Getting started with the RX72N Envision Kit

This tutorial provides instructions for getting started with the Renesas Envision Kit for RX72N. If you do not have the Renesas REK for RX72N, visit the AWS Partner Device Catalog, and purchase one from our partners.

Before you begin, you must configure AWS IoT and your FreeRTOS download to connect your device to the AWS Cloud. See First steps for instructions.

> **Note**
> RX72N Envision Kit currently does not support “Quick Connect” workflow. Please return to this “Getting started guide” after completing the instructions in “Step 1: Setting up your AWS account and permissions” and “Step 2: Registering your MCU board with AWS IoT”.

In this tutorial, the path to the FreeRTOS download directory is referred to as `freertos`.

Overview

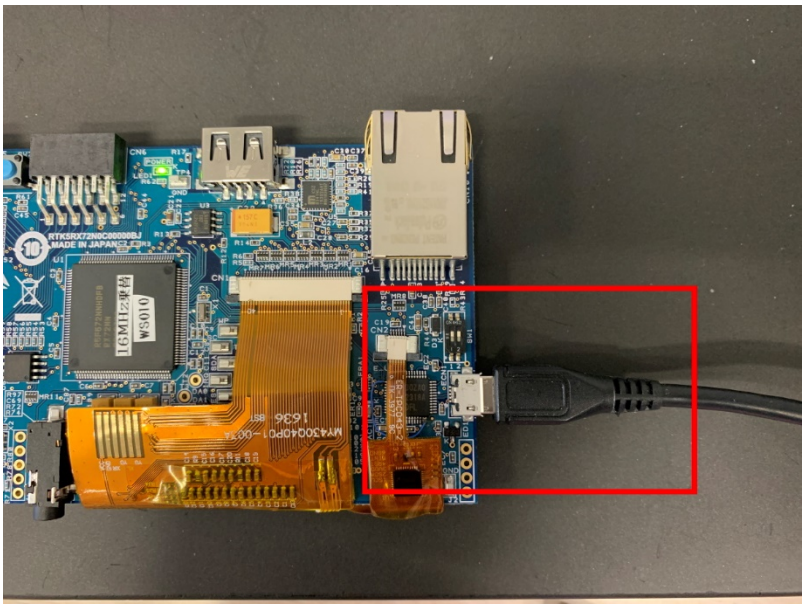
This tutorial contains instructions for the following getting started steps:

1. Connecting your board to a host machine.
2. Confirm REK for RX72N performance by confirming factory image behavior.
3. Installing software on the host machine for developing and debugging embedded applications for your microcontroller board.
4. Cross compiling a FreeRTOS demo application to a binary image.
5. Loading the application binary image to your board, and then running the application.

Set up the Renesas hardware

To confirm functionality of the REK for RX72N

1. Connect ECN1 (USB Micro-B) on the REK for RX72N to power source USB port (PC,etc).



2. Confirm SPORT GAMES starting on display.

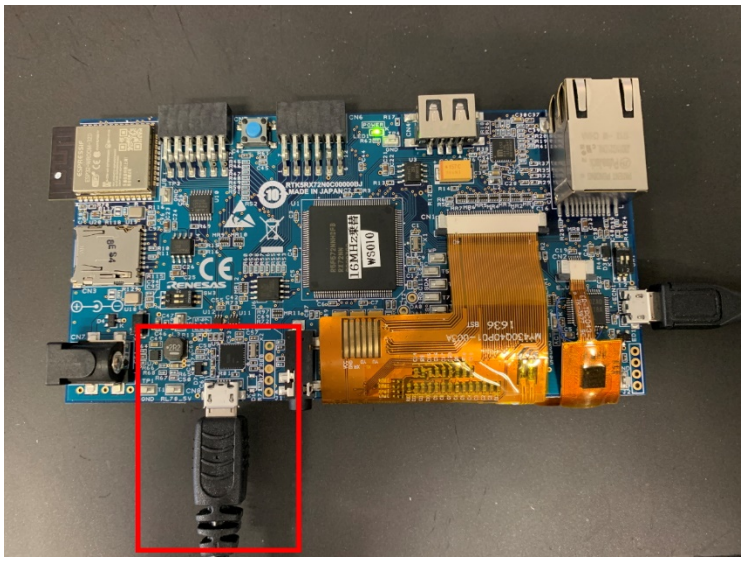
3. User can select 4 types of Games with sliding on title display, and user can start by pushing [PLAY] button.

