVPF_2023/U5L_U2B/CD_U2B/ETVPF_package_2_13/ETVPF_include/...
Generating define_interrupt.o ...
D:/00_Repo/02_ETVPF_Toolbox/CS+V8.09.00/CC/CC-RH/V2.05.00/bin/ccrh.exe -Xcpu=g4mh -g -c -g_line -Xmsg_lan...
Done ...
*****

```

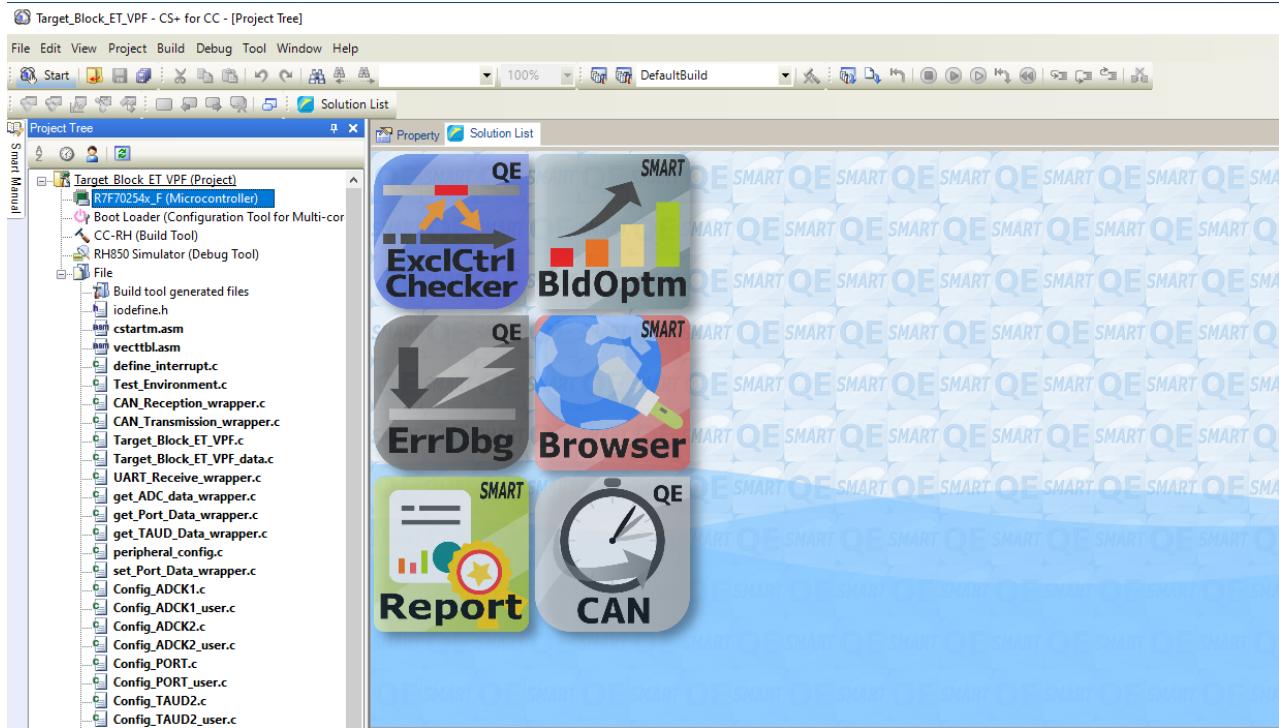
**Figure 3-39 Compiling is displayed on MATLAB Command Window**

The following explains generating and building a CS+ project when [Create Project] is set for [IDE Mode] in [ET-VPF options] of the Configuration Parameters dialog box.

- (10-b) When the [Generate the CS+ Project] is set to “Generate the CS+ project only”, ET-VPF will support automatically generating a CS+ project.



**Figure 3-40 Generating the CS+ project**



**Figure 3-41 Checking the target device**

The following steps explain the necessary setting of CS+ and building of a CS+ project.

- **Step 1:** Make option settings for a build tool (such as compiler and assembler) in the Property panel of CS+. The target device name is set same as the settings in SC (refer to section 3.3.3.1 Generating the peripherals' source code by SC). Note that the default settings of CS+ should be changed in the following cases.

[When using RH850/F1KM-S1 or RH850/F1KM-S4 with Renesas Compiler]

Select [Build Tool] in the CS+ project tree and set [Software Calculating] for [Compile Options] - [Output Code] - [Floating-point calculating type].

