RX Family

Amazon FreeRTOS download with e^2 studio

Introduction

Amazon FreeRTOS is a real-time operating system that augments the FreeRTOS kernel with libraries for connectivity, security, and over-the-air (OTA) updates. Amazon FreeRTOS also includes some demo applications that demonstrate Amazon FreeRTOS features on qualified boards.

Renesas e^2 studio is a development environment based on the popular Eclipse CDT (C/C++ Development Tooling), covers build (editor, compiler and linker control) as well as debug interface. It also supports to integrate the Amazon FreeRTOS demo applications and run them on Renesas boards.

Amazon FreeRTOS configuration feature in Smart Configurator provides a graphical user interface (GUI) configuration and code generation tool for Renesas RX microcontroller family. It helps user save valuable time to import project and take advantage of the functionality provided by the additional libraries. It also provides a function to change the setting of Amazon FreeRTOS kernel through GUI easily.

Objectives

This document helps users to be familiar with the procedures to download, configure and run the Amazon FreeRTOS demo applications using the new features of e^2 studio: downloading Renesas GitHub Amazon FreeRTOS projects and configuring FreeRTOS libraries using Smart Configurator.

Operating Environment

Operation was confirmed in the following environments.

IDE	e2 studio 7.5.0
Toolchains	CCRX Compiler v3.0.1
Target devices	Renesas RX Family
Emulators	E2, E2 Lite, E1 and E20

Contents

Note

- AWS™ is a trademark of Amazon Web Services, Inc. (https://aws.amazon.com/trademark-guidelines/)
- FreeRTOS™ is a trademark of Amazon Web Services, Inc. (https://freertos.org/copyright.html)
- GitHub® is a trademark of GitHub, Inc. (https://github.com/logos)

1. Outline

This document describes the procedure to perform the demonstration for Amazon FreeRTOS on Renesas RX family by introducing following steps:

- Prepare and configure the demo project
- Select the demo to run
- Set up AWS corresponding to the demo
- Set up hardware (target device)
- Configure debug serial port
- Build and run the demo

2. Downloading from Renesas GitHub

2.1 Download source code from GitHub

Before e² studio v7.5, user must download the source code from GitHub manually, then import the project to e² studio workspace. The new version of e² studio supports user to import the source code right inside the IDE. Amazon FreeRTOS project importing functionality is supported in e² studio v7.5 and above.

At the beginning, user would be able to choose the version of Amazon FreeRTOS package, and the selected version will be downloaded from GitHub and imported automatically into the project. This makes it easier for the user, so that the user can focus only on Amazon FreeRTOS configuration and writing application code.

The figure below shows how to import Amazon FreeRTOS project:

1. Launch e² studio v7.5
2. Select [File] → [Import..]
3. Select “Renesas GitHub Amazon FreeRTOS Project”

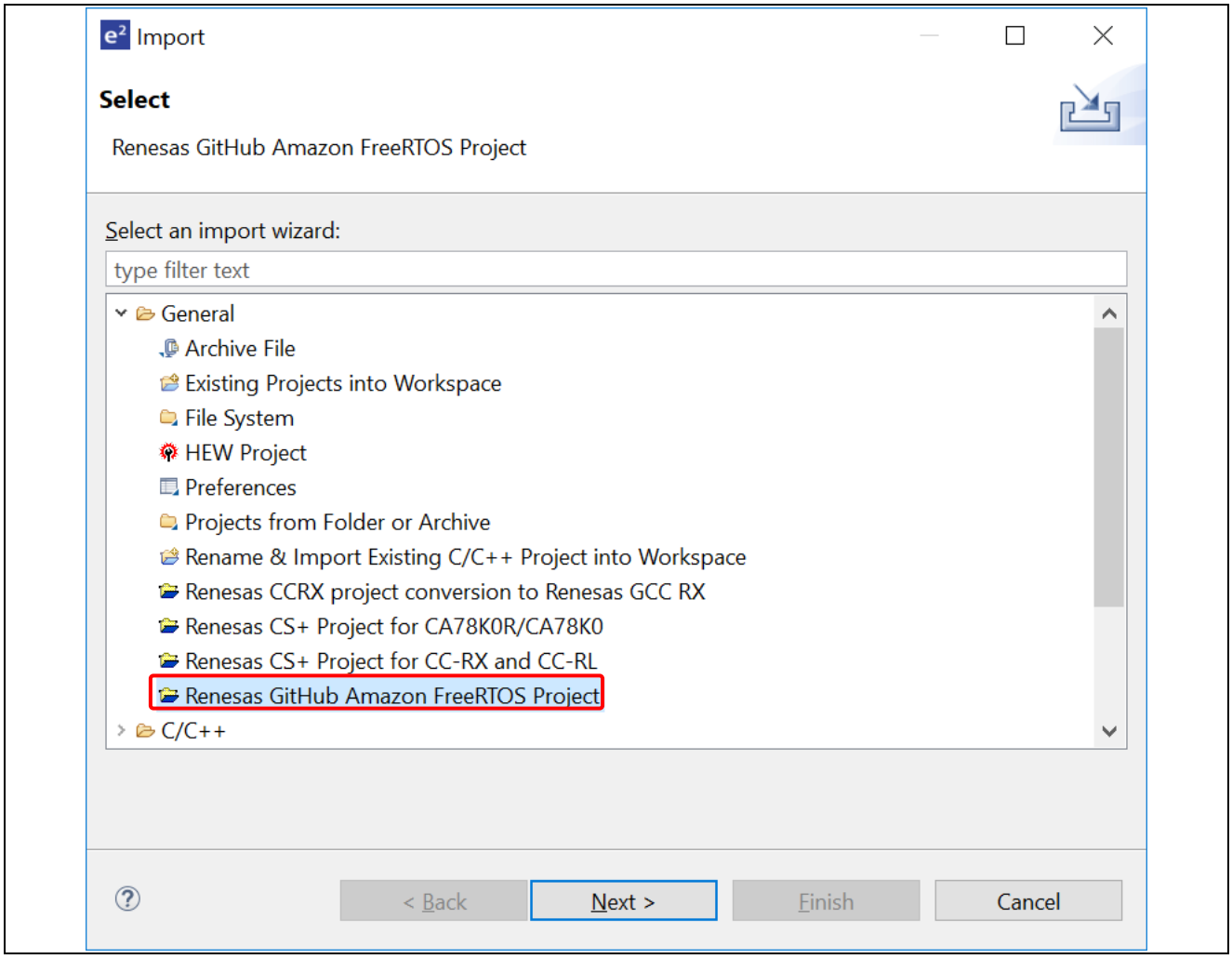


Figure 2-1 Download Amazon FreeRTOS projects from GitHub

4. If there is no Renesas GitHub Amazon FreeRTOS project which is already downloaded, “RTOS Version setting” list box will be empty.

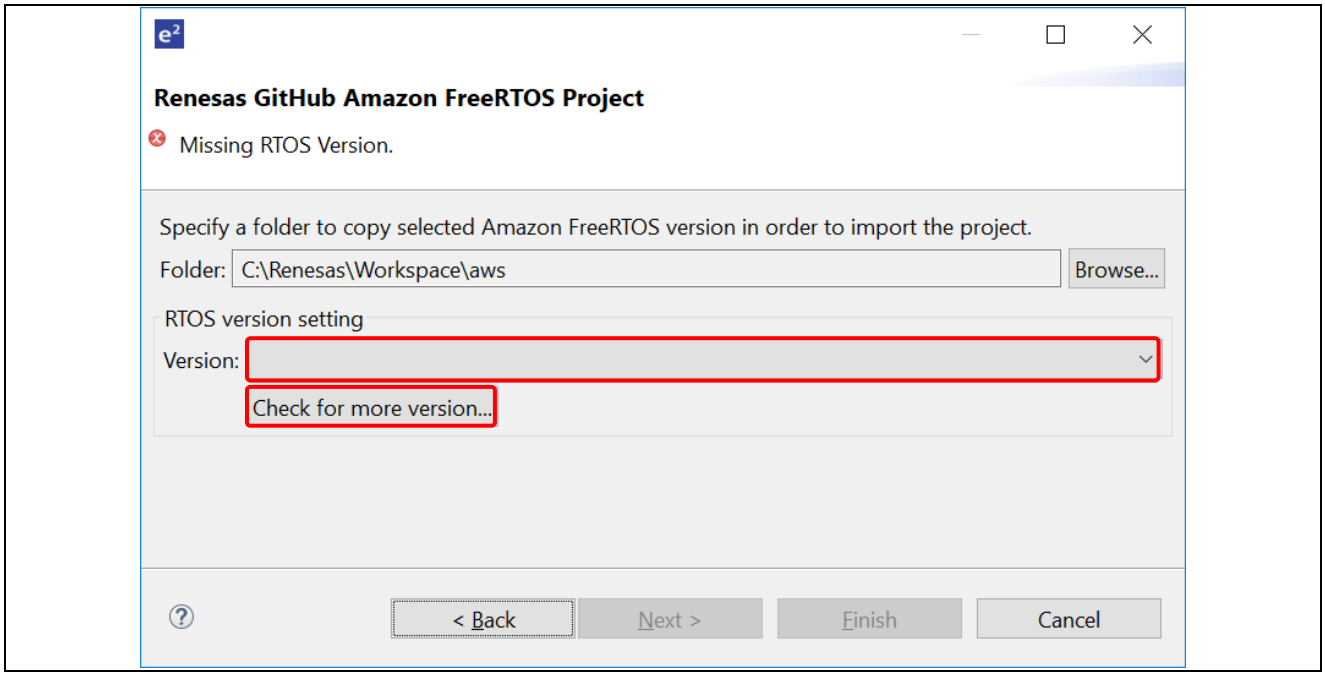


Figure 2-2 There is no Amazon FreeRTOS project downloaded yet

5. Select “Check for more version…” to show the download dialog.

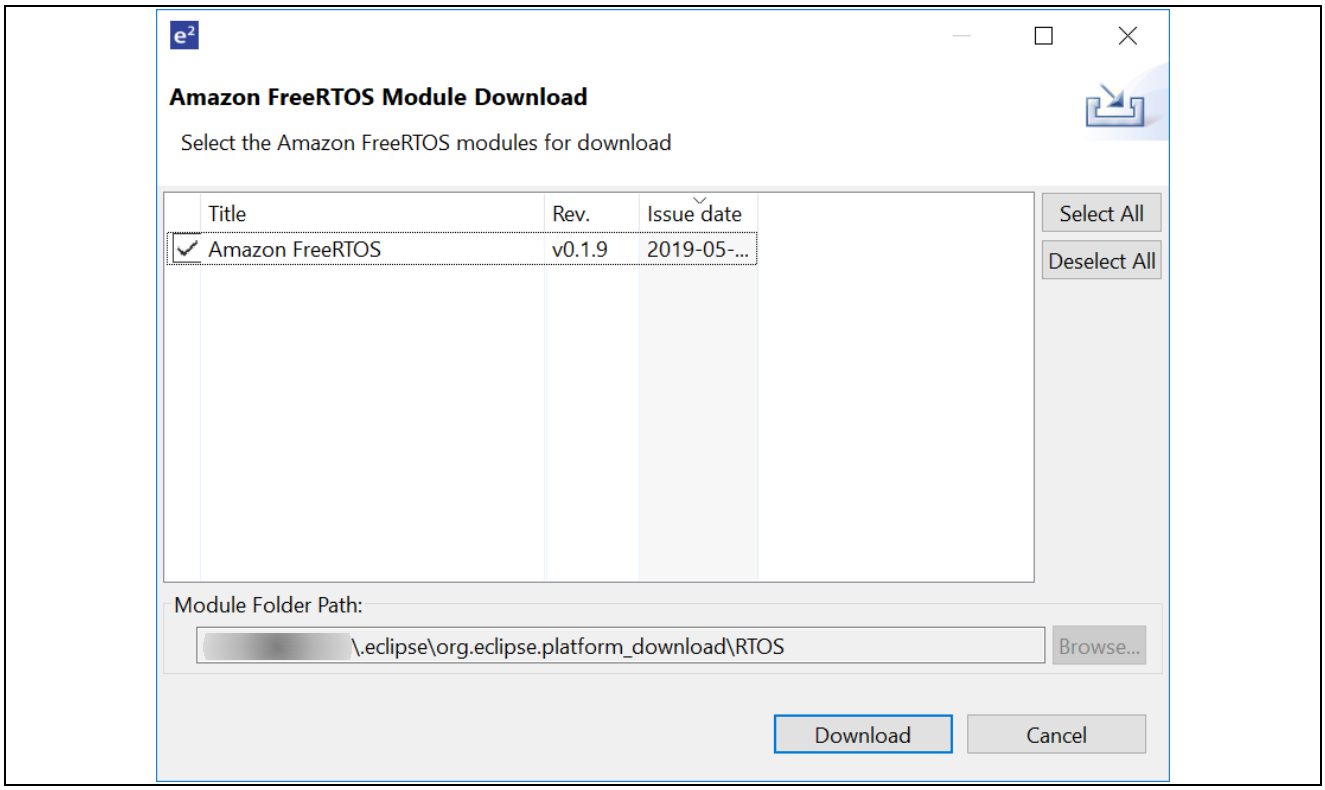


Figure 2-3 Amazon FreeRTOS download dialog

6. Agree the end user license agreement.

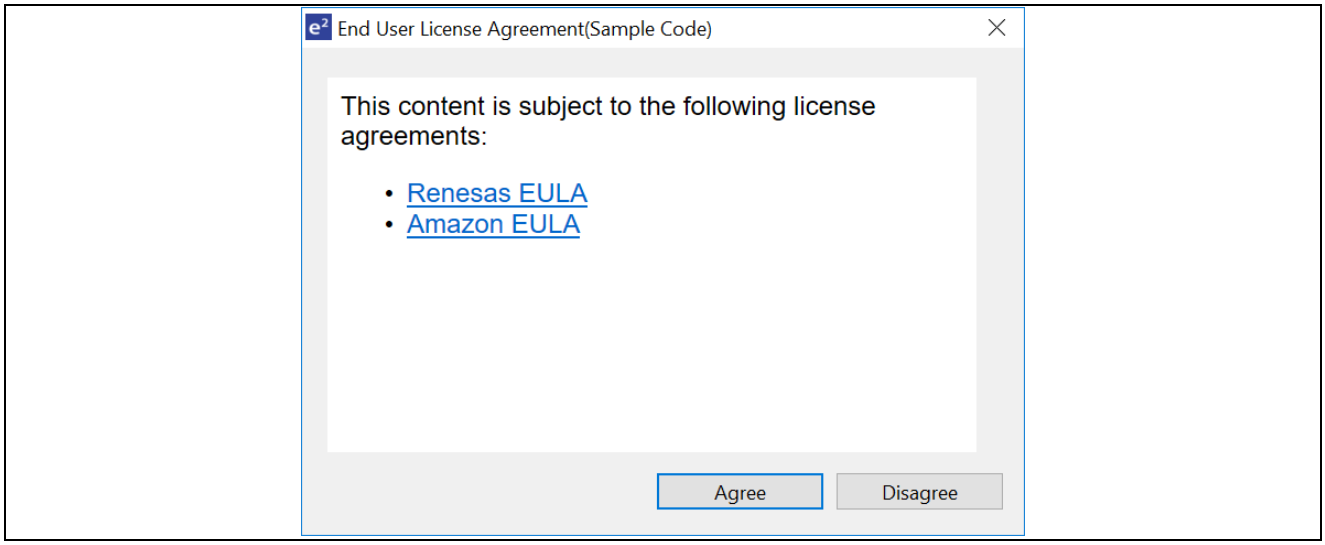


Figure 2-4 User license agreement

7. Wait for downloading completed.

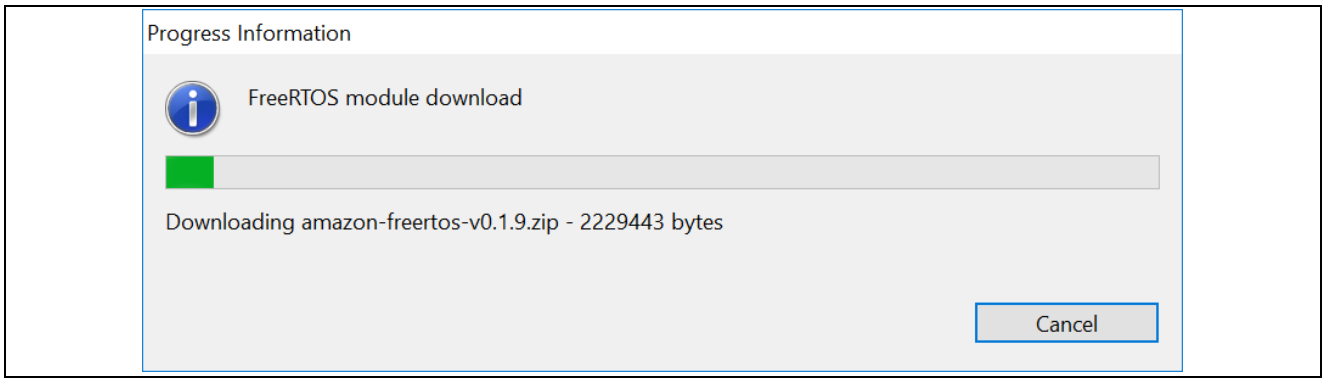


Figure 2-5 Downloading Amazon FreeRTOS project dialog

8. The downloaded version is shown.

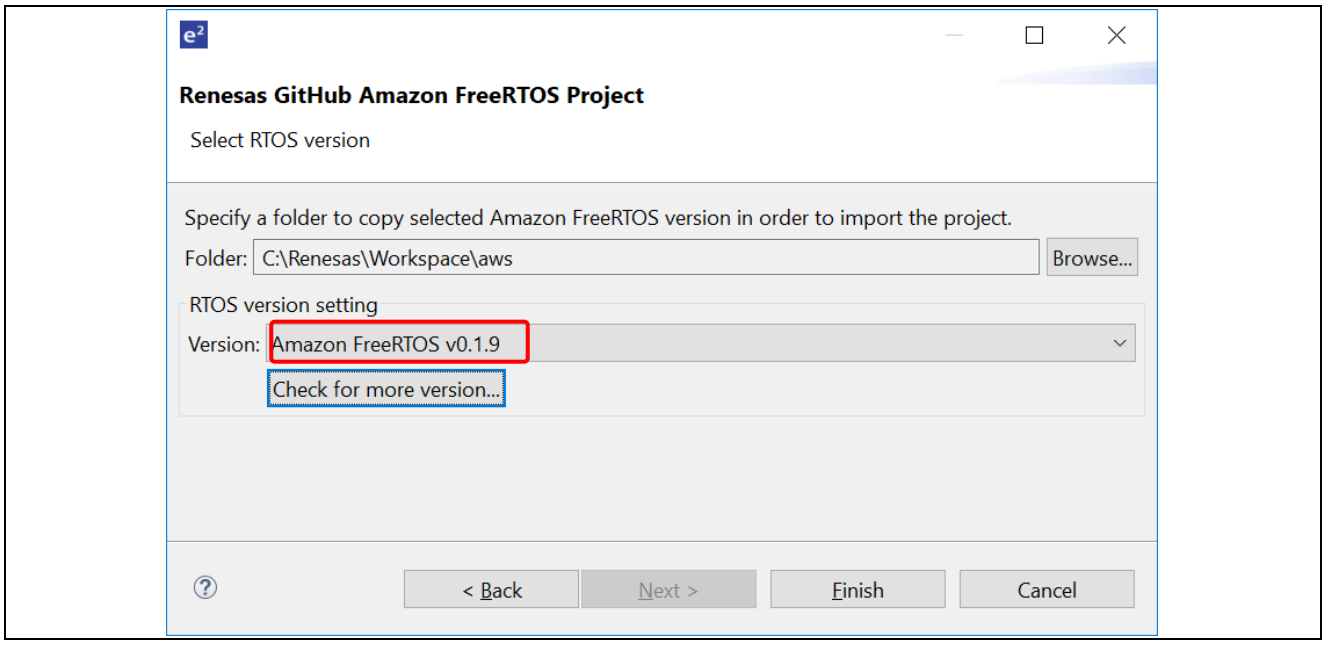


Figure 2-6 Select source code version

9. Select the project to import. Only 1 project can be imported to 1 workspace. Keep “Copy projects into workspace” unchecked.