To set up the REK for RX72N and (onboard) E2 Lite Debugger

1. REK for RX72N is equipped with an onboard E2 Lite emulator, emulator usage is controlled by switch SW1.
2. SW1-2 is turned ON by default, user needs to turn SW1-2 OFF to enable emulator use.
3. Connect ECN1 (USB Micro-B) on the REK for RX72N to power source USB port (PC,etc), LED1 illuminates when power is supplied to the board.
4. Connect your computer to the USB-to-serial port (CN8) on the REK for RX72N, the green ‘ACT’ LED flashes when E2 Lite debugger is connected.
5. After the debugger on REK for RX72N is connected to your host machine, E2 Lite debugger drivers begin installing.

   Note that Windows 7, 8, and 10 administrator privileges are required to install the drivers.

6. Connect a router or internet-connected Ethernet port to the Ethernet port (CN10) on the REK for RX72N.



Set up your development environment

To set up FreeRTOS configurations for the REK for RX72N, use the Renesas e²studio IDE and CC-RX compiler.

> **Note**
> The Renesas e²studio IDE and CC-RX compiler are only supported on Windows 7, 8, and 10 operating systems.

To download and install e²studio

1. Go to the Renesas e²studio installer download page, and download the offline installer.
2. You are directed to a Renesas Login page.

   If you have an account with Renesas, enter your username and password and then choose **Login**.

   If you do not have an account, choose **Register now**, and follow the first registration steps. You should receive an email with a link to activate your Renesas account. Follow this link to complete your registration with Renesas, and then login to Renesas.
3. After you log in, download the e²studio installer to your computer.
4. Open the installer and follow the steps to completion.

For more information, see the e²studio on the Renesas website.

To download and install the RX Family C/C++ Compiler Package

1. Go to the RX Family C/C++ Compiler Package download page, and download the v3.02.00 package.
2. Open the executable and install the compiler.

For more information, see the C/C++ Compiler Package for RX Family on the Renesas website.

> **Note**
> The compiler is available free for evaluation version only and valid for 60 days. On the 61st day, you need to get a License Key. For more information, see Evaluation Software Tools.