Remark	User can change the other settings on CS+ window then build the CS+ project. But Renesas Electronics does not guarantee if the user's settings of CS+ is inappropriate.
--------	---

- **Step 2:** Save the CS+ project.
- **Step 3:** Select [Build Project] from the [Build] menu of CS+.

Remark	For details on generating the load module, refer to "Build" of CS+ User's manual.
--------	---

- **Step 4:** Close the CS+.

- (10-c) When the [Generate the CS+ Project] is set to “Generate the CS+ project and automatically build”, generation and building of a CS+ project are carried out automatically as a series of the operations for generating a test environment. After completed building, the CS+ window is also automatically closed. Accordingly, an explicit operation by the user is unnecessary.

#### Remark

For the “Generate the CS+ project and automatically build”, the default settings of CS+ will be used when automatically building, the user cannot change the settings. Therefore, Renesas Electronics does not guarantee if the default settings of CS+ is in appropriate with use case.

- (11) 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

- (12) After the compiling completed, the VLAB is opened. The RH850 Virtual Platform of RH850/F1KM and RH850/U2B toolbox, “target.out” and source files in the working directory are loaded. The vHILS execution start on both Simulink model and VLAB. Transmission and reception protocol between MATLAB and VLAB is periodically after sampling time.

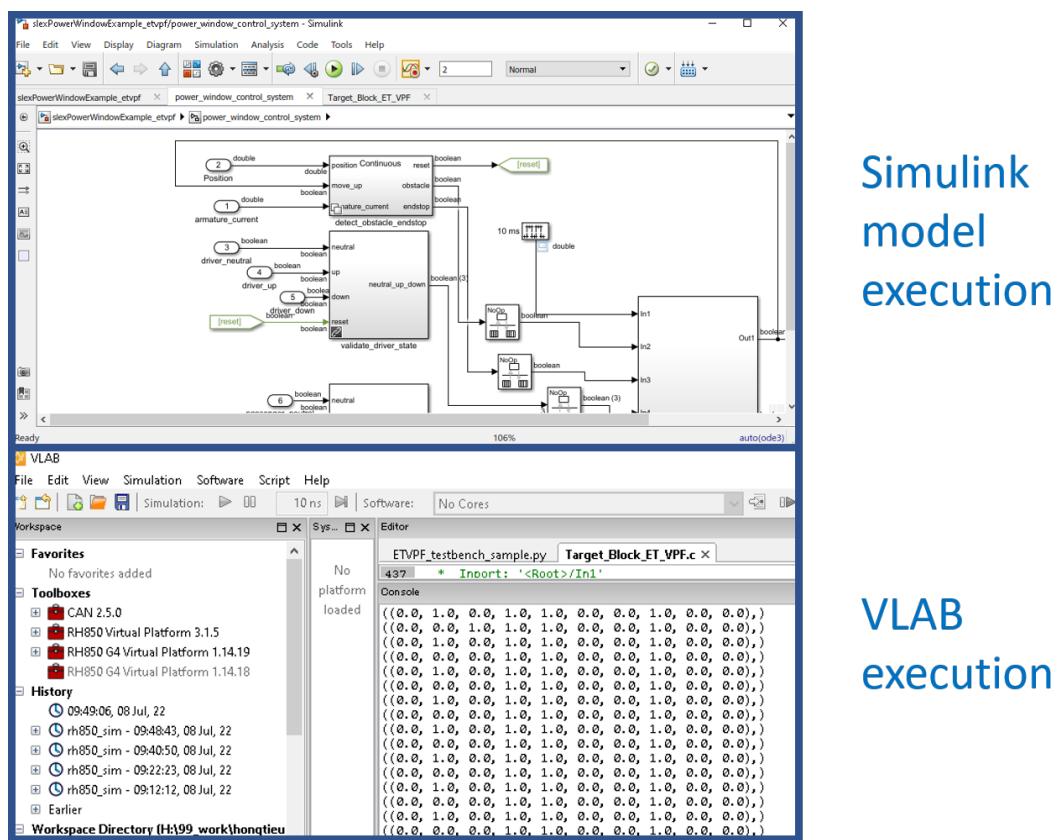