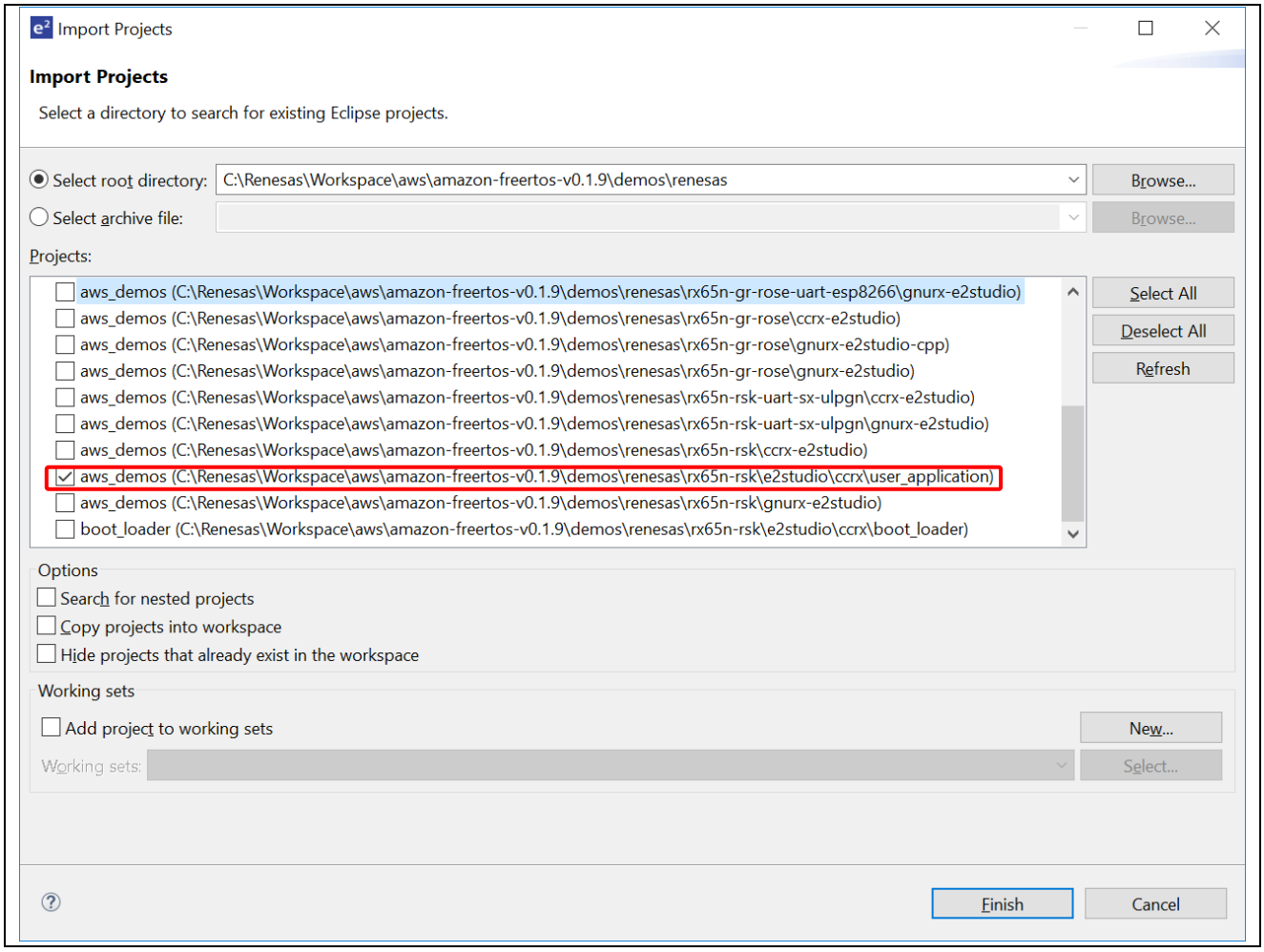
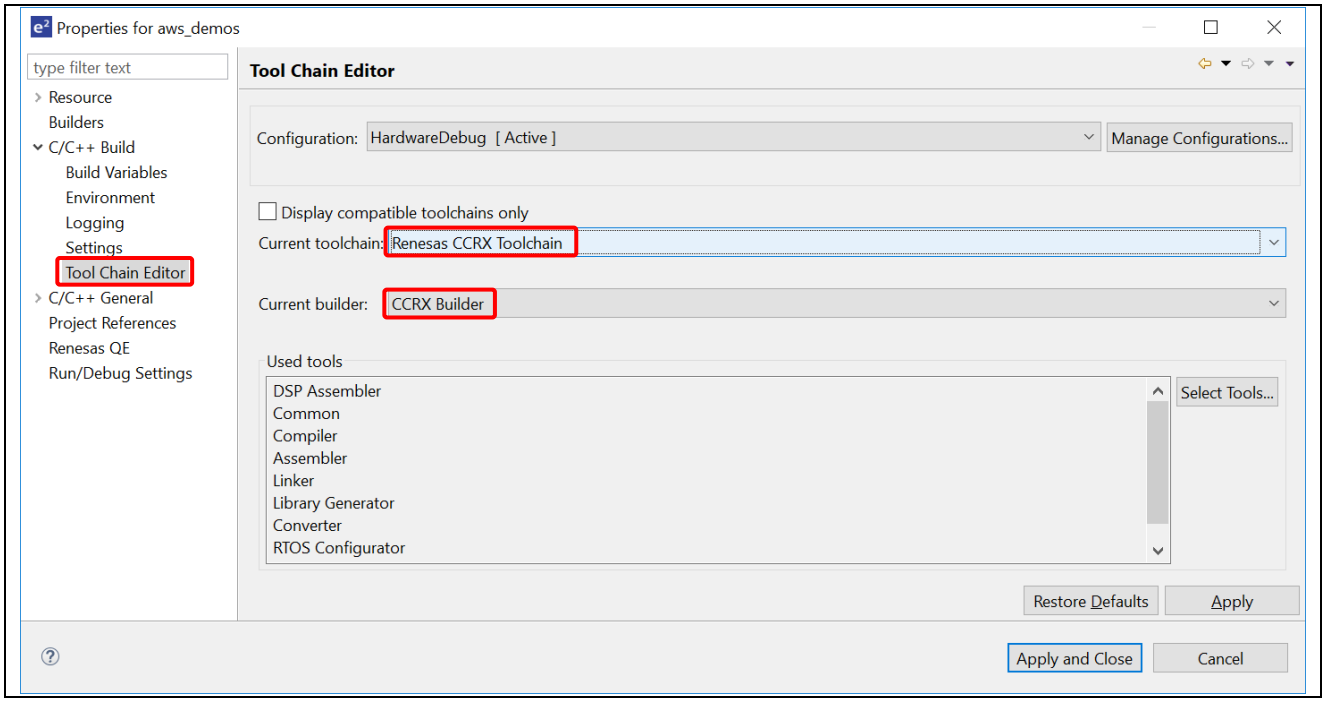


Figure 2-7 Select project to import

10. Open project properties, select toolchain and builder, then specify toolchain version.



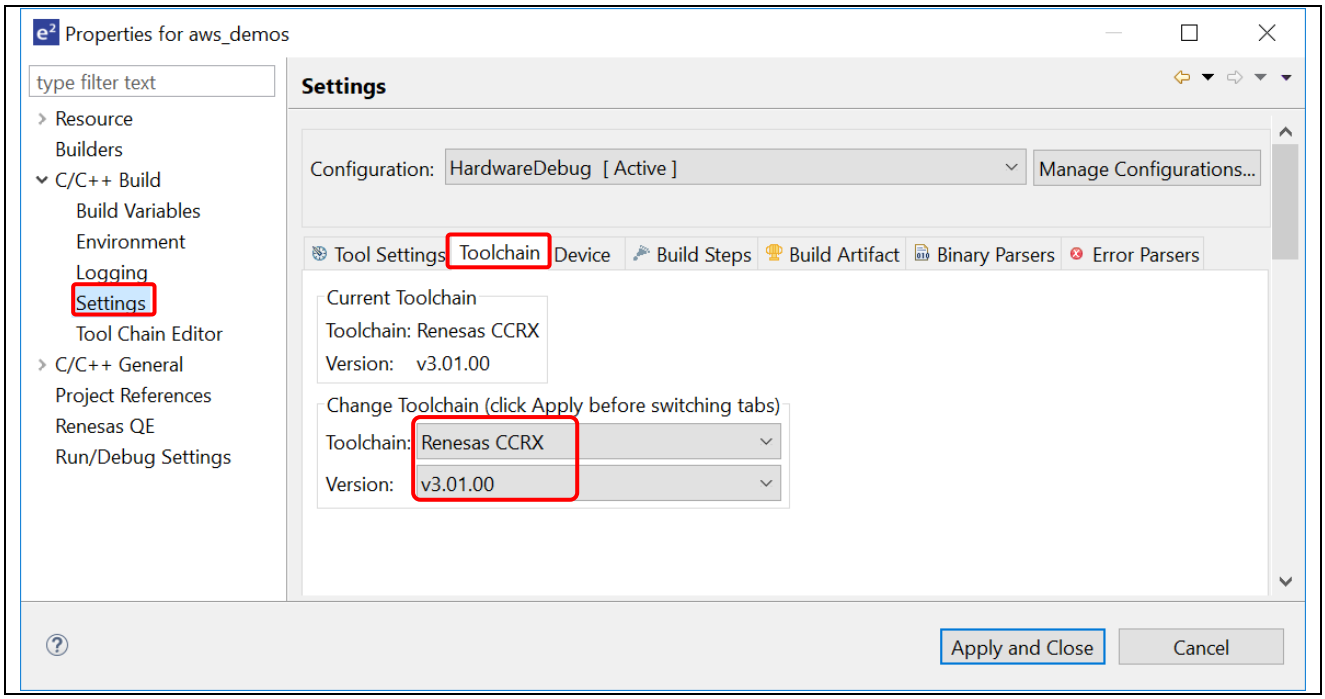


Figure 2-8 Select toolchain

2.2 New folder structure

From version 0.1.9 of GitHub Amazon FreeRTOS source code, Renesas introduces a new folder structure for device driver libraries (FIT). Instead of using generated code from Smart Configurator, a .bat file is added to exclude the generated code from build and refer to modified FIT in the package.

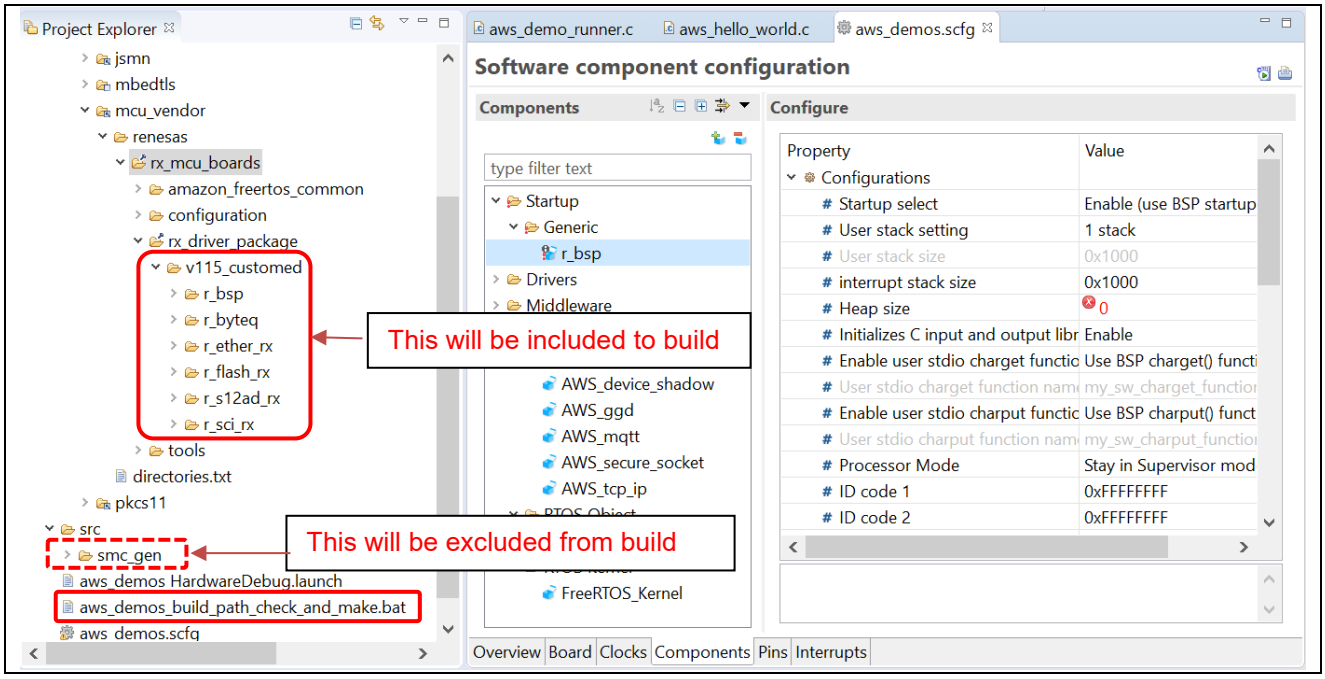


Figure 2-9 New folder structure of the project

3. Configure the Amazon FreeRTOS

Smart Configurator perspective will be launched as shown below. In Amazon_demos.scfg panel, FreeRTOS Kernel, Object and Amazon libraries packages are ready and displayed in Current Configuration in [Overview] tab

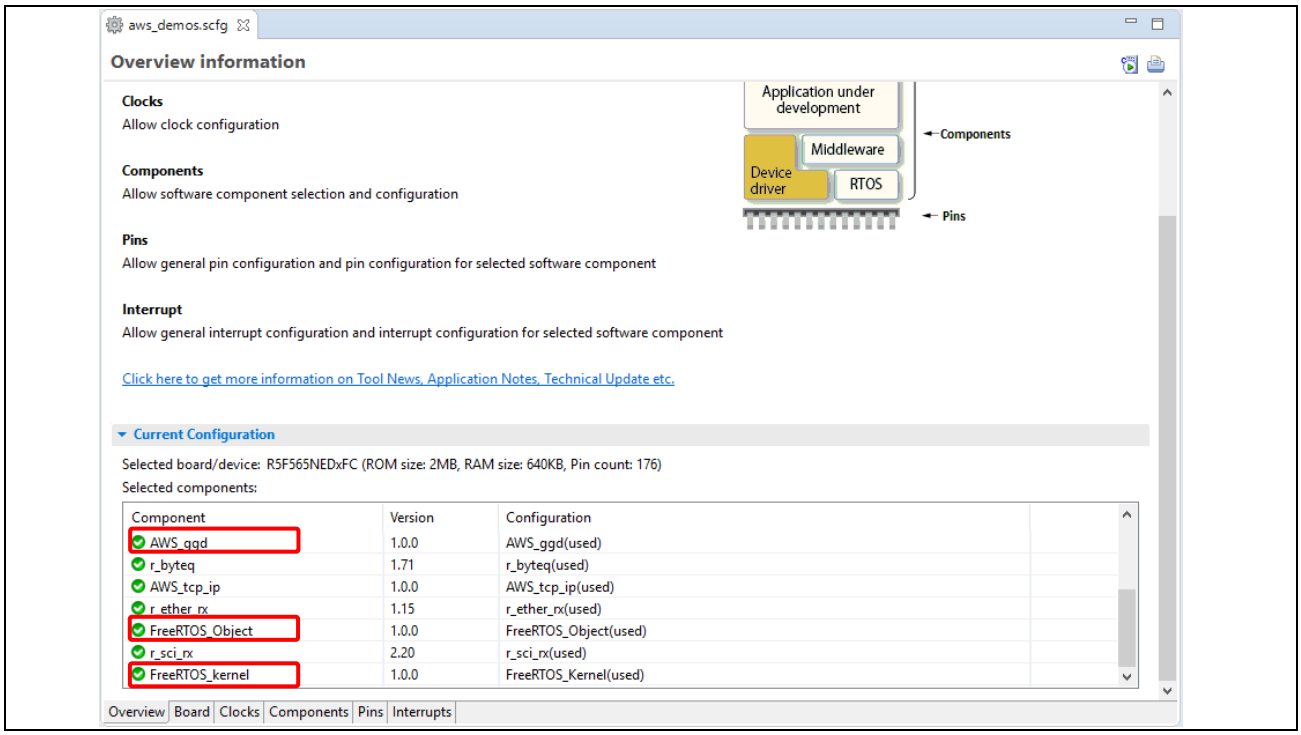


Figure 3-1 Smart Configurator perspective with Amazon FreeRTOS

3.1 Amazon FreeRTOS Kernel

1. In [Components] tab, select [FreeRTOS_kernel] layer in the Components tree at the left panel.
2. The corresponding parameter is displayed in the right panel for users to quickly manage FreeRTOS kernel setting.
   This provides all possible configuration setting options for FreeRTOS kernel.
3. Click on any configurations option setting in the right panel to display its definition as shown in the picture below.