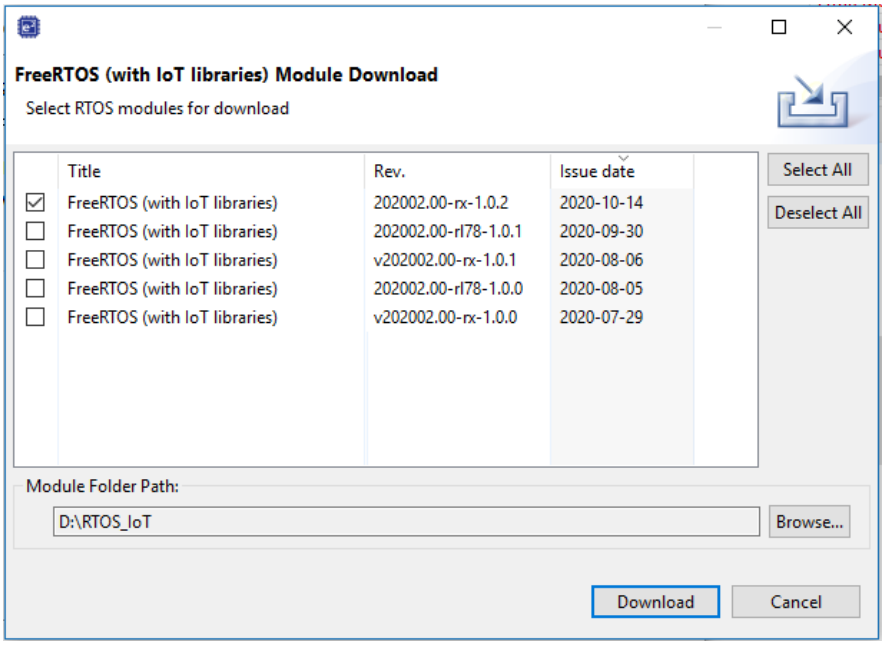
Build and run FreeRTOS samples

Now that you have configured the demo project, you are ready to build and run the project on your board.

Build the FreeRTOS Demo in e²studio

To download and build the demo in e²studio

1. Launch e²studio from the Start menu.
2. On the **Select a directory as a workspace** window, browse to the folder that you want to work in, and choose **Launch**.
3. The first time you open e2studio, the **Toolchain Registry** window opens. Choose **Renesas Toolchains**, and confirm that `CC-RX v3.02.00` is selected. Choose **Register**, and then choose **OK**.
4. If you are opening e²studio for the first time, the **Code Generator Registration** window appears. Choose **OK**.
5. The **Code Generator COM component register** window appears. Under **Please restart e²studio to use Code Generator**, choose **OK**.
6. The **Restart e²studio** window appears. Choose **OK**.
7. e²studio restarts. On the **Select a directory as a workspace** window, choose **Launch**.
8. On the e²studio welcome screen, choose the **Go to the e²studio workbench** arrow icon.
9. Right-click the **Project Explorer** window, and choose **Import**.
10. In the import wizard, choose **General**, **Renesas GitHub FreeRTOS (with IoT libraries) Project**, and the choose **Next**.
11. Choose **Browse** to specify a folder to copy downloaded RTOS content in order to import project.
12. In RTOS version setting, choose **Check for more version…** to see a list of all supported RTOS version. On the **FreeRTOS (with IoT libraries) Module Download** window, select the FreeRTOS version (recommended: 202002.00-rx-1.0.2) you want to work on by clicking the checkbox, then choose **Download**.



13. Once download is completed, choose **Next** in the **Renesas GitHub FreeRTOS (with IoT libraries) Project** window.
14. If you are *not* using an empty folder, the **Copy Resources** warning message appears. Choose **Yes**.
15. Choose the project aws_demos (`${FOLDER_DIR}/projects/renesas/rx72n-envision-kit/e2studio/aws_demos`), then choose **Finish**.
16. From **Project** menu, choose **Build All**.

To import and build the demo in e²studio

17. Launch e²studio from the Start menu.
18. On the **Select a directory as a workspace** window, browse to the folder that you want to work in, and choose **Launch**.
19. The first time you open e2studio, the **Toolchain Registry** window opens. Choose **Renesas Toolchains**, and confirm that `CC-RX v3.02.00` is selected. Choose **Register**, and then choose **OK**.
20. If you are opening e²studio for the first time, the **Code Generator Registration** window appears. Choose **OK**.
21. The **Code Generator COM component register** window appears. Under **Please restart e²studio to use Code Generator**, choose **OK**.
22. The **Restart e²studio** window appears. Choose **OK**.
23. e²studio restarts. On the **Select a directory as a workspace** window, choose **Launch**.
24. On the e²studio welcome screen, choose the **Go to the e²studio workbench** arrow icon.
25. Right-click the **Project Explorer** window, and choose **Import**.
26. In the import wizard, choose **General**, **Existing Projects into Workspace**, and the choose **Next**.
27. Choose **Browse**, locate the directory `projects/renesas/rx72n-envision-kit/e2studio/aws_demos`, and the choose **Finish**.
28. From **Project** menu, choose **Build All**.

The build console issues a warning message that the License Manager is not installed. You can ignore this message unless you have a license key for the CC-RX compiler. To install the License Manger, see the License Manager download page.