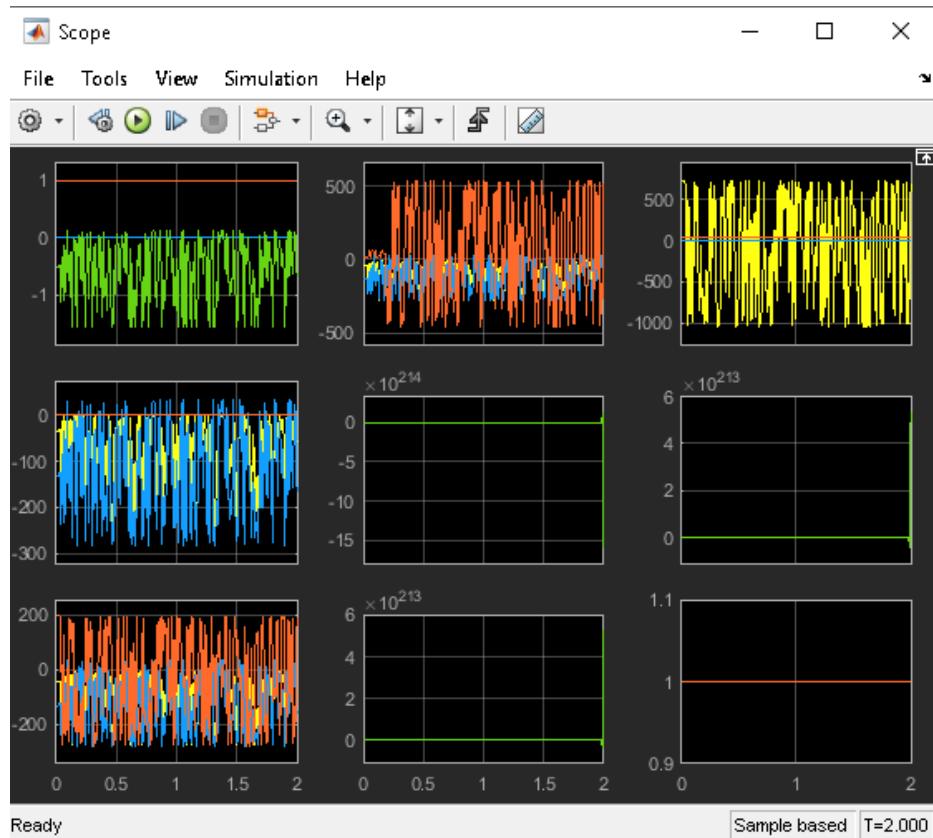


Figure 3-42 The vHILS execution

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

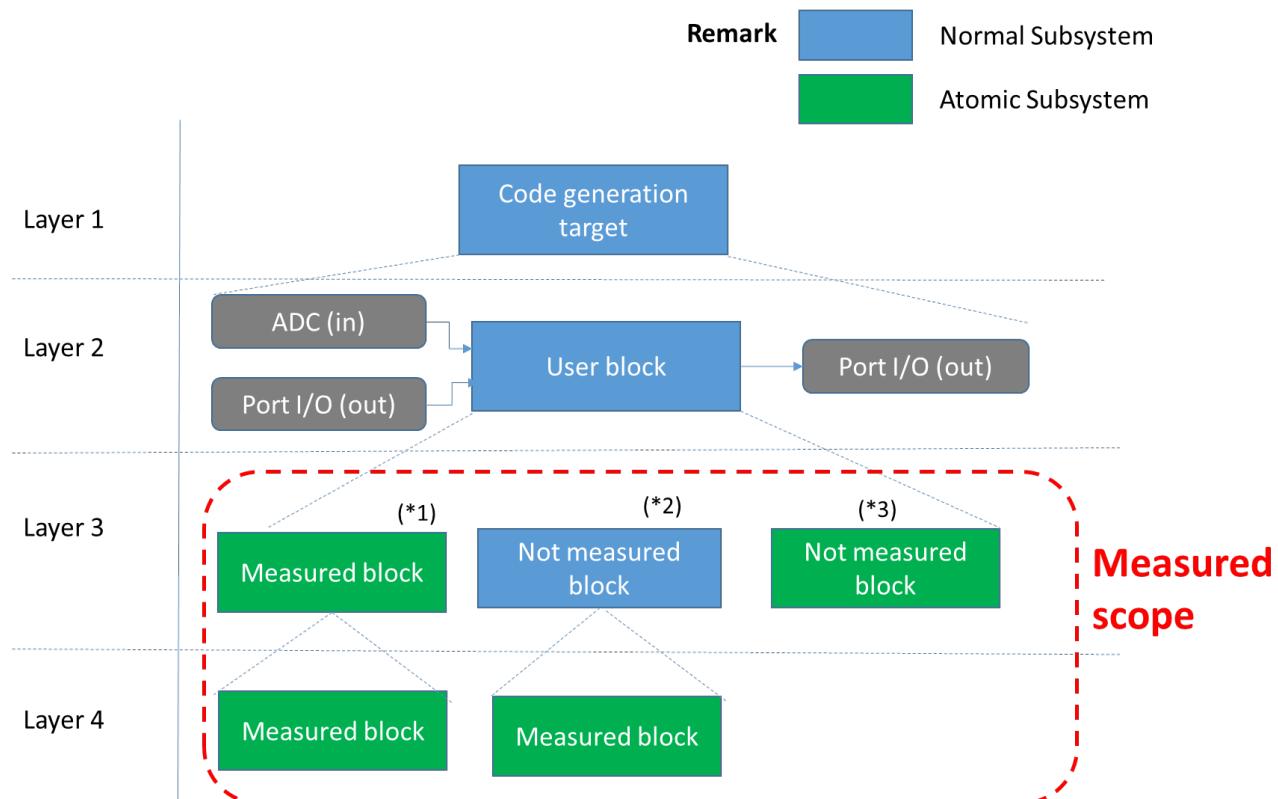


**Figure 3-43 The result of scope**

### 3.4 Time measurement

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

#### 3.4.1 Structure of Simulink Model for measurement



**Figure 3-44 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:

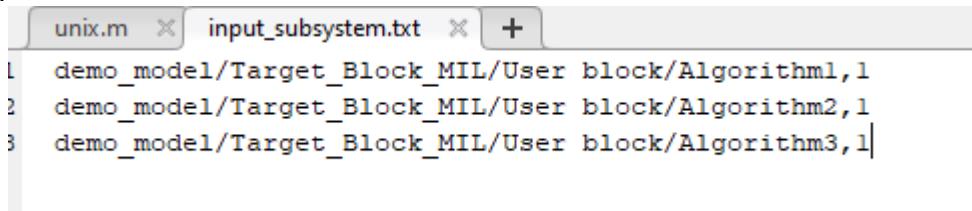
1. Only measure for the block inside the User Block.
2. Only measure for the block that define in the input\_subsystem.txt. Refer to section **3.4.2 Input file for Measurement**.
3. It must be the Atomic Subsystem.

### 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:



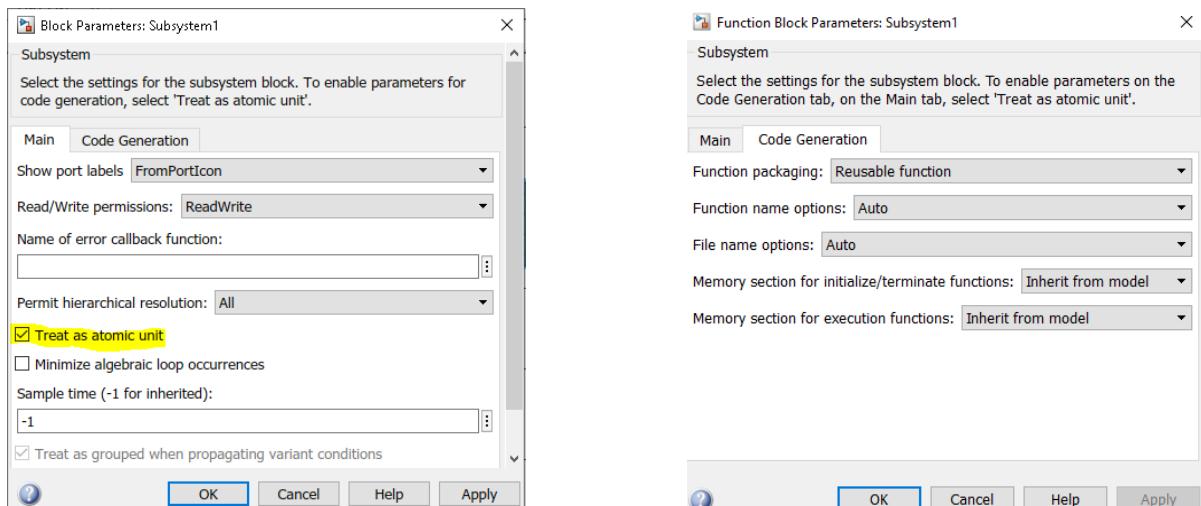
```
unix.m × input_subsystem.txt × +
1 demo_model/Target_Block_MIL/User block/Algorithm1,1
2 demo_model/Target_Block_MIL/User block/Algorithm2,1
3 demo_model/Target_Block_MIL/User block/Algorithm3,1
```

**Figure 3-45 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”.



**Figure 3-46 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.

	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	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
5	2	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
6	3	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
7	4	1	69001500	67119000	67509000	67521500	67679000	68531500	68551500	