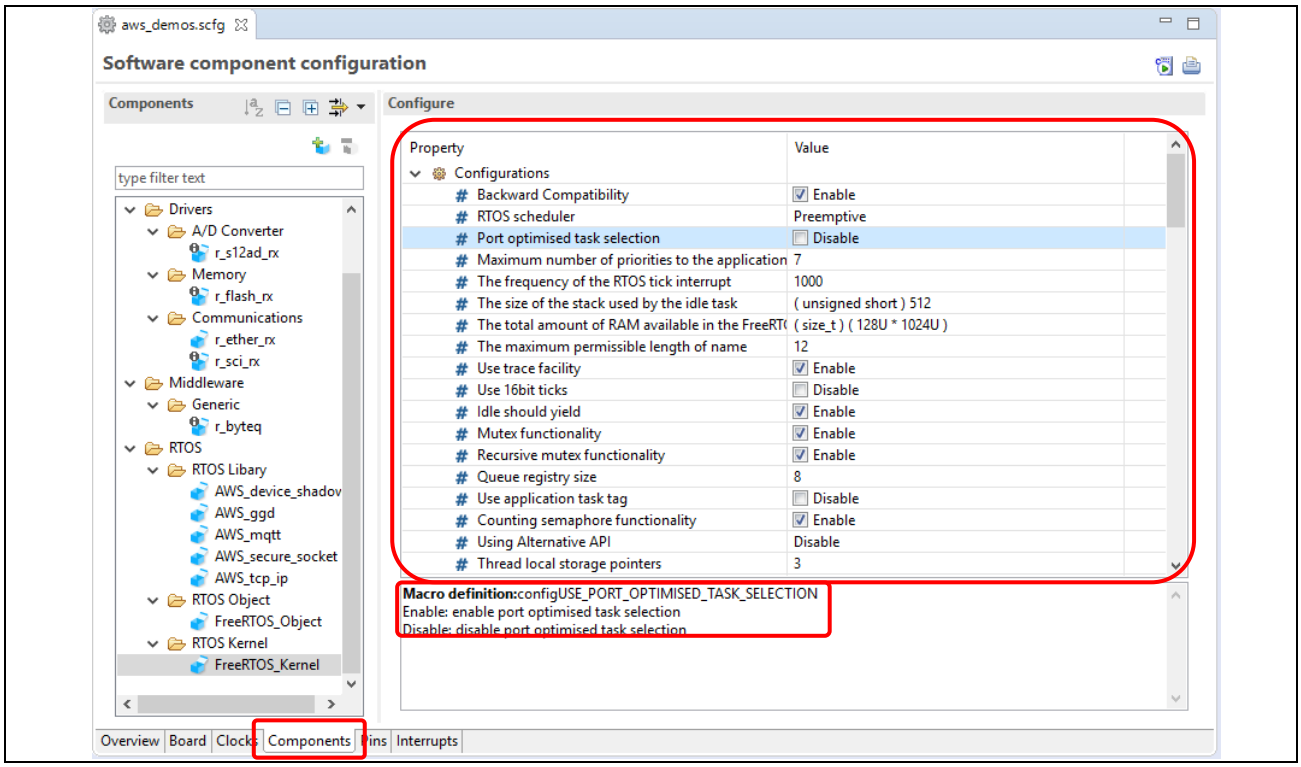


Figure 3-2 FreeRTOS_Kernel configuration panel

3.2 Amazon FreeRTOS Object

1. In [Components] tab, select [FreeRTOS_Object] layer in the Components tree at the left panel.
2. Go to option setting in the right panel to configure objects such as task, semaphores, queues, software timer, event groups, stream and message buffers.
3. Under object labels, click +/- buttons to create new objects. All options settings can be edited and updated at any time.

- **Tasks:**
  New task will be created/deleted after clicking +/- buttons. Option setting will be showed in the right panel where **Initialize, Task Code, Task Name, Stack Size, Task Handler, Parameter and Priority** can be edited.
  The task can be created by two ways:
  — kernel start: created task will be executed after calling vTaskStartScheduler().
  — manual: prepare some tasks that user thinks it will be useful later. The tasks created by this way will not be executed at the beginning after vTaskStarScheduler() is called unless changing to kernel start mode.

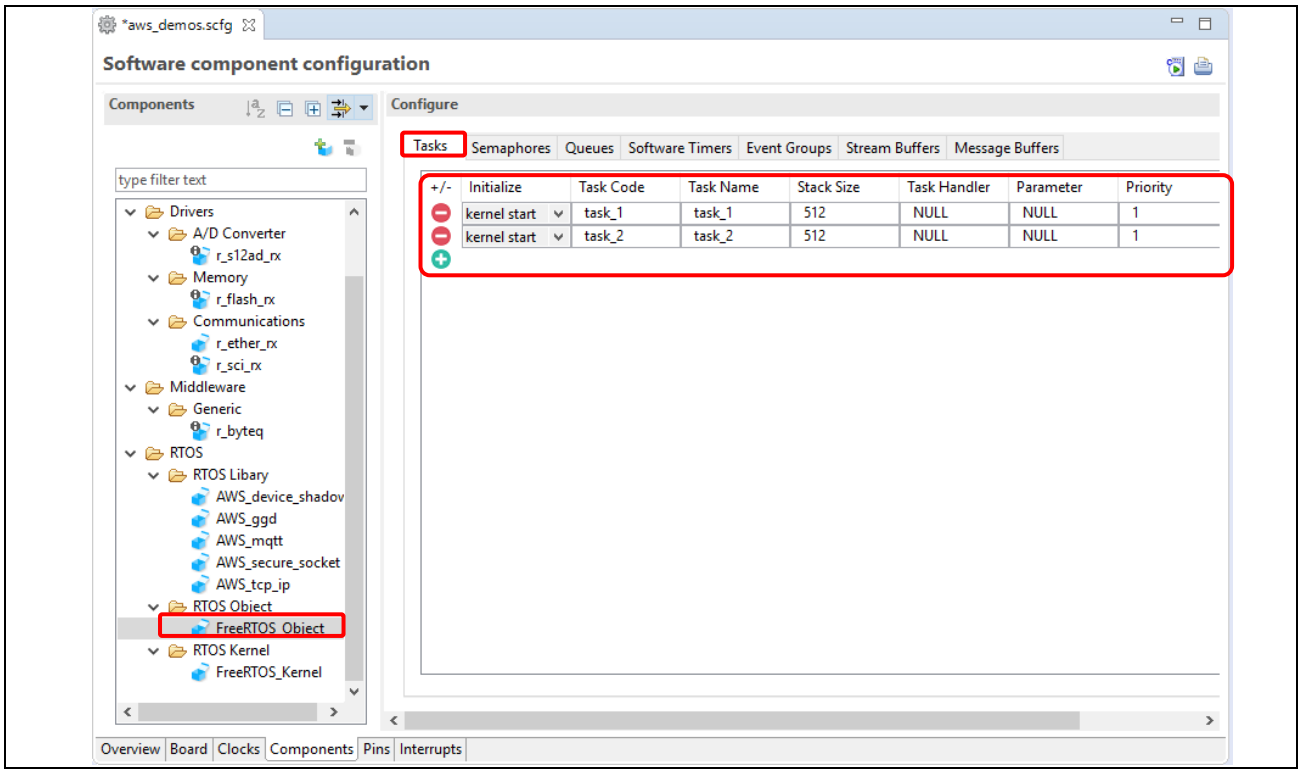


Figure 3-3 Tasks configuration panel

- **Semaphores:**
  New **Semaphores** will be created/deleted after clicking +/- buttons. Option setting will be showed in the right panel where **Semaphore Type and Semaphore Handler** can be edited.

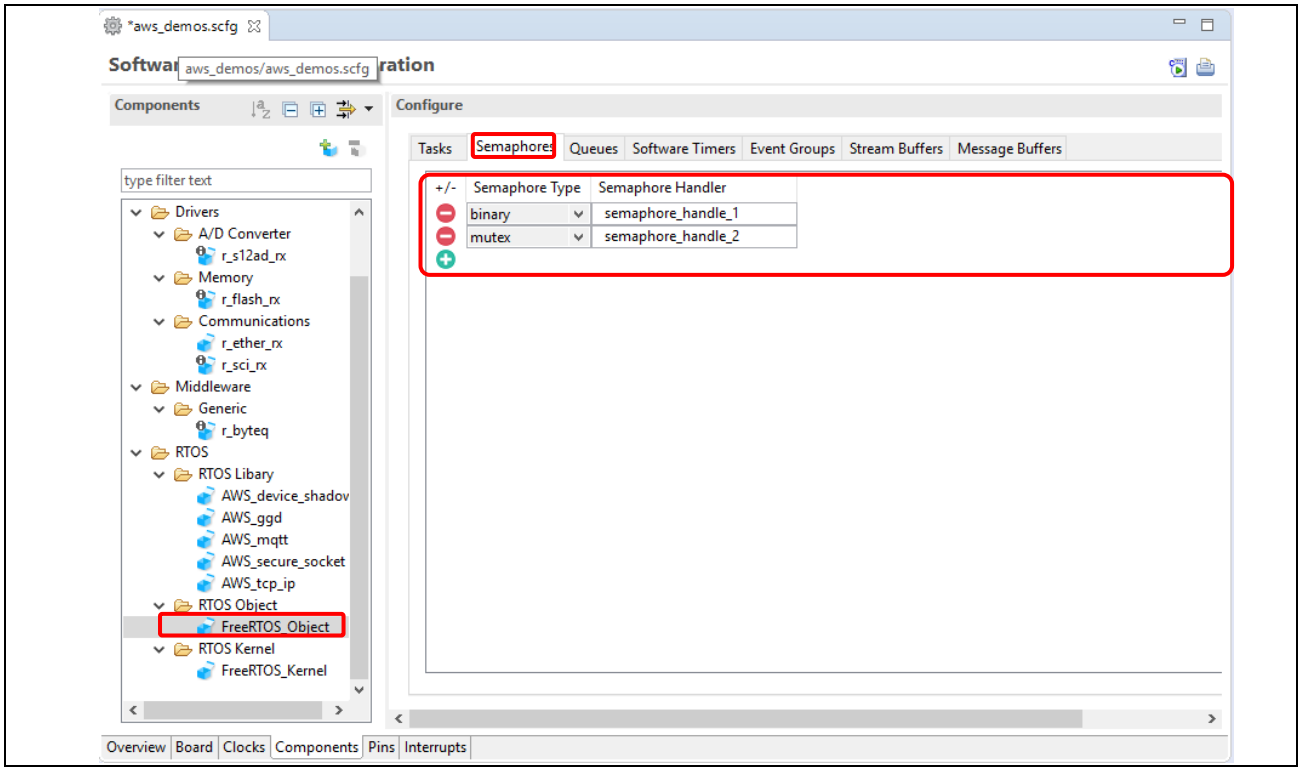


Figure 3-4 Semaphores configuration panel

- **Queues:**
  New **Queues** will be created/deleted after clicking +/- buttons. Option setting will be showed in the right panel where **Queue Handler, Queue length and Items size** can be edited.

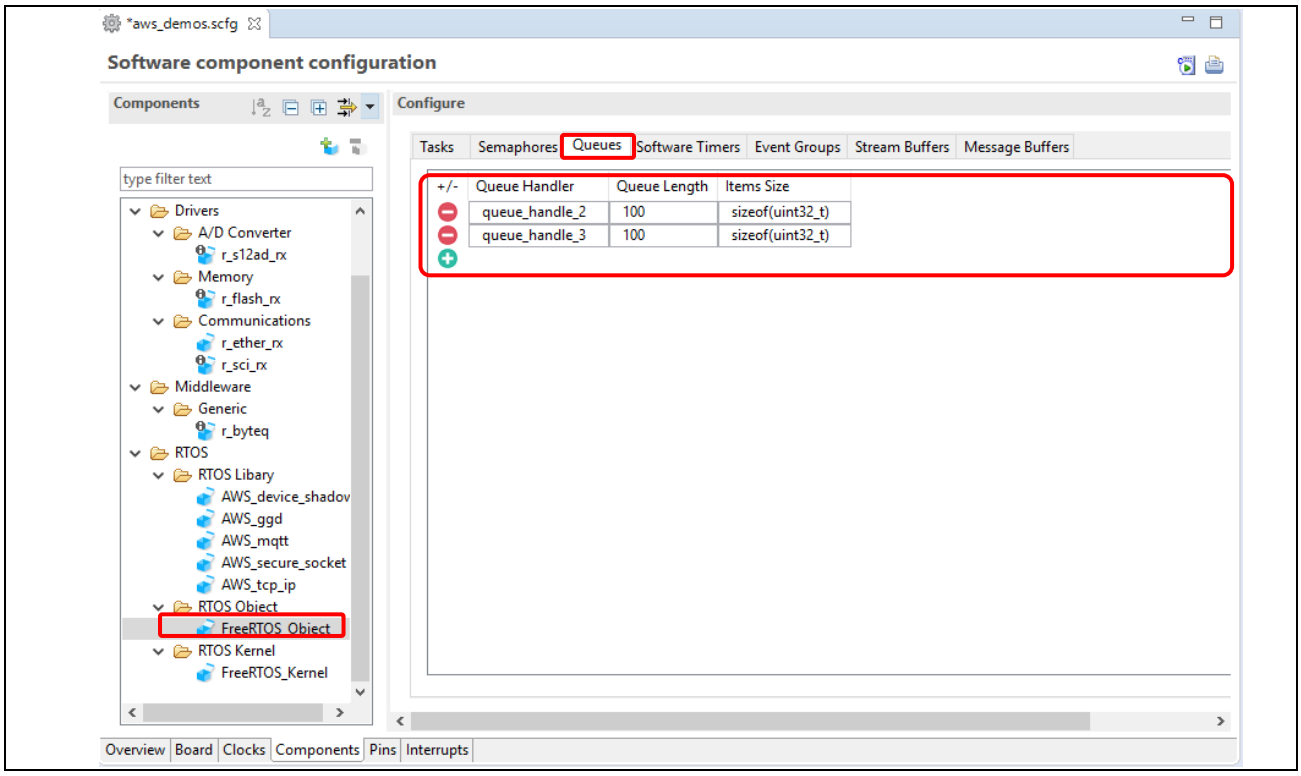


Figure 3-5 Queue configuration panel

- **Software timer:**
  New **Software timer** will be created/deleted after clicking +/- buttons. Option setting will be showed in the right panel where **specific parameters** can be edited.

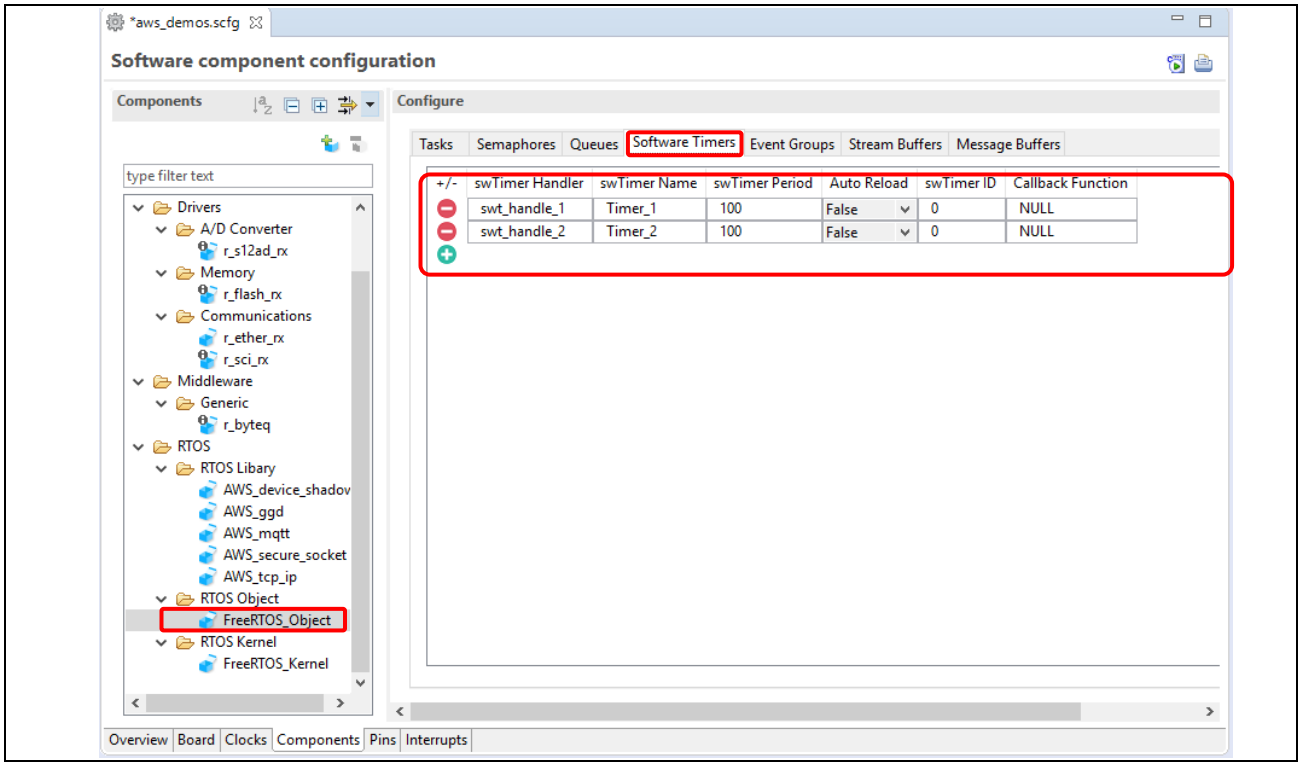


Figure 3-6 Software timer configuration panel

- **Event group:**
  New **Event group** will be created/deleted after clicking +/- buttons. Option setting will be showed in the right panel where **specific parameters** can be edited.