Run the FreeRTOS project

To run the project in e²studio

1. Confirm that you have connected your computer to the USB-to-serial port (CN8) on your REK for RX72N.
2. From the top menu, choose **Run**, **Debug Configurations…**.
3. Expand **Renesas GDB Hardware Debugging**, and choose **aws_demos HardwareDebug**.
4. Choose the **Debugger** tab, and then choose the **Connection Settings** tab. Confirm that your connection settings are correct.
5. Choose **Debug** to download the code to your board and begin debugging.

You might be prompted by a firewall warning for `e2-server-gdb.exe`. Check **Private networks, such as my home or work network**, and then choose **Allow access**.

6. e²studio might ask to change to **Renesas Debug Perspective**. Choose **Yes**.

   The flashing green ‘ACT’ LED on the REK for RX72N illuminates.

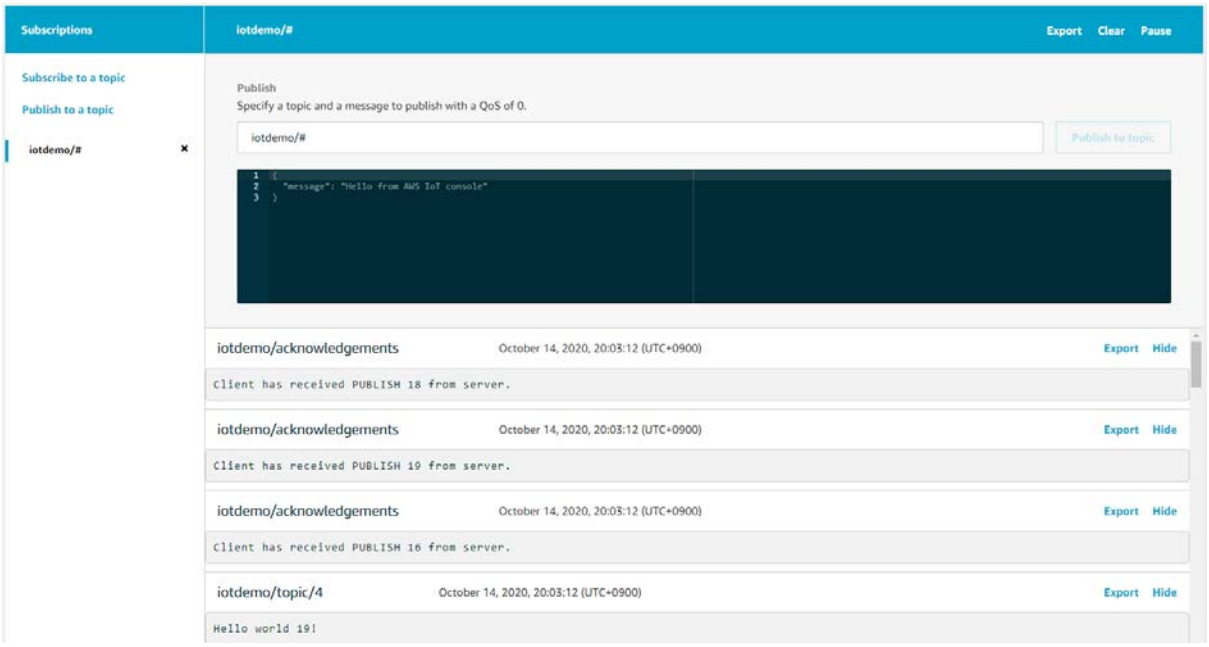
7. After the code is downloaded to the board, choose **Resume** to run the code up to the first line of the main function. Choose **Resume** again to run the rest of the code.

Monitoring MQTT messages on the cloud

You can use the MQTT client in the AWS IoT console to monitor the messages that your device sends to the AWS Cloud.