8	5	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
9	6	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
10	7	1	68966500	67084000	67474000	67486500	67644000	68496500	68516500	
11	8	1	69001500	67119000	67509000	67521500	67679000	68531500	68551500	
12	9	1	69001500	67119000	67509000	67521500	67679000	68531500	68551500	
13	10	1	68966500	67084000	67474000	67486500	67644000	68496500	68516500	
14	11	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
15	12	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
16	13	1	68966500	67084000	67474000	67486500	67644000	68496500	68516500	
17	14	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
18	15	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
19	16	1	67484000	65859000	65991500	66004000	66161500	67014000	67034000	
20	17	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
21	18	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
22	19	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
23	20	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
24	21	1	69001500	67119000	67509000	67521500	67679000	68531500	68551500	
25	22	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	
26	23	1	68971500	67089000	67479000	67491500	67649000	68501500	68521500	

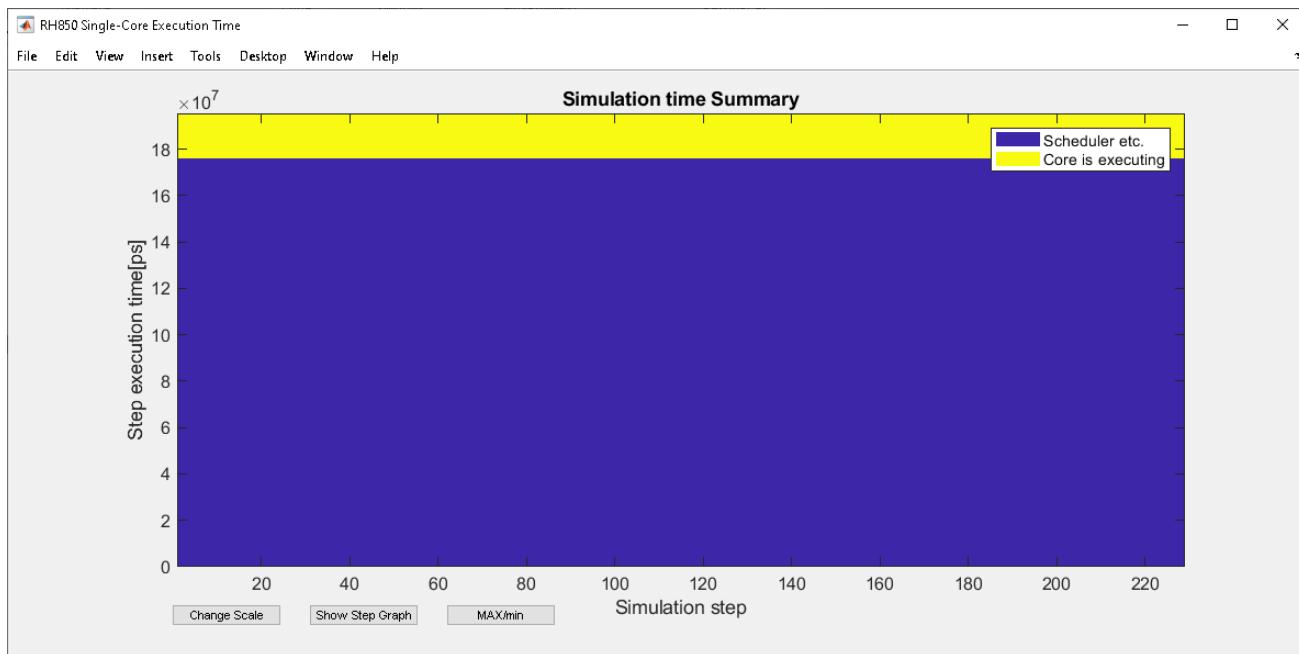
Input 2: input\_subsystem.txt file that specifies the subsystem to be measured, refer to section **3.4.2 Input file for Measurement**.

### 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]
```