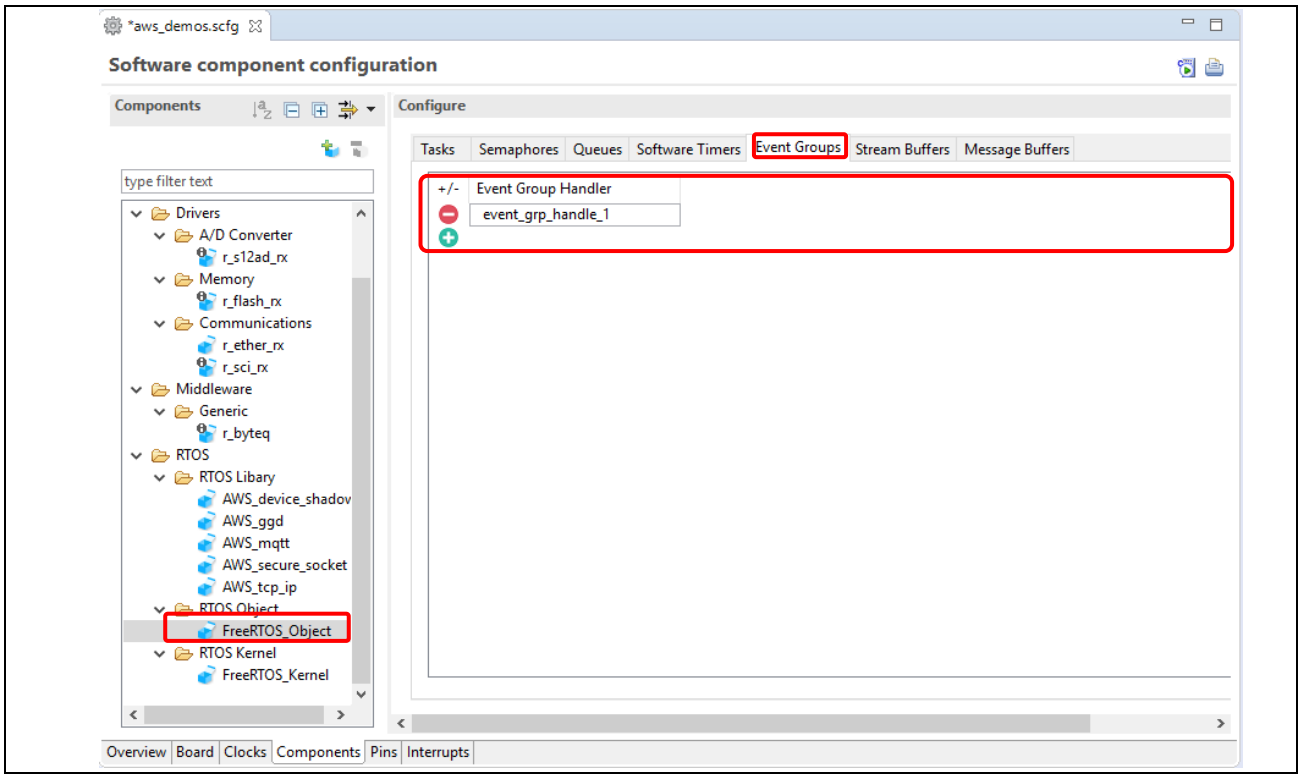


Figure 3-7 Event group configuration panel

- **Stream buffers:**
  New **Stream buffers** will be created/deleted after clicking +/- buttons. Option setting will be showed in the right panel where **specific parameters** can be edited.

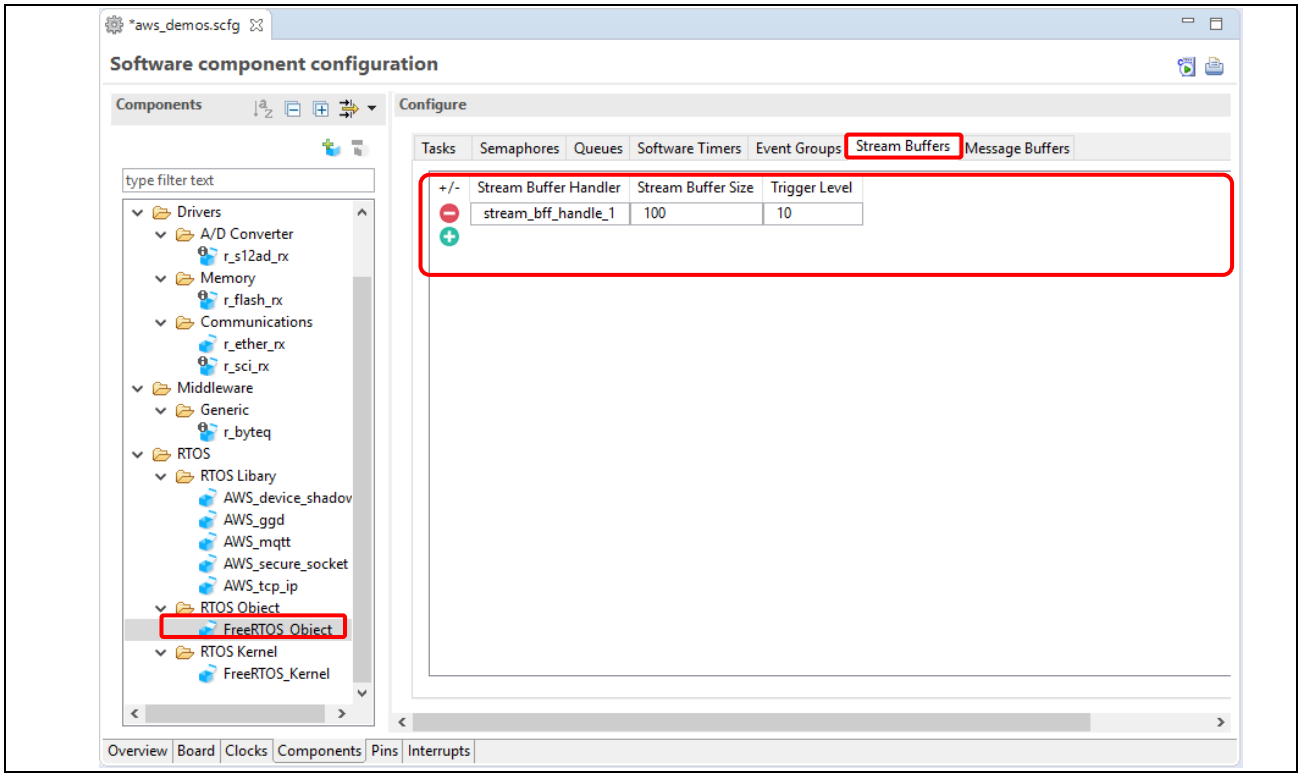


Figure 3-8 Semaphores configuration panel

- **Message buffers:**
  New **message buffers** will be created/deleted after clicking +/- buttons. Option setting will be showed in the right panel where **specific parameters** can be edited.

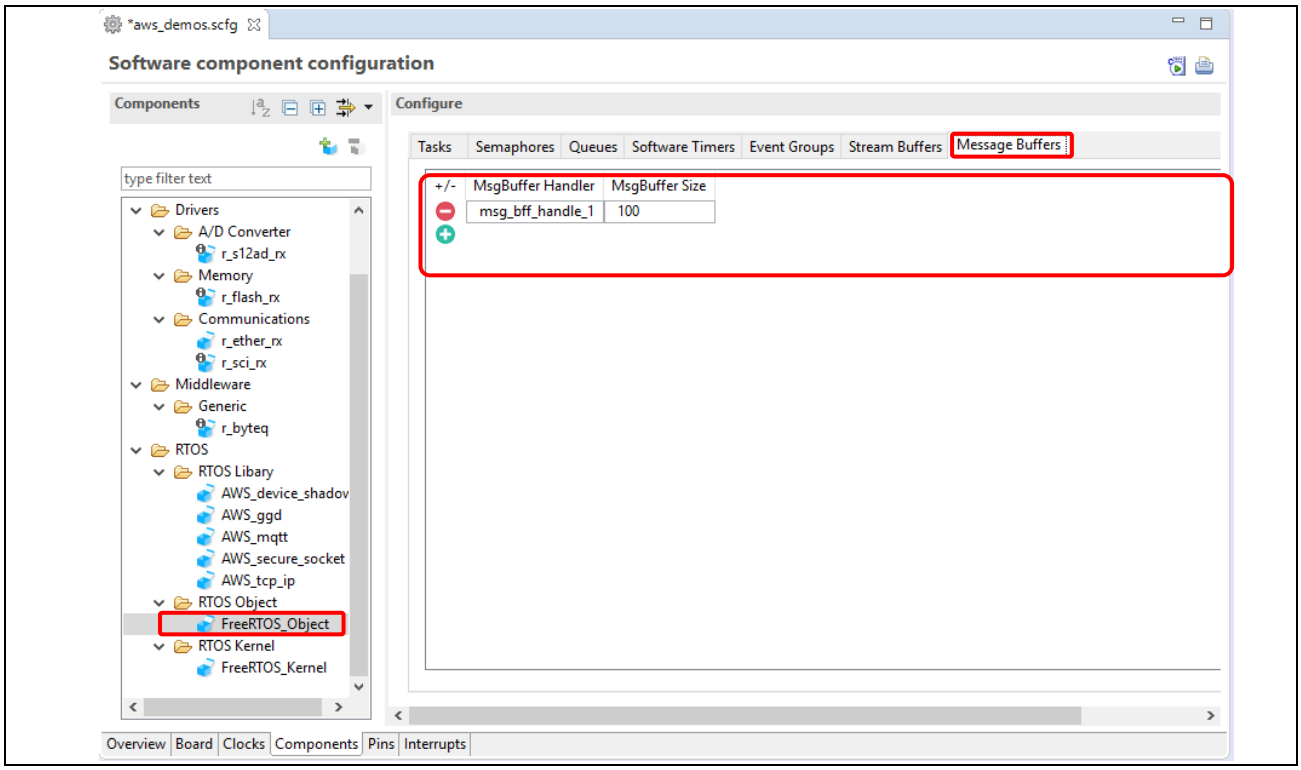


Figure 3-9 Semaphores configuration panel

3.3 Amazon FreeRTOS libraries

All supported configurations of these Amazon FreeRTOS libraries are showed as following contents:

- Device shadow: AWS_device_shadow
- Green Grass: AWS_ggd
- MQTT: AWS_mqtt
- Secure Socket: AWS_secure_socket
- TCP IP: AWS_tcp_ip

1. In [Components] tab, select [RTOS_Library] layer in the Components tree at the left panel.
2. Go to option setting in the right panel to configure these Amazon FreeRTOS libraries as Figure 3-10.

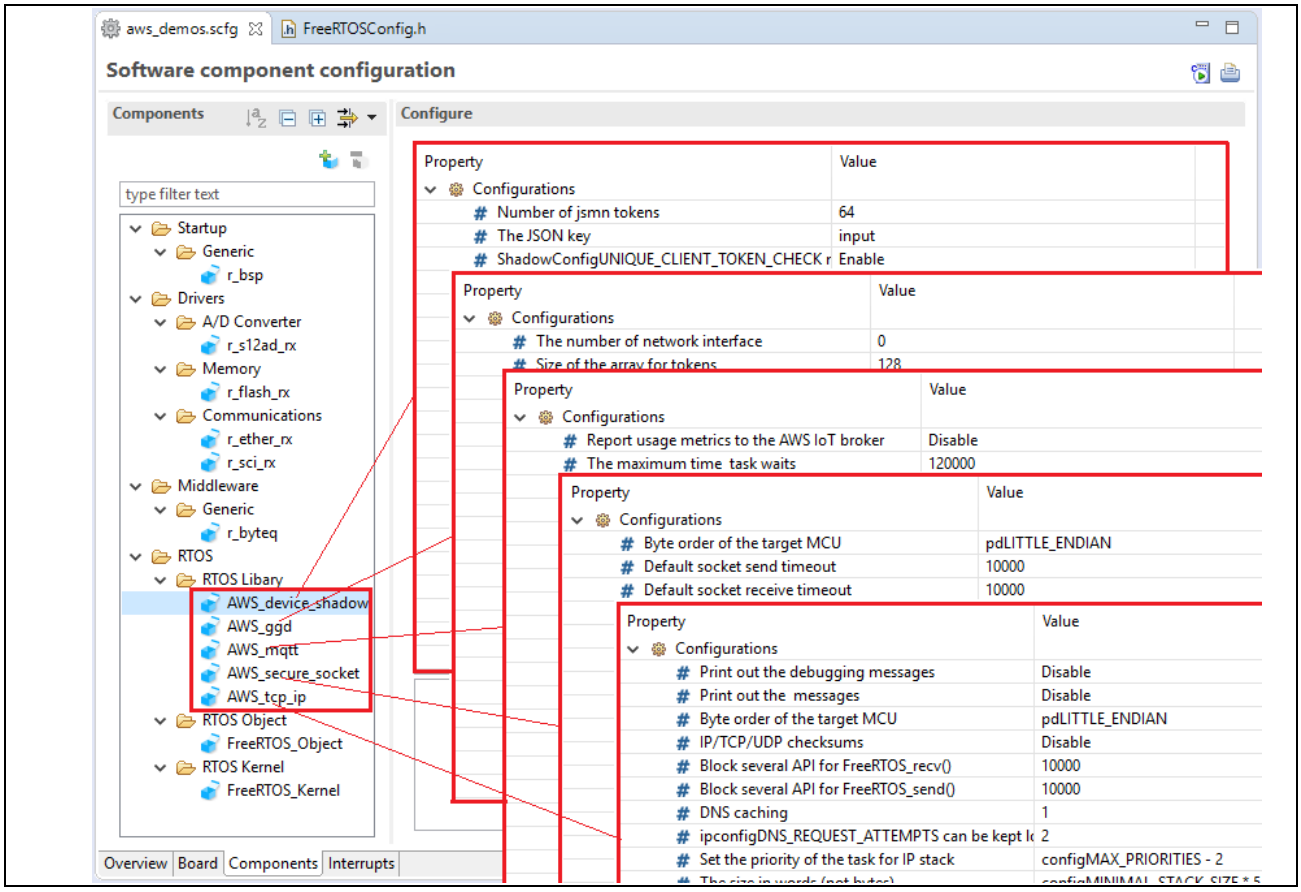


Figure 3-10 Amazon FreeRTOS libraries configuration panel

For example, Amazon TCP IP configuration:

- The corresponding parameter is displayed in the right panel for user to quickly manage the Amazon TCP IP setting. This provides all possible configuration setting options for the Amazon TCP IP setting.
- Click on any configurations option setting in the right panel to display its definition as shown in the picture below.

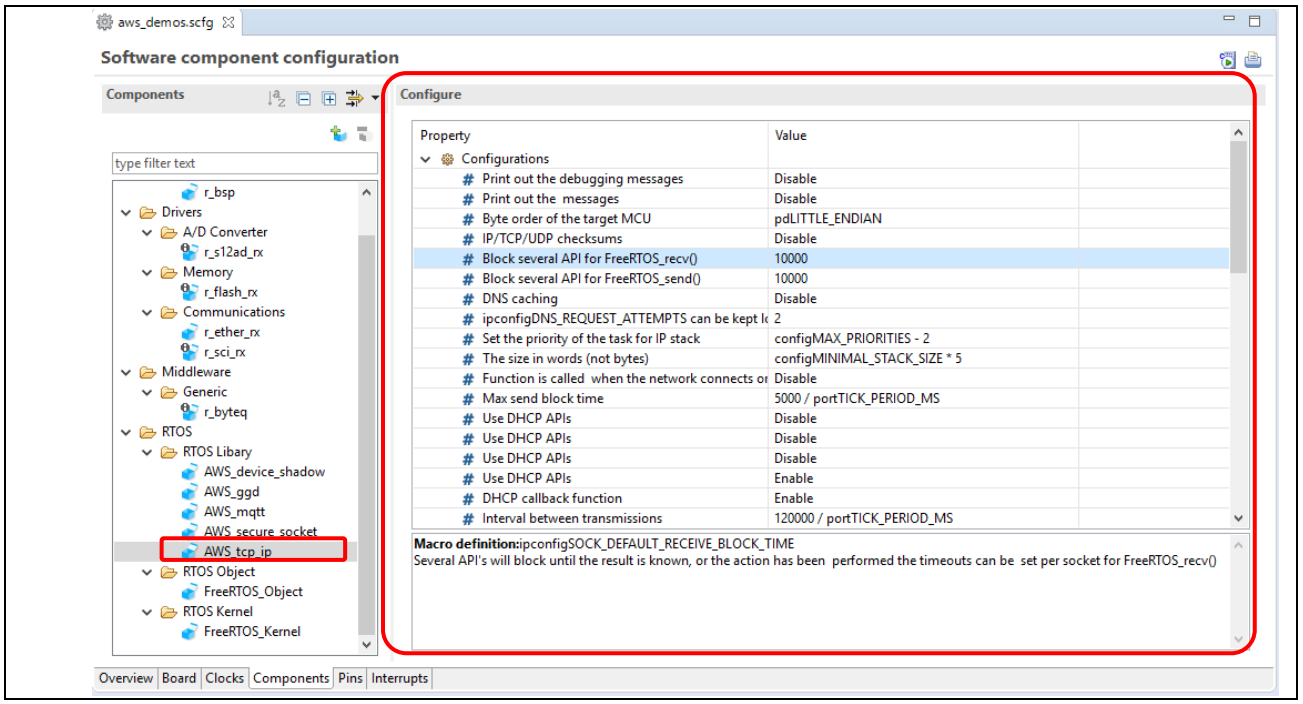


Figure 3-11 TCP IP configuration panel

4. Code generation

1. After configuring, FreeRTOS kernel, object and libraries code or middleware modules can be generated and imported to the project source folder by clicking “Code Generation” button.