To subscribe to the MQTT topic with the AWS IoT MQTT client

1. Sign in to the AWS IoT console..
2. In the navigation pane, choose **Test** to open the MQTT client.
3. In **Subscription topic**, enter `iotdemo/#`, and then choose **Subscribe to topic**.
4. Successful demo run looks like following the picture



For the latest projects released by Renesas, see the `renesas` fork of the `amazon-freertos` repository on GitHub.

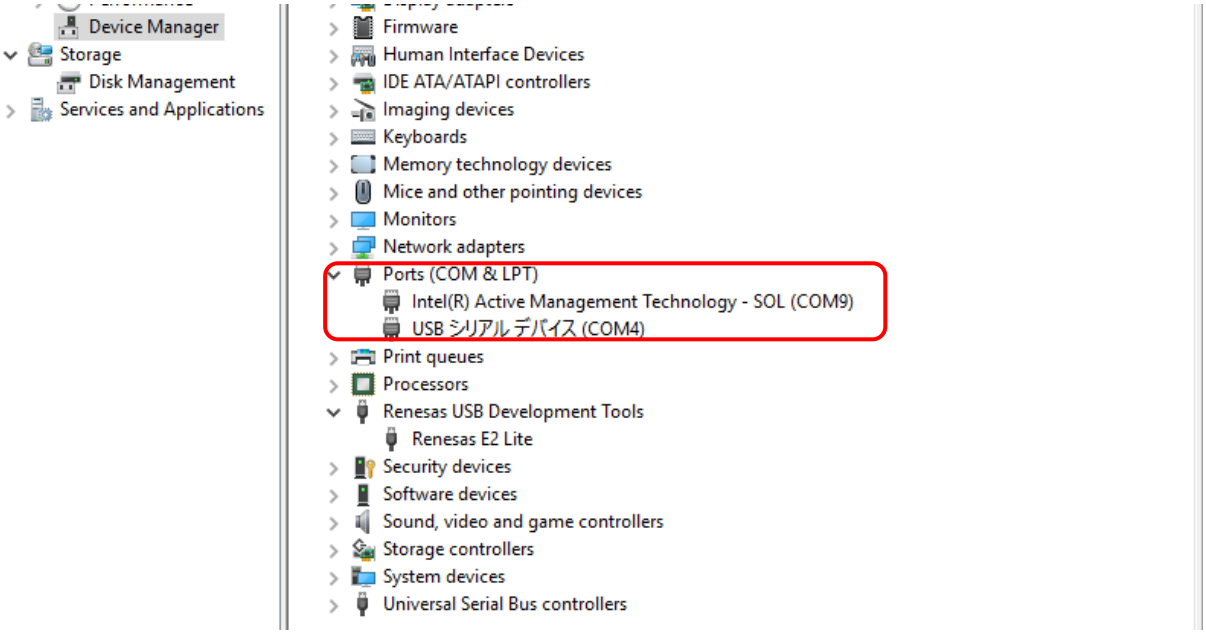
We are supporting FreeRTOS project creation feature on e2 studio, please download the latest version to create RX72N Envision Kit project with FreeRTOS by one-click.

Troubleshooting

Checking serial port output

Serial connection is established upon driver installation which is executed automatically when you connect your computer to the USB-to-serial port (CN8) on your REK for RX72N.

After driver installation is completed, a new COM port will appear in the Windows Device Manager window:



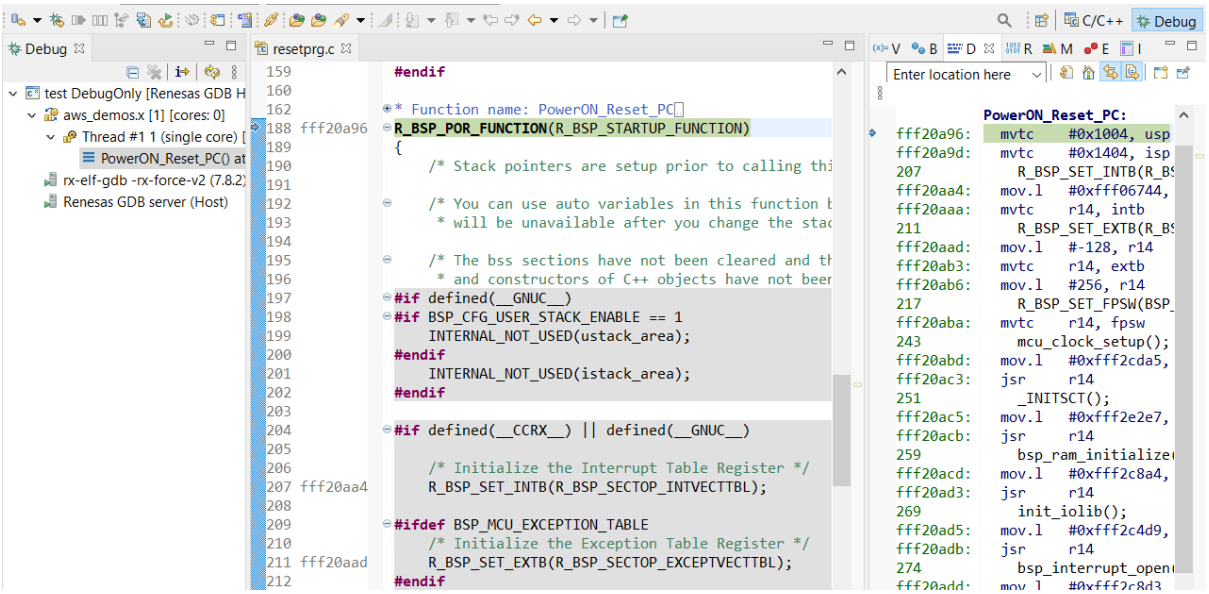
For debugging purposes, you can start a session to the port with any terminal utility tool (e.g: TeraTerm).

> **Note**
> If new COM port is not displayed after connecting computer to the USB-to-serial port, please reestablish the connection (disconnect and reconnect microUSB cable).