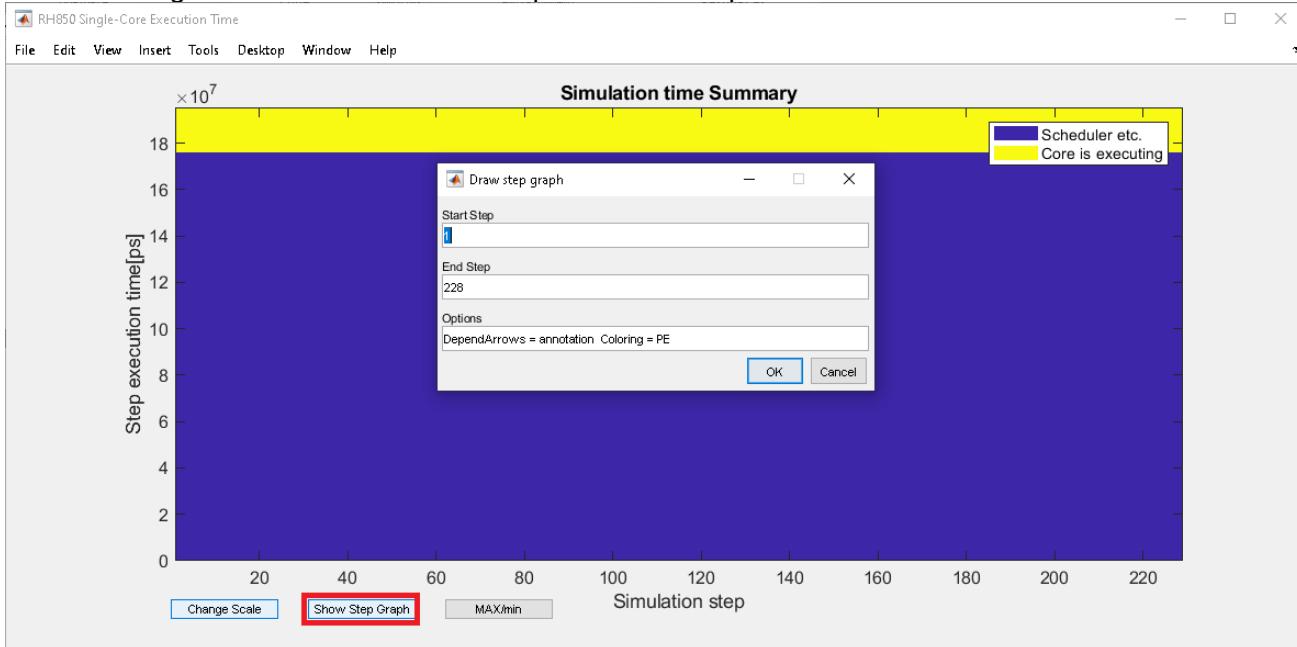


**Figure 3-47 Output figure of time measurement**

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

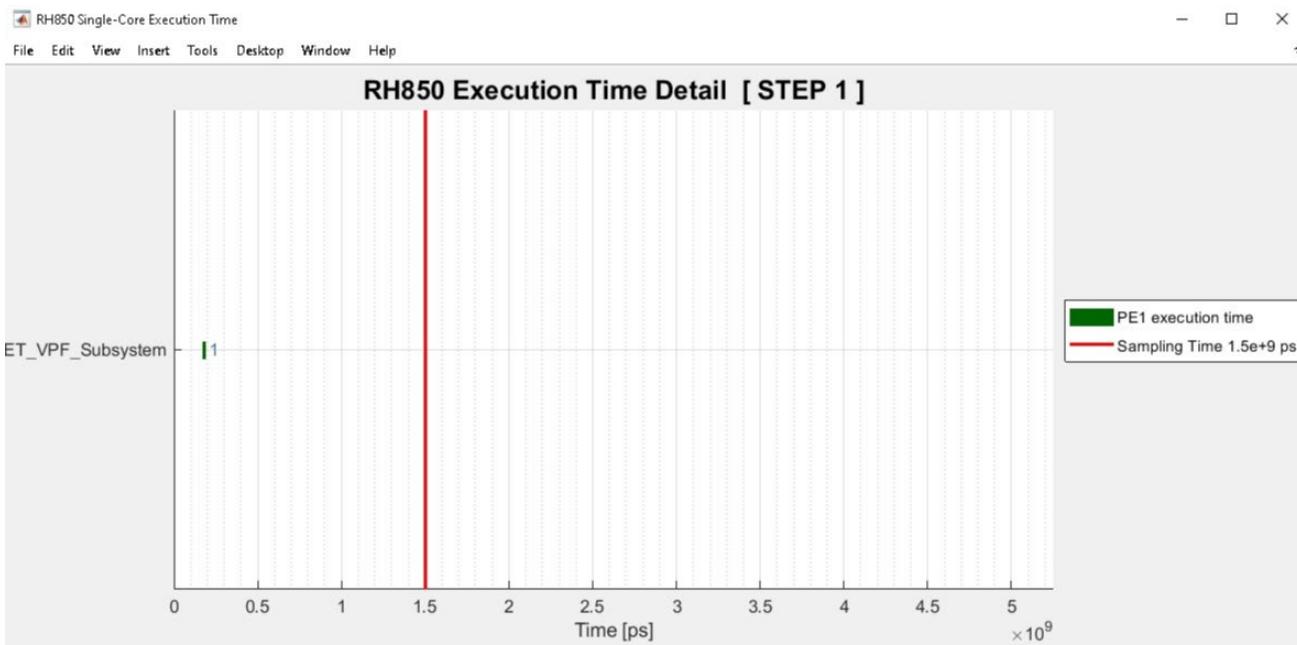
Users can perform some other functions as followings:

- (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”.



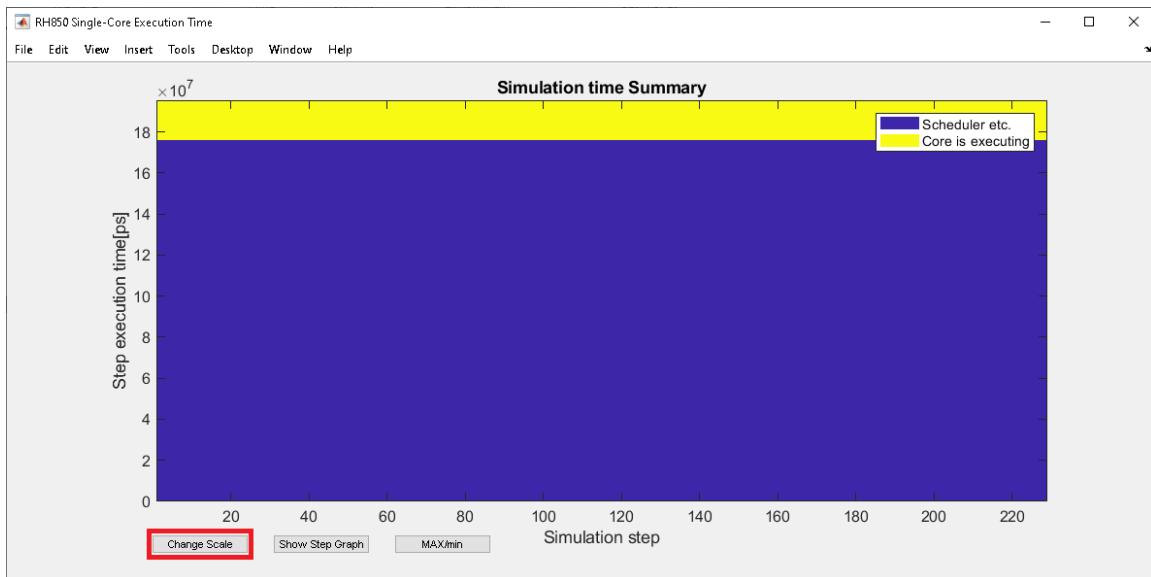
**Figure 3-48 Press [Show Step Graph] to show execution time each Subsystem**