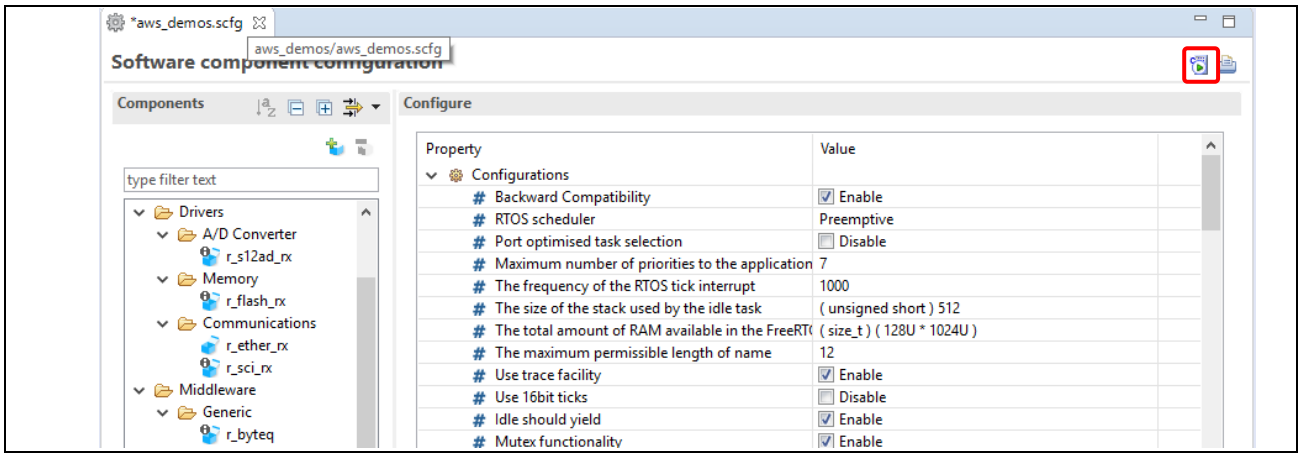


Figure 4-1 “Code Generation” button

2. The source code will be generated under renesas_code folder: [frtos_skeleton] and [frtos_startup]

5. Application development

1. Two folders are generated under renesas_code folder: [frtos_skeleton] and [frtos_startup]

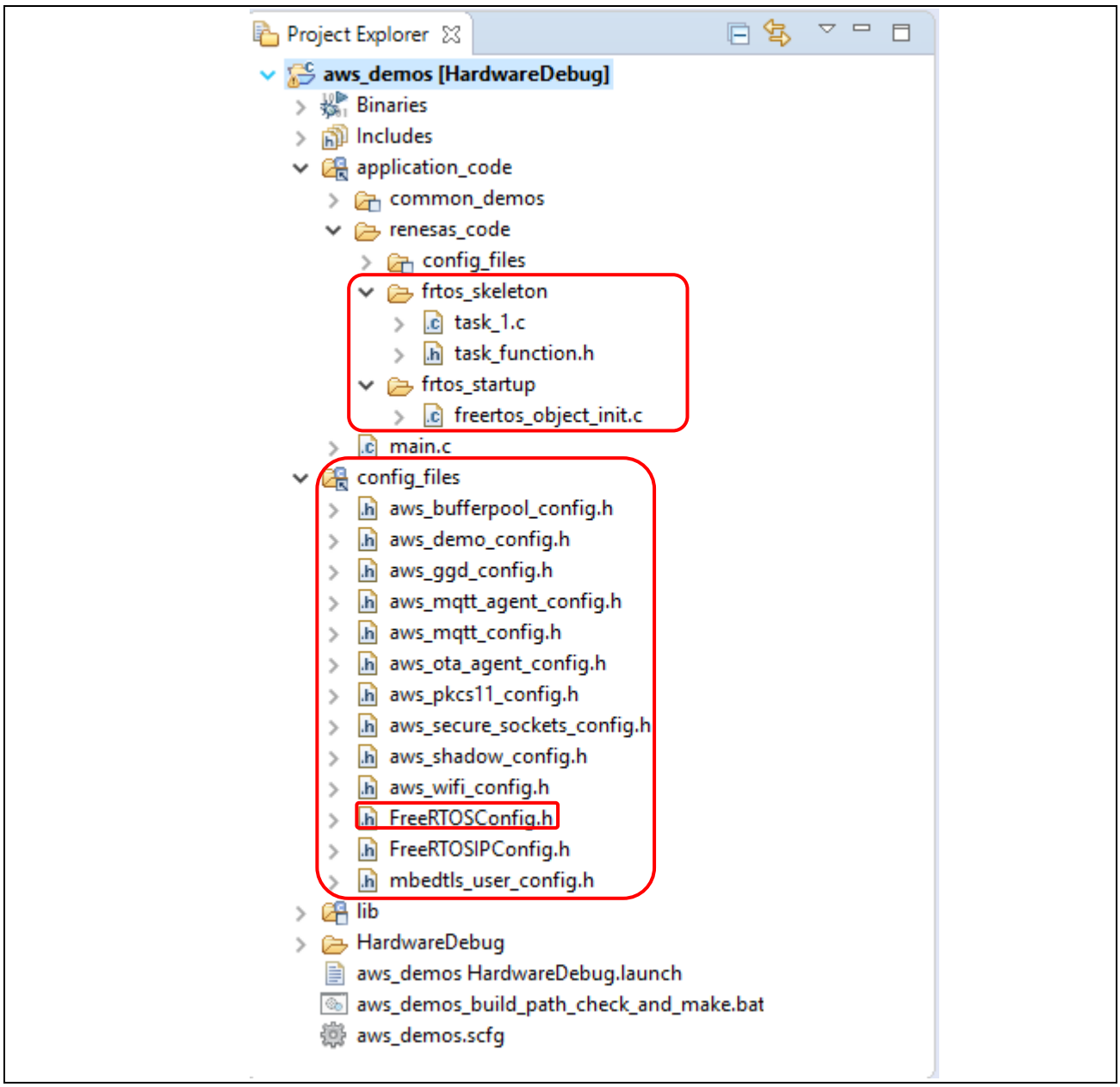


Figure 5-1 Project Explorer

frtos_skeleton includes task's skeleton where user implements own code.

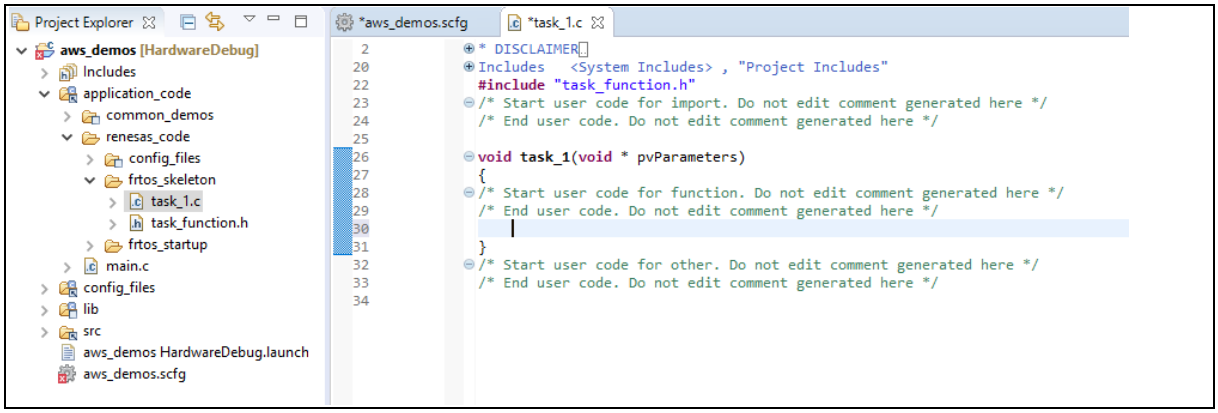


Figure 5-2 Task's skeleton where user implement application

frtos_startup includes corresponding initialization code which is created after clicking generation button 

2. The configurator automatically generates the code reflecting the configuration choices
   - Kernel: in <Amazon_demos>/config_files/ FreeRTOSConfig.h"
   - Object: in <Amazon_demos>/application_code/renesas_code/frtos_startup/ FreeRTOSConfig.h"
   - Amazon libraries: in <Amazon_demos>/config_files/". For example, Amazon_mqtt_config.h

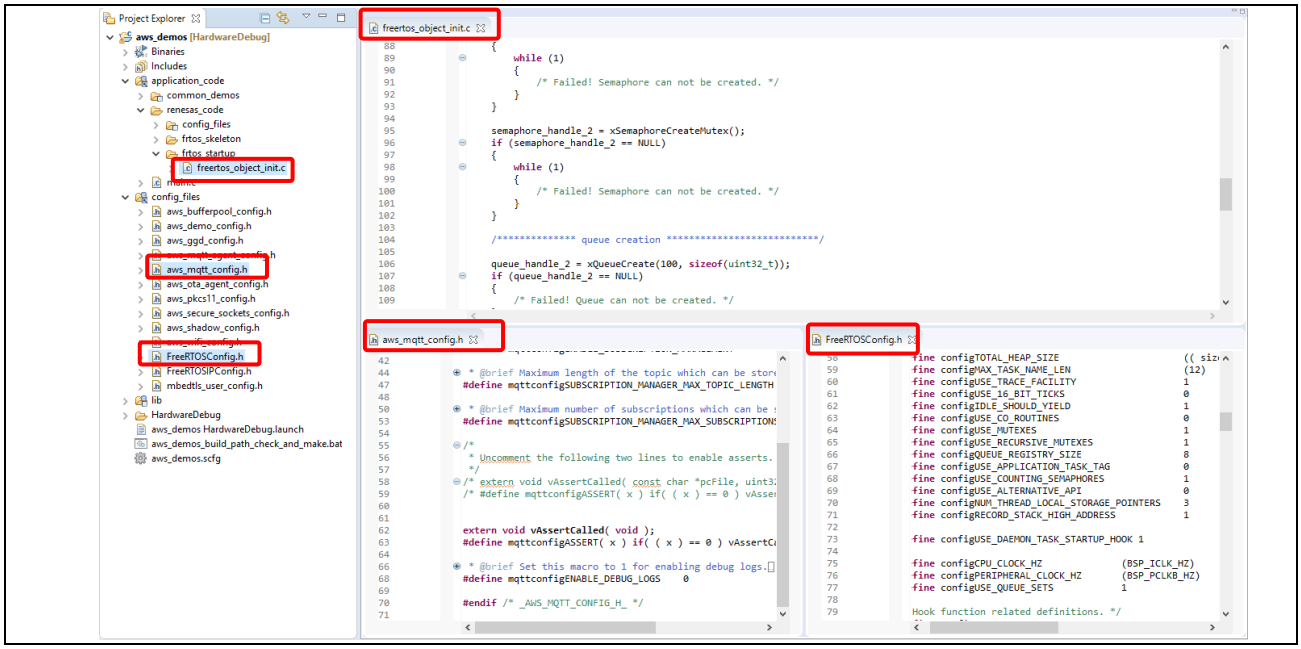


Figure 5-3 Kernel, object and Amazon libraries configuration file

6. Select the demo to run

User can select the project to run in ${PROJECT_LOC}/application_code/common_demos/source/aws_demo_runner.c by commenting out all functions except the selected one.

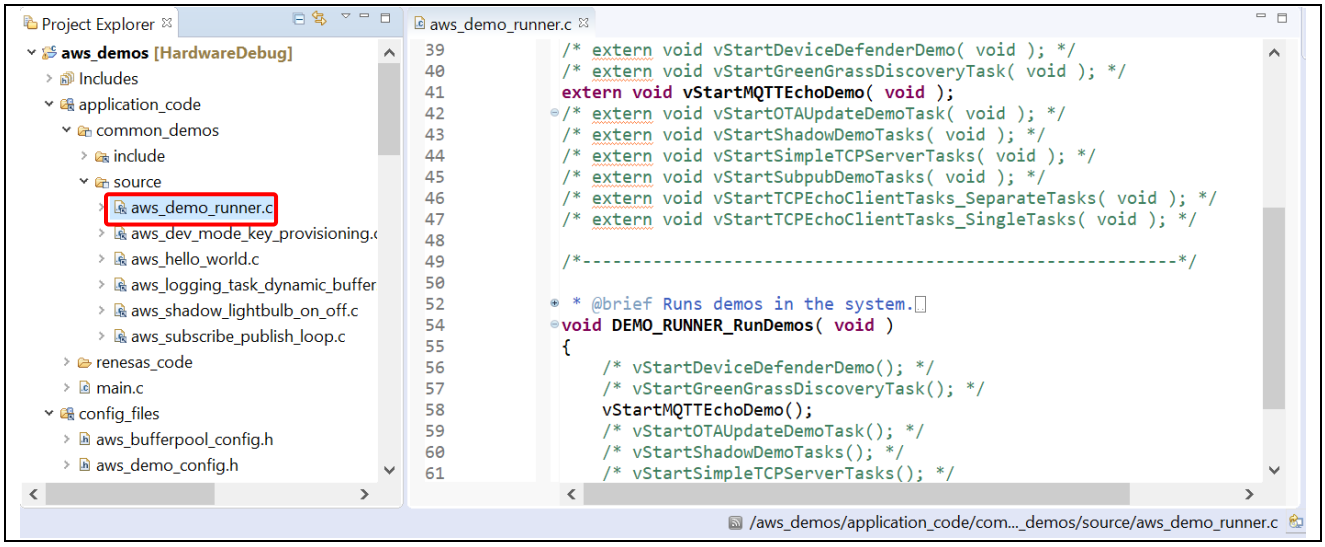


Figure 6-1 Demo runner file

The following table lists the functions in aws_demon_runner.c file and their corresponding demos.

Table 6-1 Functions and corresponding demos

Function	Demo	Description
vStartMQTTEchoDemo	MQTT Echo	Appendix 9.1.
vStartGreenGrassDiscoveryTask	Greengrass Discovery	Appendix 9.2.
vStartOTAUpdateDemoTask	OTA Update	T.B.D
vStartShadowDemoTasks	IoT Shadow	T.B.D
vStartSimpleTCPServerTasks	Simple TCP Server	Appendix 9.3
vStartTCPEchoClientTasks_SeparateTasks vStartTCPEchoClientTasks_SingleTasks	TCP Echo	Appendix 9.4
vStartDeviceDefenderDemo	Device Defender	T.B.D

7. Set up AWS

To run the Amazon FreeRTOS demos, user needs an AWS account, an IAM user with permission to access AWS IoT and Amazon FreeRTOS cloud services.

To set up AWS account and permission, please refer to
https://docs.aws.amazon.com/freertos/latest/userguide/freertos-account-and-permissions.html.

Next, user needs to register the board with AWS IoT as described at
https://docs.aws.amazon.com/freertos/latest/userguide/get-started-freertos-thing.html.

To make the demo communicate with AWS, user needs to configure the source code as described at
https://docs.aws.amazon.com/freertos/latest/userguide/freertos-configure.html.

For steps to set up the services for each demo in the package, please refer to the appendix as Table 3-1.

8. Hardware setup

User also needs to set up the specific hardware to work with the source code setup. For example:

- RSK64M:
  - J3: pin 1-2 shorted.
  - J4: pin 1-2 shorted.
  - Other pins/switches: default settings as RSK schematics.
- RSK71M:
  - J9: pin 1-2 shorted.
  - J13: pin 1-2 shorted.
  - Other pins/switches: default settings as RSK schematics.

9. Debug log

The demo outputs debug log via SCI port. If user wants to check the debug logs, connect a terminal emulator, such as Tera Term, to the serial port which is used by SCI driver.

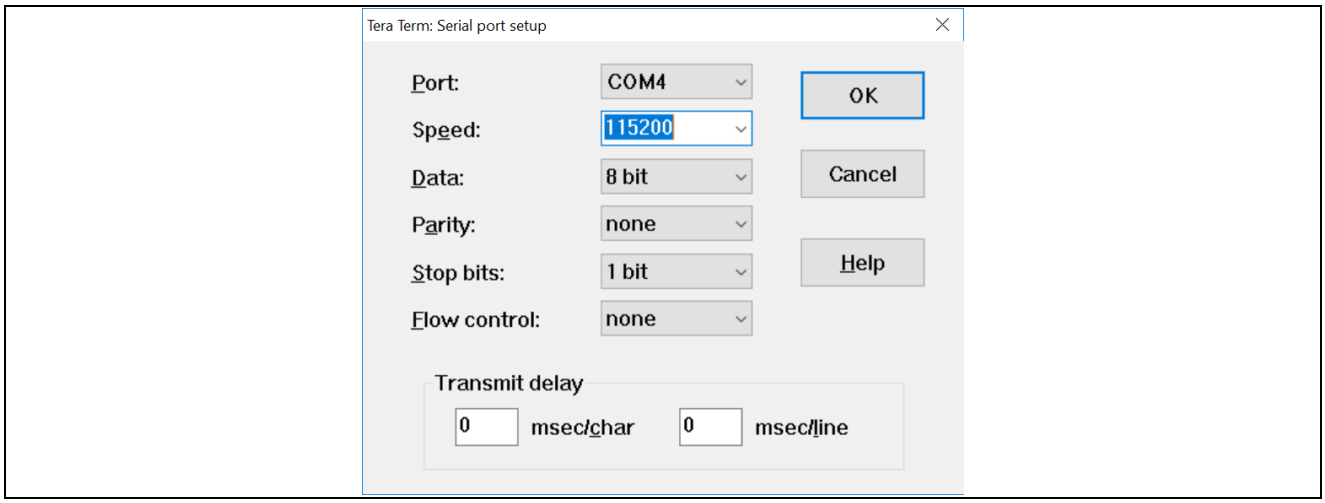


Figure 9-1 Serial port setup for terminal emulator (e.g. Tera Term)

10. Build and run

After performing all above setups, continue following steps to build and run the demo.

1. Right click on the project in Project Explorer, select “Build”.
2. Confirm that the emulator (E2/E2 Lite) is connected to the board.
3. From top menu, select [Run] → [Debug Configuration].
4. Expand Renesas GDB Hardware Debugging, and choose aws_demos HardwareDebug

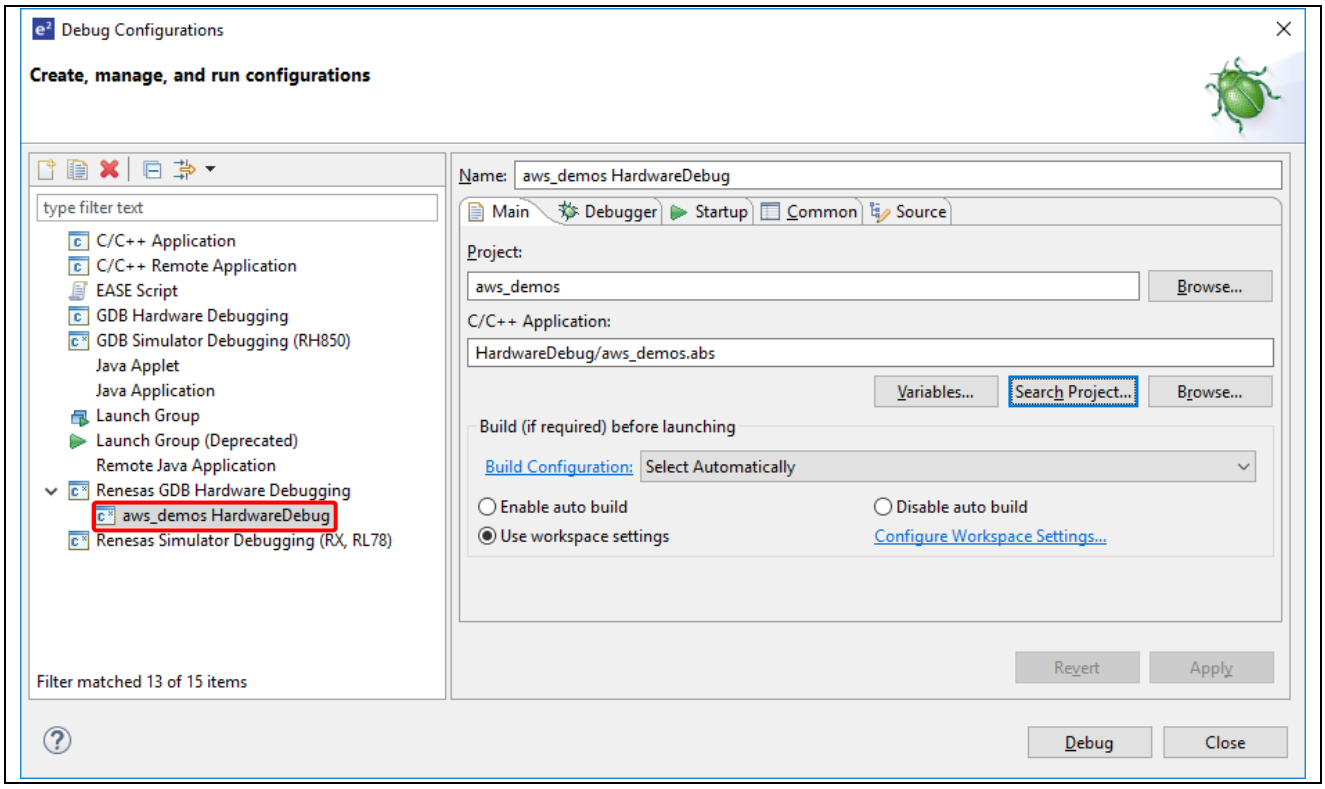


Figure 10-1 Select launch configuration

5. Choose the Debugger tab, and then choose the Connection Settings tab. Confirm that your connection settings are correct.

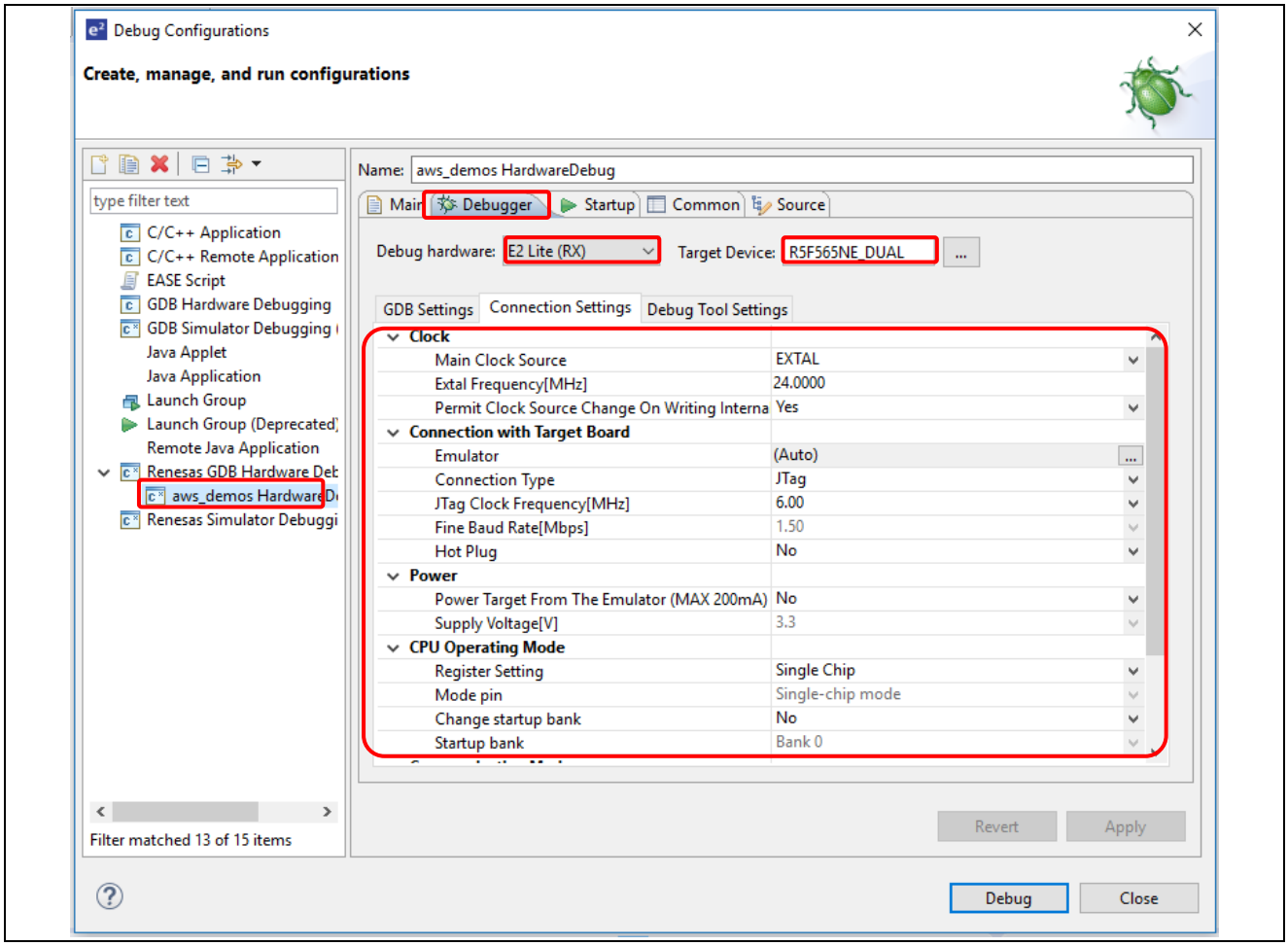


Figure 10-2 Hardware debug configuration

6. Choose Debug to download the code to your board and begin debugging.
7. e^2 studio might ask to change to Renesas Debug Perspective. Choose [Yes].
8. After the code is downloaded to the board, choose [Resume] to run the code up to the first line of the main function. Choose [Resume] again to run the rest of the code.
9. Check the debug log shown in terminal emulator.
10. Check the expected output on AWS console (if any) as described in appendix.

11. Website and support

AWS Amazon FreeRTOS forum: http://forums.aws.amazon.com.
Renesas GitHub for RX MCUs: https://github.com/renesas-rx/amazon-freertos.

12. Appendix

The Appendix contains detailed descriptions of AWS setup for each demo, as well as steps to get the expected output.

12.1 MQTT Echo

This demo application uses the Amazon FreeRTOS MQTT library to connect to the AWS Cloud and then periodically publish messages to an MQTT topic hosted by the AWS IoT MQTT broker.

12.1.1 Set up AWS MQTT client

This setup is to check the messages sent by this demo.

1. Sign into the AWS IoT console.
2. In the navigation pane, choose [Test] to open the MQTT client.
3. In Subscription topic, enter “freertos/demos/echo”, and then choose [Subscribe to topic].

Then user can see the messages that device sends to AWS Cloud.

12.2 Greengrass Discovery

The Greengrass Discovery demo publishes a series of messages to the Greengrass core, and to the AWS IoT MQTT client. In addition to the setup described in chapter 4, user needs to set up AWS IoT Greengrass permission, Greengrass group, Greengrass Core.

12.2.1 Set up environment for Greengrass Core

To set up the Greengrass Core, user need a Raspberry Pi 3 Model B+ or Model B with an 8 GB microSD card, or an Amazon EC2 instance.

To set up for Raspberry Pi, please refer to
https://docs.aws.amazon.com/greengrass/latest/developerguide/setup-filter.rpi.html.

To set up for EC2 instance, please refer to
https://docs.aws.amazon.com/greengrass/latest/developerguide/setup-filter.ec2.html.

12.2.2 Install Greengrass Core software

This procedure includes steps for configuring and starting the core software on the Greengrass Core device. These instructions are applied for Raspberry Pi, but user can use any supported device.

To configure AWS IoT Greengrass on AWS IoT, please refer to
https://docs.aws.amazon.com/greengrass/latest/developerguide/gg-config.html.

To start AWS IoT Greengrass on core device, please refer to
https://docs.aws.amazon.com/greengrass/latest/developerguide/gg-device-start.html.

12.2.3 Set up AWS IoT Greengrass permission

After setting up AWS and AWS IoT Greengrass, user needs to configure some additional permissions for AWS IoT Greengrass. User can achieve this step by referring to
https://docs.aws.amazon.com/freertos/latest/userguide/gg-demo.html and focusing on following items:

1. To set up AWS IoT Greengrass permissions
2. To create a new AWS IoT Greengrass policy
3. To attach the AWS IoT Greengrass policy to your device's certificate (Renesas RX board)

12.2.4 Add RX board to Greengrass group

In order to communicate with Greengrass Core, user needs to add the IoT thing associated with Renesas RX board to the Greengrass Group.

Note: Greengrass may not be available in some regions. If the existing device is not in the same region with new Greengrass group & core, user needs to create new IoT thing in the same region.

1. In the AWS IoT Core console, choose [Greengrass], choose [Groups], and then choose your group.
2. On the group configuration page, choose [Devices], and then choose [Add your first Device].

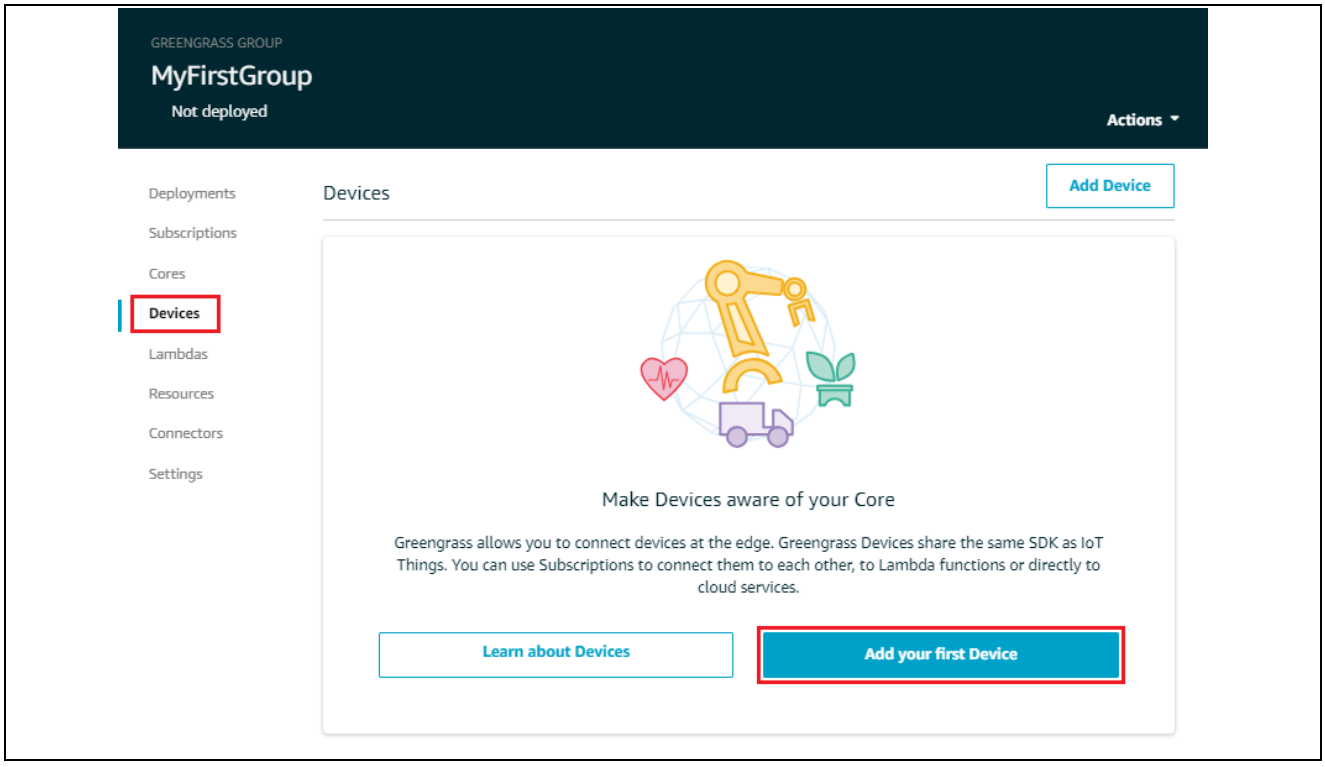


Figure 12-1 Add device to Greengrass group

3. Choose [Select an IoT thing].

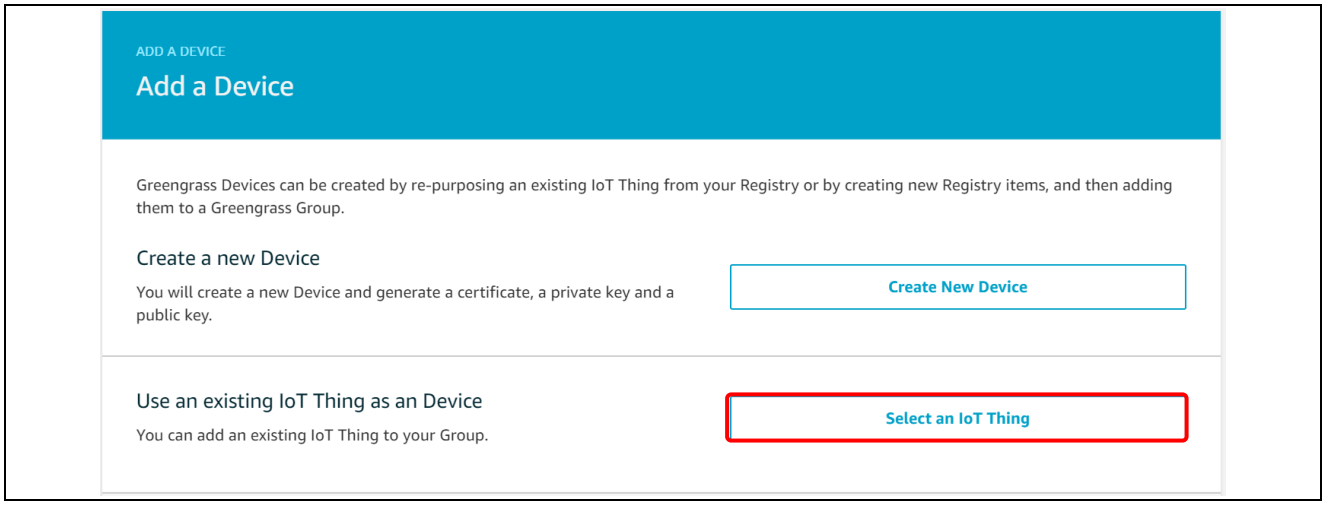


Figure 12-2 Select device to add to Greengrass group

4. Select the IoT thing configured for RX board, then click [Finish].

12.2.5 Create subscription and deploy the Greengrass group

1. On the group configuration page, choose [Subscriptions], and then choose [Add Subscription].
2. Configure the subscription.
   a. Under [Select a source], choose [Devices], and then choose the IoT thing that associates with RX board.
   b. Under Select a target, choose [Services], and then choose “IoT Cloud”.
   c. Choose Next.