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

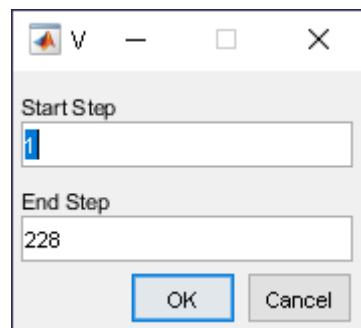


**Figure 3-49 Example of Execution Time Detail at step 1**

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



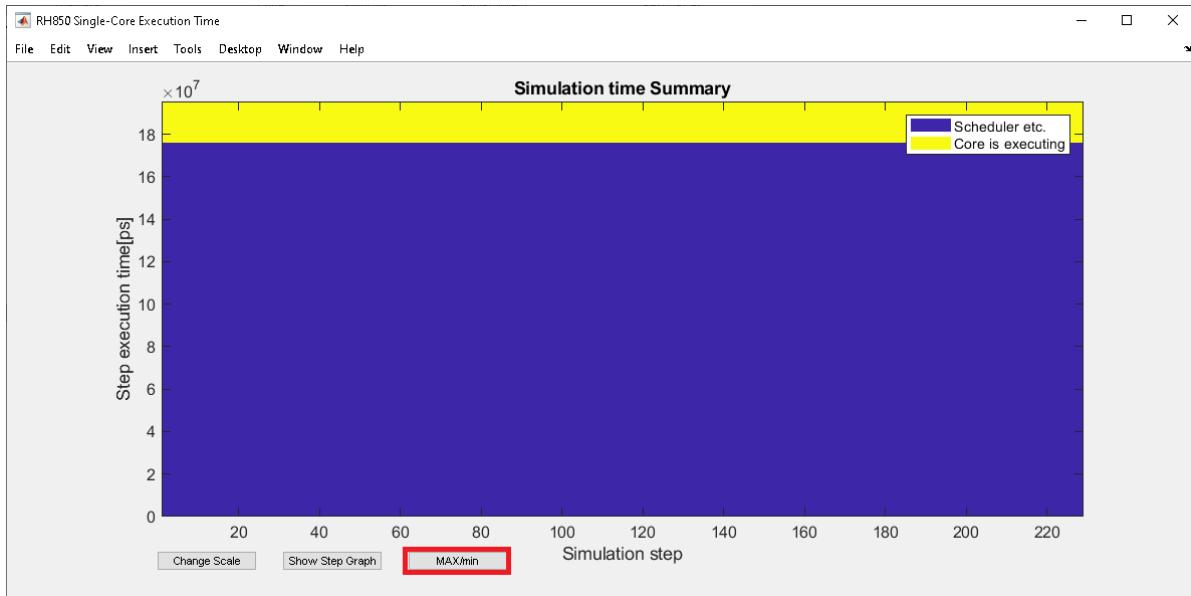
**Figure 3-50 Press [Change Scale] to change scale of step**



**Figure 3-51 Set “Start Step” and “End Step” value**

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

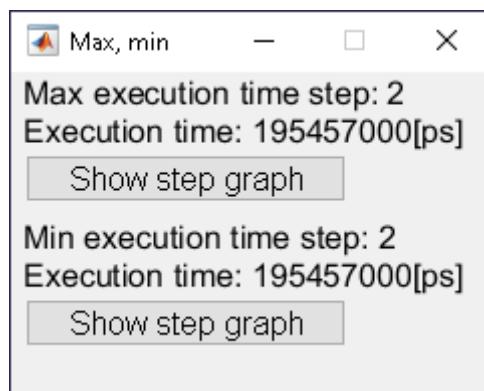
- (3) Press [MAX/min] to shows which steps has the maximum execution time and minimum execution time.



**Figure 3-52 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.



**Figure 3-53 Max/Min Dialog**

## 4 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.
- (3) 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.
- (4) For ADC in RH850/U2B, this version supports for Scan group 0, Multicycle scan mode  
For ADC in RH850/F1KM this version only supports for Scan group 1, Multicycle scan mode.
- (5) RESETOUT pin is assigned P8-6 (for RH850/F1KM-S1 and RH850/F1KM-S4) or P27\_0 (for RH850/U2B), so use other port.
- (6) 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.
- (7) 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.
- (8) 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)
- (9) TAUD MILS for Triangle PWM Output is not supported.
- (10) 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 Fastlss to CAIss.

- vlab.read\_register("RH850.CPUSS.PE1\_FastIiss.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.

- (11) 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).
- (12) For Port, ADC in RH850/U2B, this version does not support for AN050, AN051 and Port out 31 due to bugs of VLAB toolbox.
- (13) For CAN Pack and CAN Unpack parameter, MATLAB variable input is not supported, please input immediate value instead.
- (14) If the sampling time is short, the updated result cannot be prepared on time. Each peripheral cannot be in time depending on the sampling time.
- (15) The setting baud rate of peripherals high or low may affect to the communication time.

## 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.

### 4.2.3 Constant input of peripheral S-function block

When code generation from the MATLAB/Simulink models (subsystems) contains the constant output, MATLAB will generate unexpected source code. S-function source code for peripheral will be optimized.

To avoid this issue, please multiply the output of constant block with the “Repeating Sequence Stair” block (vector of output values is 1) in the user algorithm layer. After that connect the output to peripheral S-Function block. By this way, the output of constant block will be available every step with the same value.

## 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.

#### 4.3.5 Build the CS+ project when using the “Generate the CS+ project and automatically build”

For the “Generate the CS+ project and automatically build”, the default settings of CS+ will be used when automatically building, the user cannot change the settings. Therefore, an error can occur if the default settings of CS+ is inappropriate with use case.

To avoid this issue, please select “Generate the CS+ project only” to change the necessary settings of CS+ manually or select “None” to compile and build source code by makefile.

## 5 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]	<ol style="list-style-type: none"> <li>1. Set the correct path of the installation destination of Cygwin to [Cygwin Path]. The folder containing “/bin/bash.exe” must be specified.</li> <li>2. Check that Cygwin has normally been installed.</li> </ol>
[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]	<ol style="list-style-type: none"> <li>1. Set the correct path of the installation destination of VLAB to [VLAB Path]. The folder containing “vlab-ide.exe” must be specified.</li> <li>2. Check that VLAB has normally been installed.</li> </ol>
[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]	<ol style="list-style-type: none"> <li>1. Set the correct path of the installation destination of CS+ to [CS+ Path]. The folder containing “CubeSuiteW+.exe” must be specified.</li> <li>2. Check that CS+ has normally been installed.</li> </ol>
[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]	<ol style="list-style-type: none"> <li>1. Set the correct path of the installation destination of Smart Configurator to [SC Path]. The folder containing “SmartConfigurator.exe” must be specified.</li> <li>2. Check that Smart Configurator has normally been installed.</li> </ol>

**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]	E0107 The parameter <paramName> does not exist.
[Explanation]	The parameter <paramName> does not exist in ET-VPF Options
[Action by User]	1. Check if the System Target File is set to etvpf.tlc 2. Check if the path to ET-VPF installation folder is set to MATLAB path
[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]	<ol style="list-style-type: none"><li>1. Check if the model settings, components are correctly.</li><li>2. If above information is correct, re-execute vHILS to generate execution time measurement result again.</li></ol>

## 5.4 Errors during starting CS+ and building CS+ project

The following table lists the messages that are detected in ET-VPF processing from starting CS+ to building CS+ project.