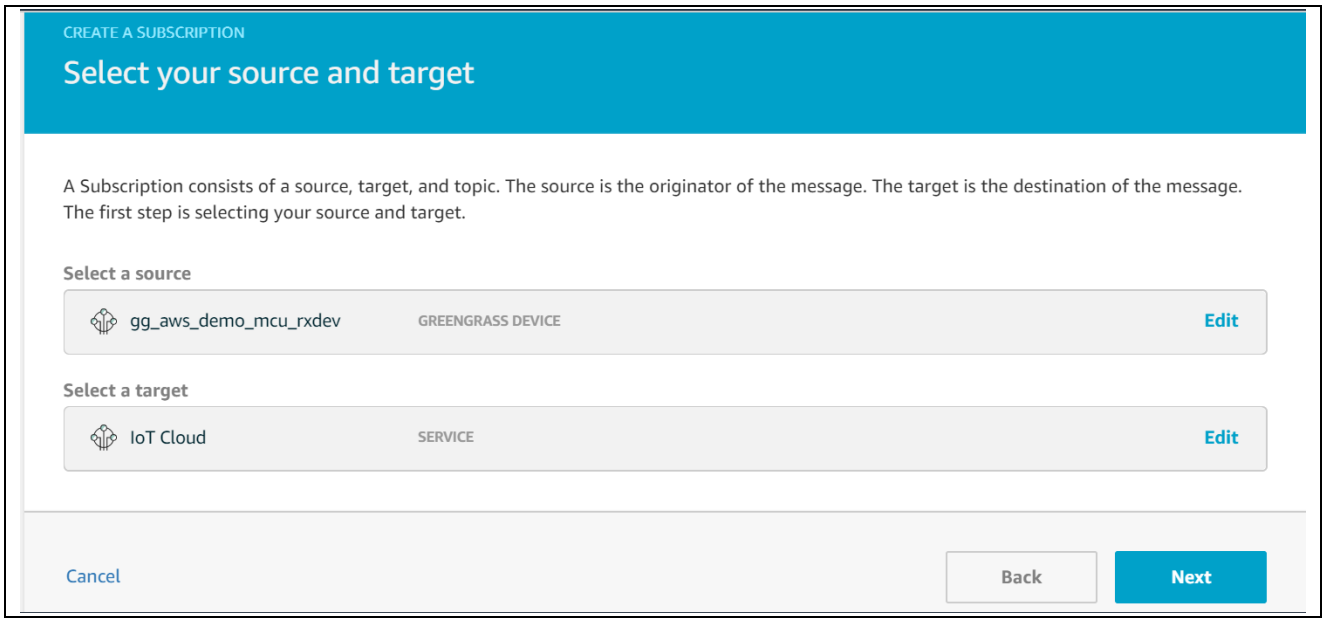


Figure 1212-3 Configure subscription

3. On the group configuration page, from [Actions], choose [Deploy].

Figure 12-4 Deploy Greengrass group

This deploys the group configuration to your AWS IoT Greengrass core device.

12.2.6 Check messages published by RX board

To view messages published by RX board to the Greengrass core, and to the AWS IoT MQTT client, please refer to chapter 8.1.1., but replace the subscription topic by “freertos/demos/ggd”.

After build and run the demo, user can see the published messages in MQTT client.

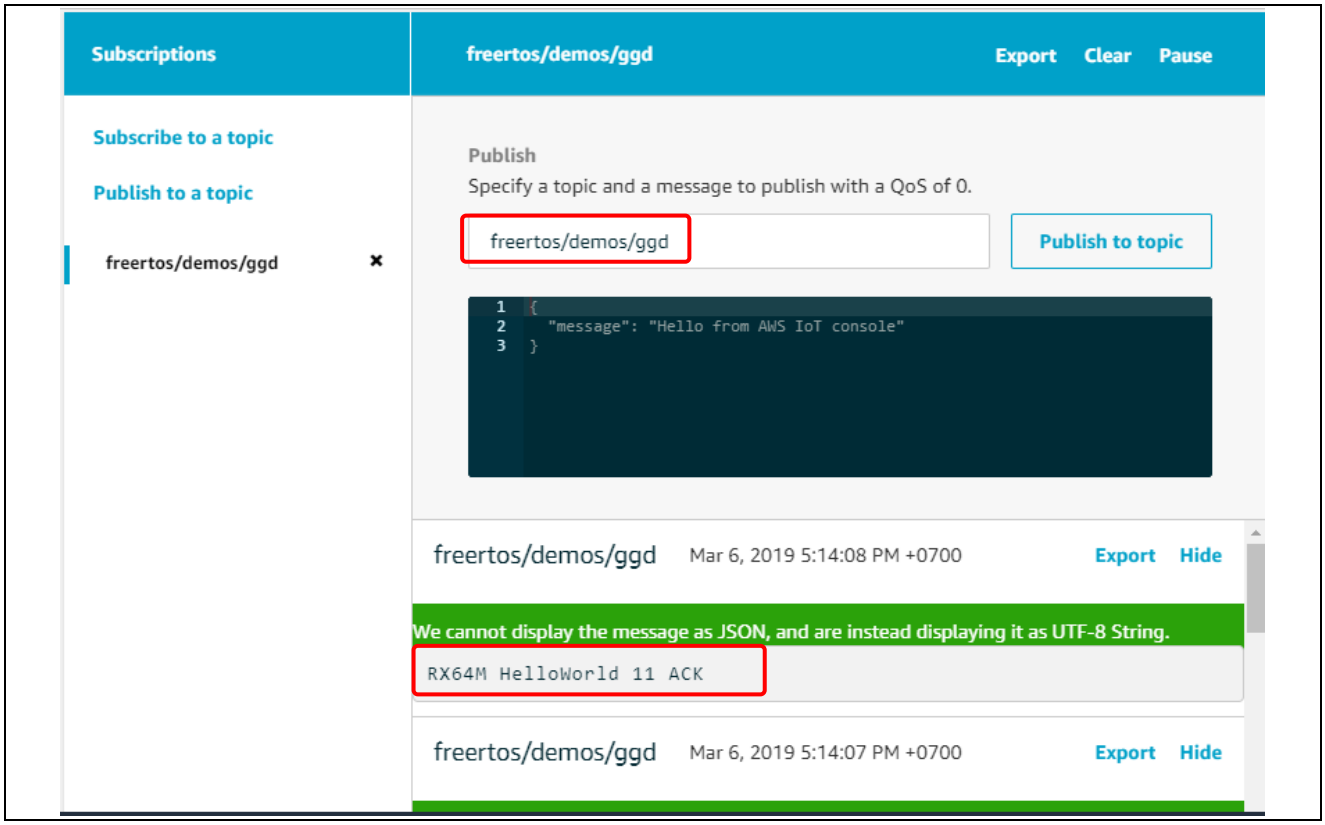


Figure 12-5 Confirm messages sent by Greengrass Discovery demo

12.3 Simple TCP Server

This demo uses FreeRTOS+TCP to create an echo server that listens for echo requests on the standard echo protocol.

12.3.1 Include demo to the build

This demo is not included in the project by default. To include it, please follow below steps.

1. In Project Explorer, right click on the folder ${PROJECT_LOC}/application_code/common_demos/source and select [New] → [File].

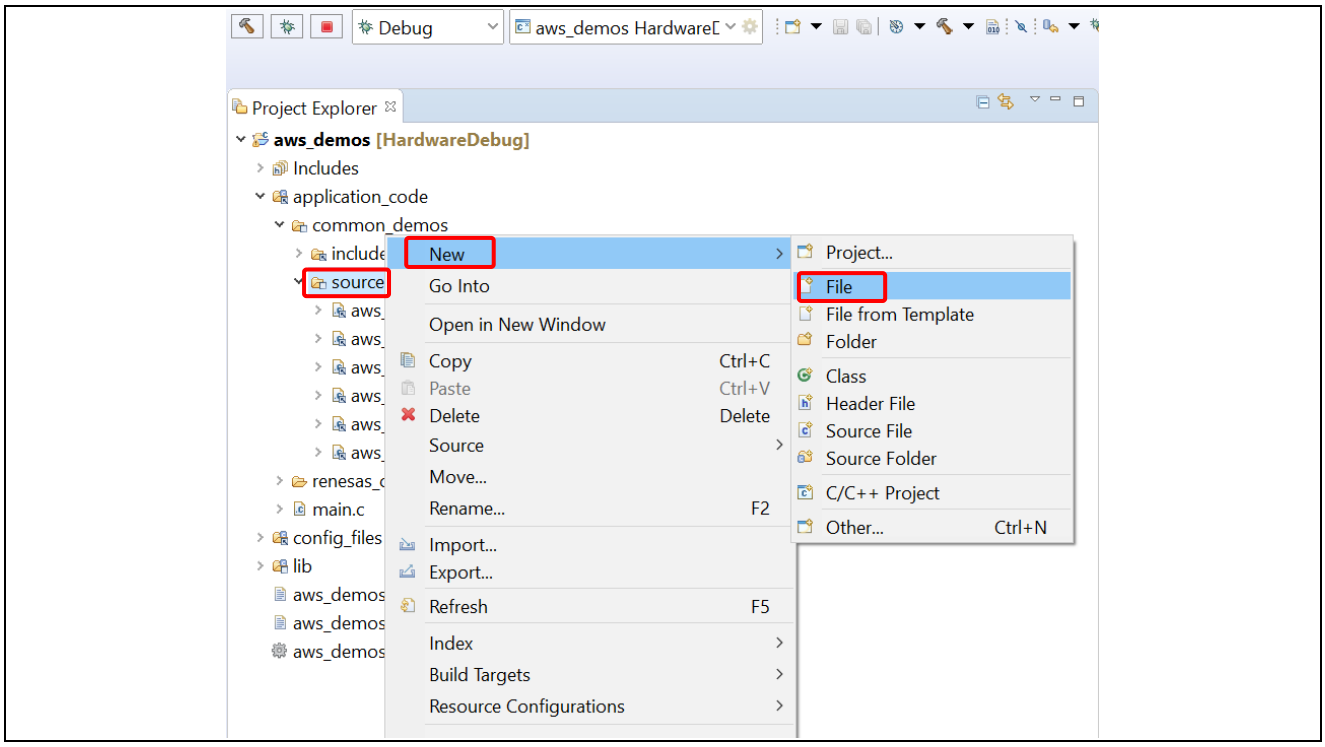


Figure 12-6 Add source for TCP Server demo

2. In the new file dialog, click [Advanced], check “Link to file in the file system”, and input “AFR_HOME \demos\common\tcp\aws_simple_tcp_echo_server.c” to the text box. Click [Finish].

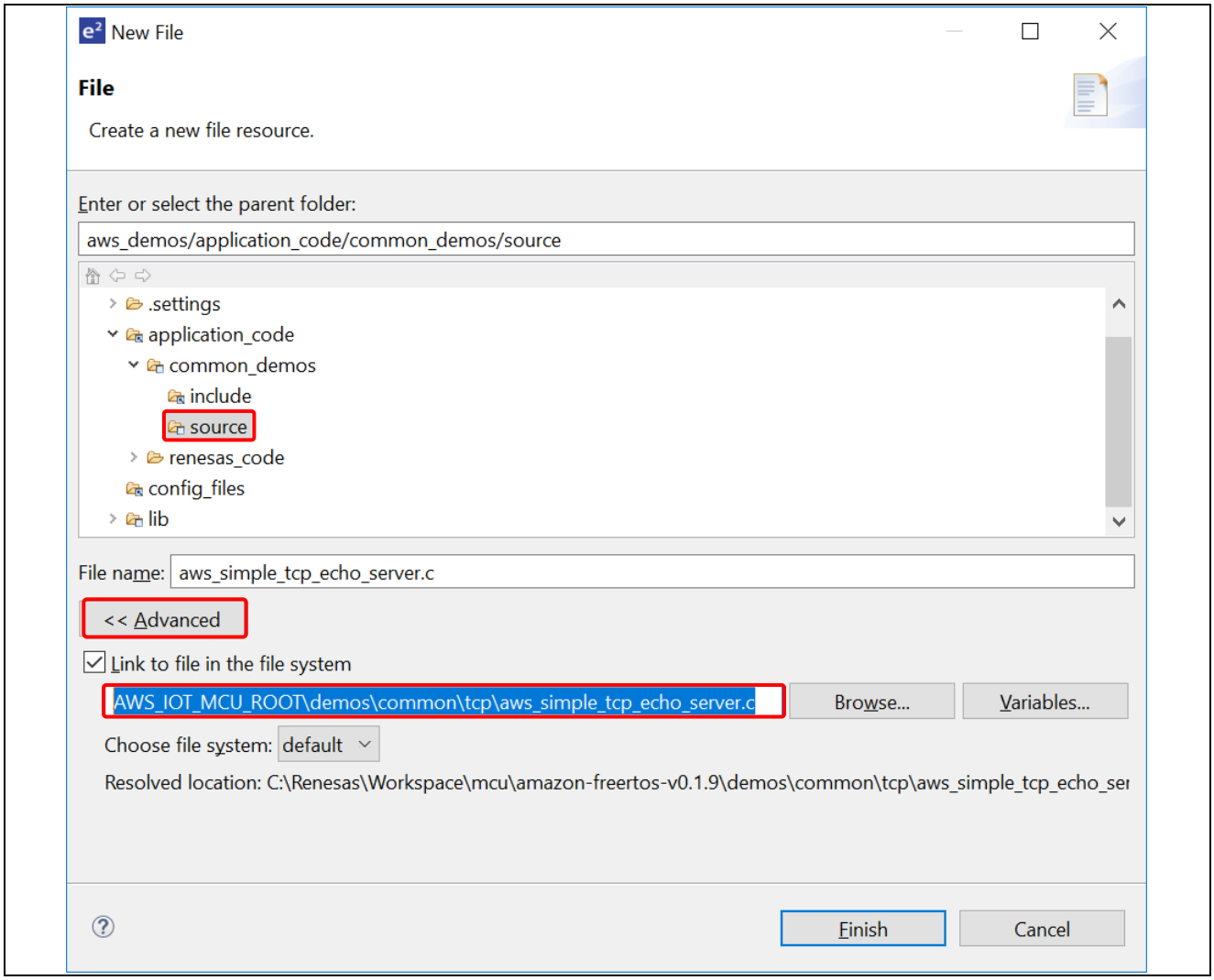


Figure 12-7 Add new source file to the TCP Server demo

3. Select Simple TCP Server demo to run as described in chapter 3, then re-build the source code.

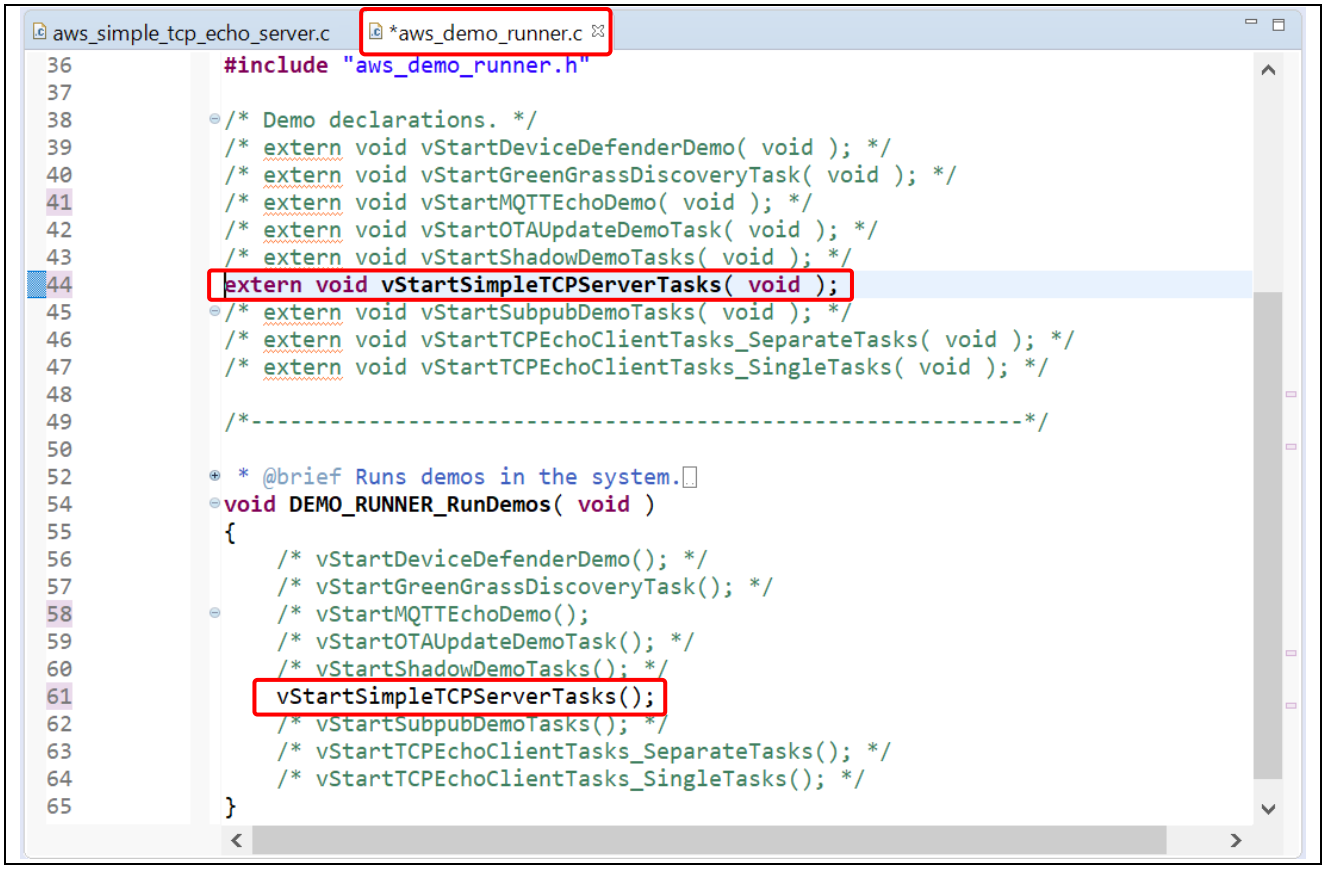


Figure 12-8 Select Simple TCP Server demo to run

12.3.2 Configure EchoTool

It's necessary to send echo requests to the server (created by the demo) manually. The third party EchoTool utility can be used for this purpose.

User can build the tool from the source code on GitHub or download a pre-built executable. Please refer to this link for detailed information: https://github.com/PavelBansky/EchoTool.

Follow below steps to configure the tool.

1. Check the value of configTCP_ECHO_CLIENT_PORT

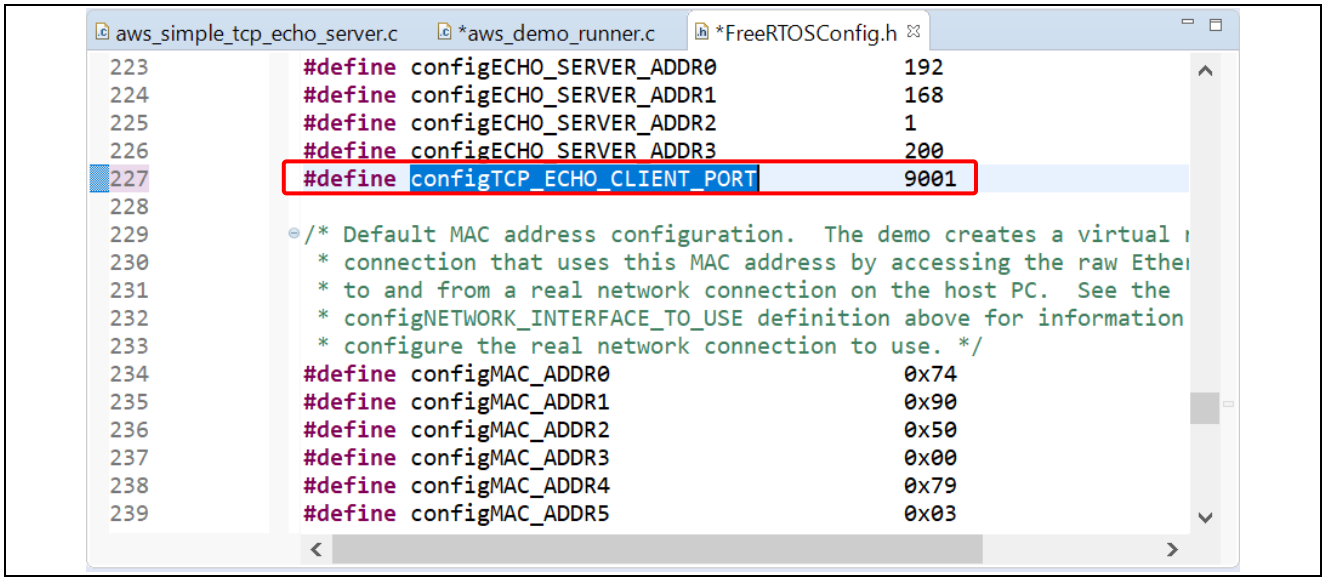


Figure 12-9 Check echo port

2. Run the demo. Check the IP address of the board assigned by DHCP in debug terminal.

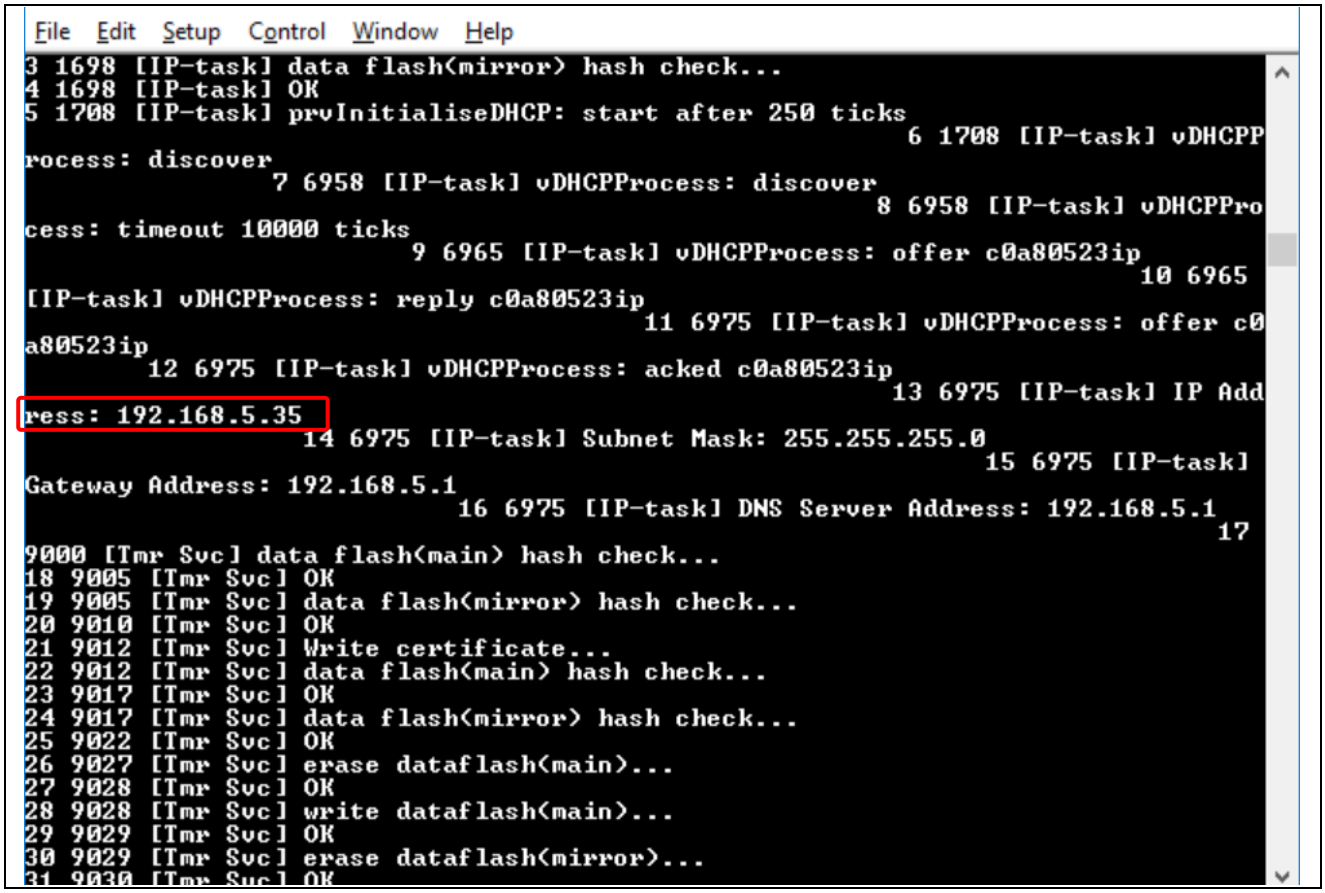


Figure 12-10 IP address assigned by DHCP

3. Run the EchoTool to send echo requests to the demo using port and IP as confirmed above: `echotool <ip_address> /p tcp /r <echo_client_port> /n 0`.

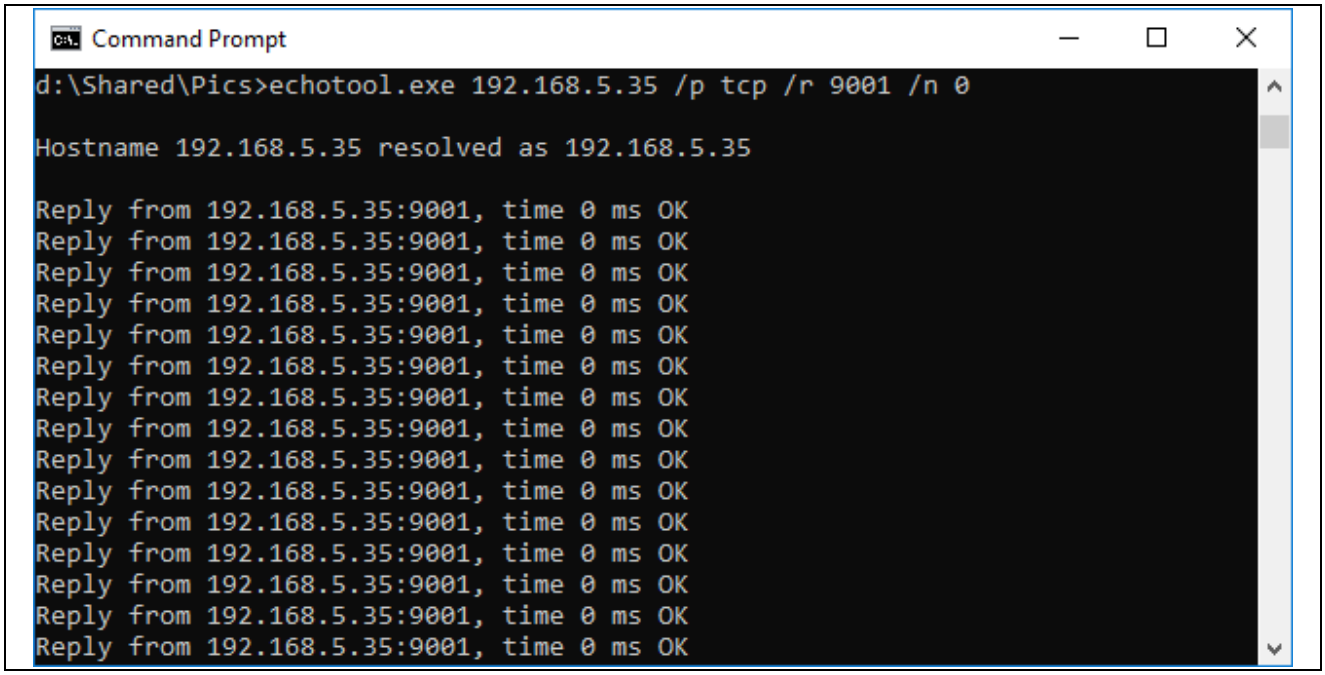


Figure 12-11 Run EchoTool

4. Confirm that the messages “Reply from…” appears. This shows that the requests from EchoTool are replied by the demo.

12.4 TCP Echo Client

This demo creates FreeRTOS tasks that send TCP echo requests to an external echo, then wait to receive the echo reply. There are 2 examples for this demo: the 1st one uses the same RTOS task to both send echo requests and listen for echo replies (“single task”); the 2nd one uses the same TCP socket from two different RTOS tasks – one RTOS task sends the echo request and another RTOS tasks receives the echo reply (“separate tasks”).

To run this demo, please follow below steps.

1. Select the demo to run: “vStartTCPEchoClientTasks_SingleTasks” for 1st example, or “vStartTCPEchoClientTasks_SeparateTasks” for 2nd example. Please note that only 1 example can be run at once.

Figure 12-12 Select the TCP Echo client demo

2. Build and run the demo.

3. Check the TCP port and IP address as chapter 9.3.2, then run the EchoTool in server mode using information about port and IP: `echotool <IP_address> /p tcp /s <port>`
4. Check the debug message on debug terminal and debug log of the EchoTool to confirm that send & receive is OK.

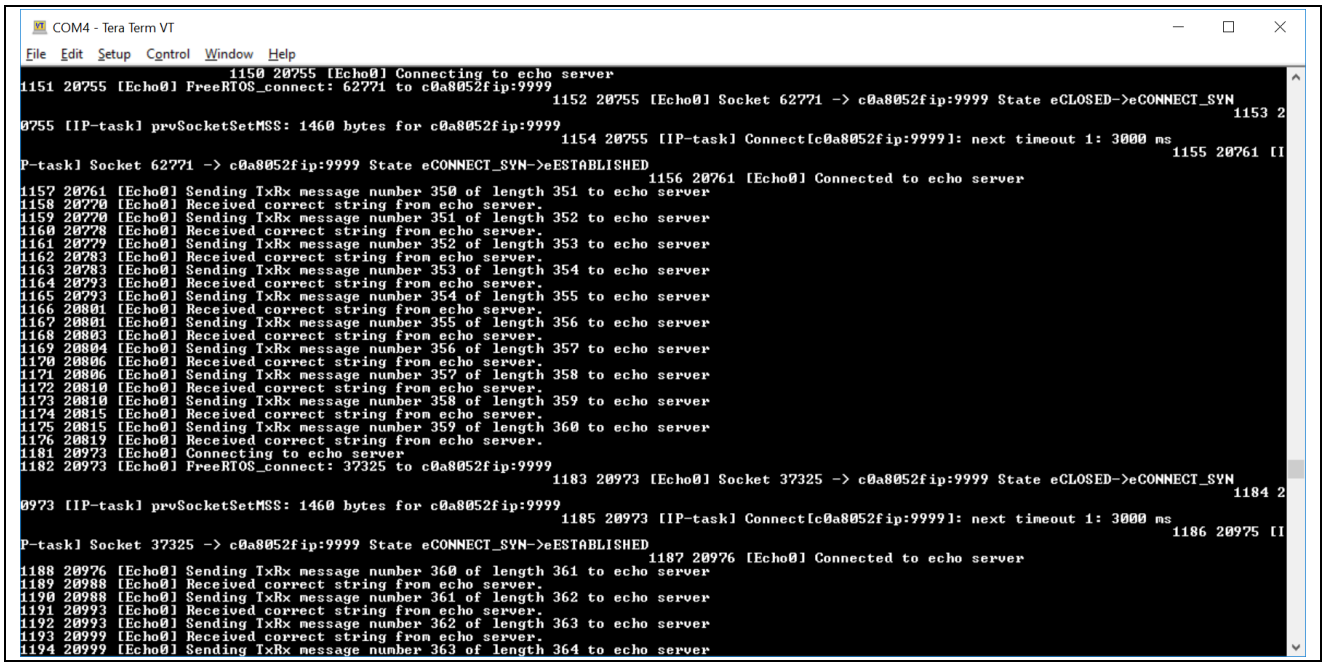


Figure 12-13 Debug terminal shows messages from the demo

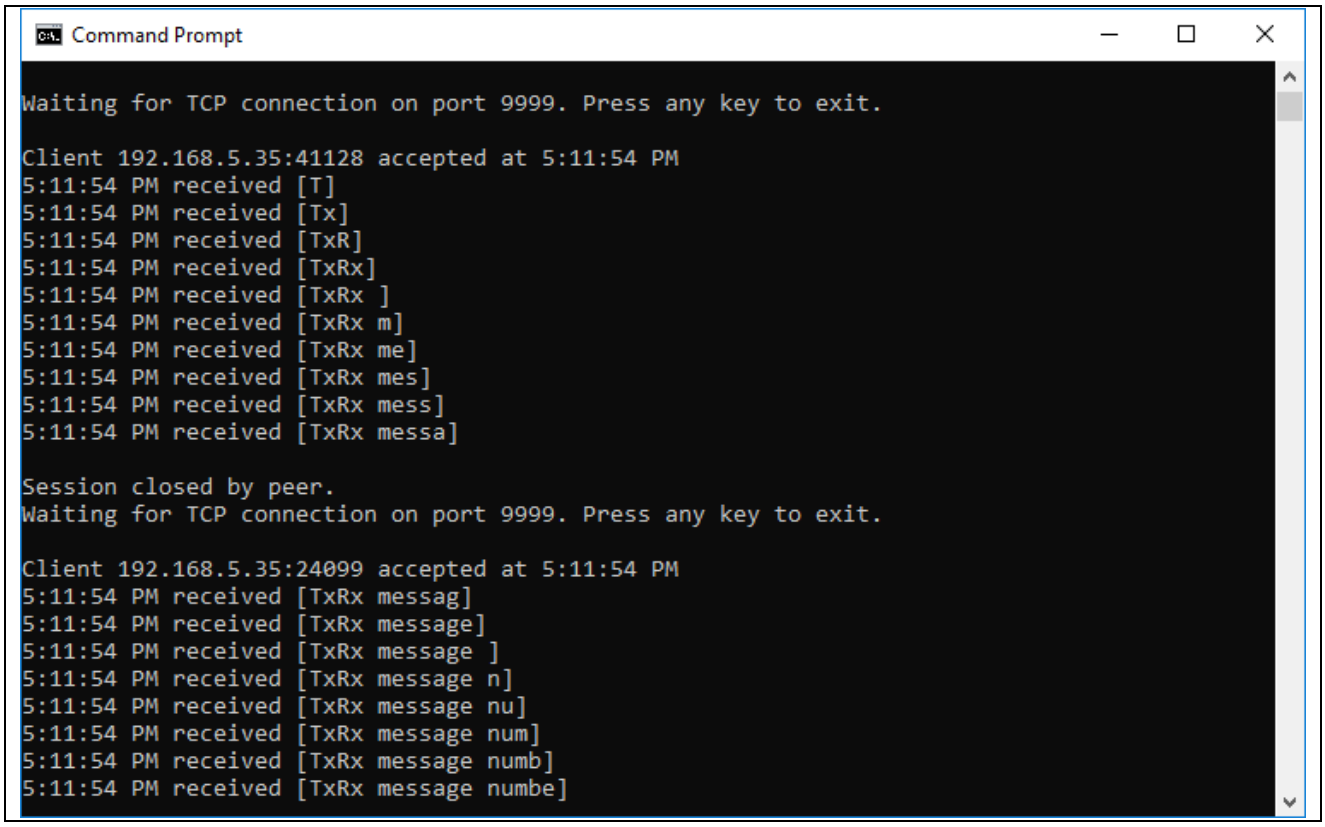


Figure 12-14 Debug log from EchoTool

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug. 20, 2019	-	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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 not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product 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.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

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