**Table 5-4 Error messages display in CS+**

[Message]	E0301 Creating the CS+ project was not possible (project.Create error).
[Explanation]	The CS+ project file could not be generated.
[Action by User]	<ol style="list-style-type: none"> <li>1. Check that the CS+ version is supported by ET-VPF.</li> <li>2. Check that the CS+ Python plug-in is enabled.</li> </ol>
[Message]	E0302 Adding the source file was not possible (project.File.Add error).
[Explanation]	The source file could not be registered in the CS+ project file.
[Action by User]	<ol style="list-style-type: none"> <li>1. Check that the MATLAB version is supported by ET-VPF.</li> </ol>
[Message]	E0303 Removing the source file was not possible (project.File.Remove error).
[Explanation]	The source file could not be removed from the CS+ project file.
[Action by User]	<ol style="list-style-type: none"> <li>1. Check the removal permission of the CS+ project file.</li> <li>2. Check whether the CS+ project file has existed or not.</li> </ol>
[Message]	E0304 Setting the link options was not possible (link_option.sec error).
[Explanation]	An error occurs when link_option.sec contains incorrect value for setting Section Start Address value on CS+ Project.
[Action by User]	<p>Do the following procedure</p> <ol style="list-style-type: none"> <li>1. Review the value and syntax, and then do correction in the section file (renesas_link_RH850_option_single.sec) in ET-VPF installation folder.</li> <li>2. Re-generate the test environment.</li> <li>3. When the memory size of the device is small, consider the use of a device of large memory size.</li> </ol>
[Message]	E0305 Building was not possible (build.All error).
[Explanation]	An error occurred at build.
[Action by User]	<p>Check the following and regenerate the test environment.</p> <ol style="list-style-type: none"> <li>1. Review the property setting of CS+.</li> <li>2. Check the error message displayed in the CS+ output panel.</li> <li>3. When the memory size of the device is small, consider the use of a device of large memory size.</li> </ol>

## 5.5 Errors during building and generating the vHILS environment

The following table lists the messages that are detected in ET-VPF processing when building and generating the vHILS environment.

**Table 5-5 Error messages display when building and generating the vHILS environment**

[Message]	E0310 The <Full File Name> file is not existed. Please check this file.
[Explanation]	This error message is displayed when cannot find the input file.
[Action by User]	<ol style="list-style-type: none"> <li>1. Check if the input file has existed or not.</li> <li>2. Check if the full path of input file is right or not.</li> </ol>

## 5.6 Errors during communication between MATLAB and VLAB

The following table lists the messages that are detected in ET-VPF processing when communication between MATLAB and VLAB

**Table 5-6 Error messages display when communication between MATLAB and VLAB**

[Message]	E0401 An incorrect automation server is registered.
[Explanation]	A different version of MATLAB registered as the Automation Server.
[Action by User]	<ol style="list-style-type: none"> <li>1. Use “regmatlabserver” command (without quotes) to register current MATLAB as an Automation Server.</li> </ol>

# Index

## A

Atomic ..... 48, 49

## E

error messages ..... 59

## G

Graph Viewer ..... 50

## I

Installation ..... 11

## L

License types ..... 8

## M

maximum execution time ..... 54

MILS ..... 5, 27, 28, 55

minimum execution time ..... 54

## O

Operating Environment ..... 6

## P

peripheral block ..... 15

peripherals' source cod ..... 5

peripherals' source code ..... 38

**Points for Caution** ..... 55

## S

Simulink ..... 6

## T

time measurement method ..... 48

## U

Uninstallation ..... 13

## V

vHILS ..... 5

vHILS environment ..... 37

VLAB ..... 7

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
1.00	September 19, 2022	-	First edition issued
1.01	February 1, 2023	5	Updated overview
		6	Added explanation of MEX-file compiler
		9	Added notice of installation folder
		10,11	Added licensing guide
		13	Added list of usable blocks
		20	Added sample using Vehicle Network Toolbox
		41	Correct format of input_subsystem.txt
		42	Deleted how to collect the measurement time
		7, 30, 47, 48	Added and updated cautions
1.02	January 16, 2024	-	Deleted descriptions of officially unsupported devices
		-	Added description for RH850/U2B

---

Embedded Target for RH850 Virtual Platform  
V1.00.00

RH850 Model-Based Development Tool User's Manual

Publication Date: Rev.1.02 January 16, 2024

Published by: Renesas Electronics Corporation

---

Embedded Target for RH850 Virtual Platform  
V1.00.00

RH850 Model-Based Development Tool



Renesas Electronics Corporation

R20UT5142EJ0102