Debugging FreeRTOS projects in e^{2}studio

To use the debugging feature in e^{2}studio

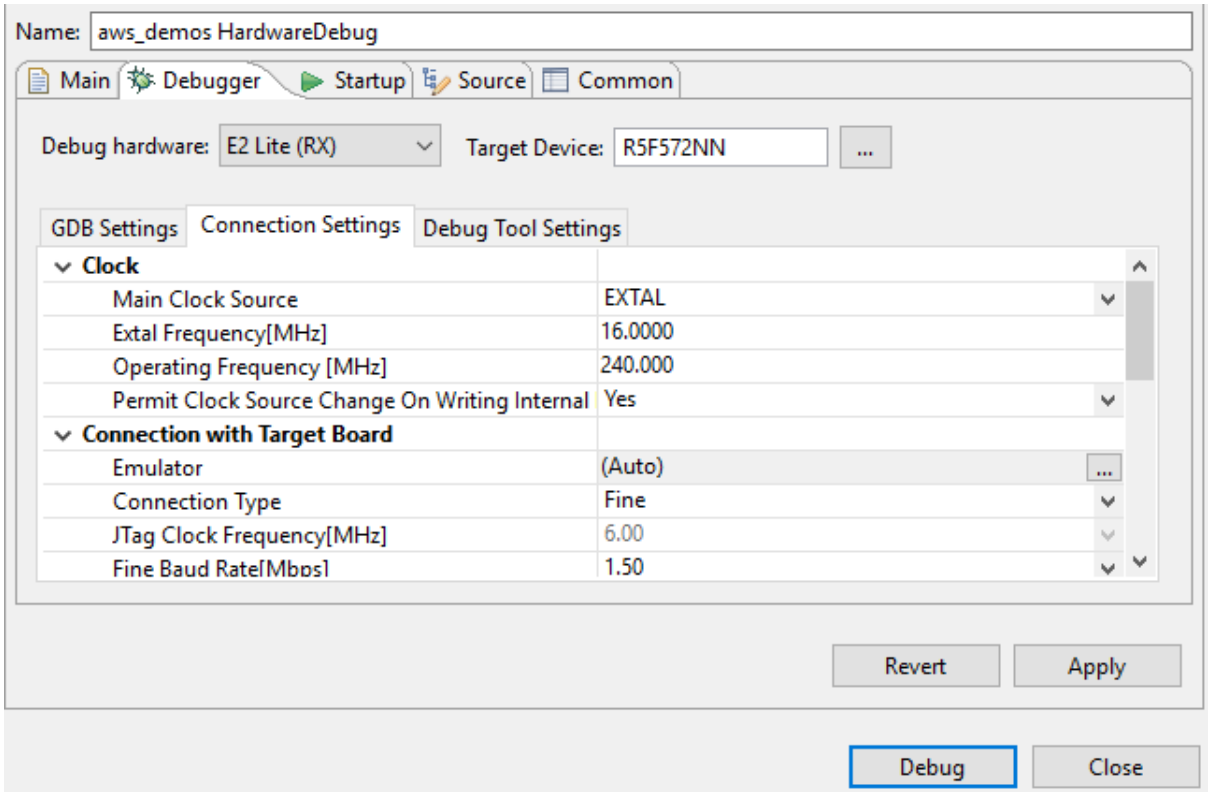
1. Confirm that you have connected your computer to the USB-to-serial port (CN8) on your REK for RX72N.
2. Launch e^{2}studio.
3. Follow the steps to build and run the FreeRTOS demo project in Build and run the FreeRTOS demo project.
4. When e^{2}studio ask to change to **Renesas Debug Perspective**. Choose **Yes**.
5. After the code is downloaded to the board, choose **Resume** to run the code up to the first line of the main function.
6. In the **Renesas Debug Perspective**, you can set breakpoints in the source code of your project. It should look something as follows:



To configure debug settings in e²studio

If you encounter problems while debugging a FreeRTOS project, please check that your debug settings are set correctly in e²studio.

1. Launch e²studio.
2. Follow the steps to build the FreeRTOS demo project in Build and run the FreeRTOS demo project.
3. From the top menu, choose **Run**, **Debug Configurations…**.
4. Expand **Renesas GDB Hardware Debugging**, and choose **aws_demos HardwareDebug**.
5. Choose the **Debugger** tab, and then choose the **Connection Settings** tab. Confirm that your connection settings are correct:
    - Debug hardware: E2 Lite (RX)
    - Target Device: R5F572NN
    - Main Clock Source: EXTAL
    - Extal Frequency:16.0000
    - Power Target From The Emulator (MAX 200mA): No



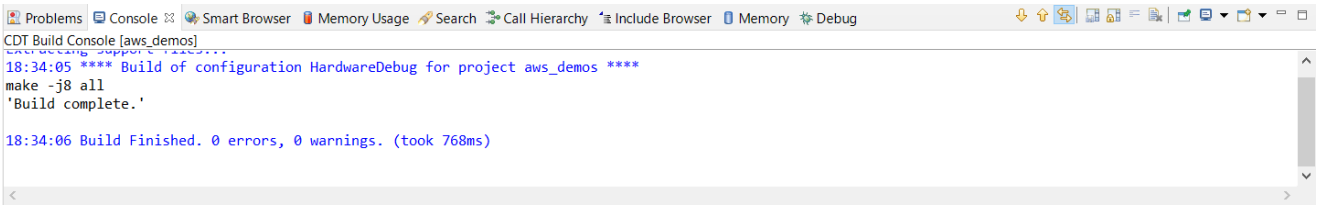
6. After confirming, choose **Debug** to download the code to your board and begin debugging.

Troubleshooting FreeRTOS projects in e²studio

If you do not see the expected messages in AWS IoT console, please refer to e²studio build console log. It will provide further information regarding the cause of the problem.

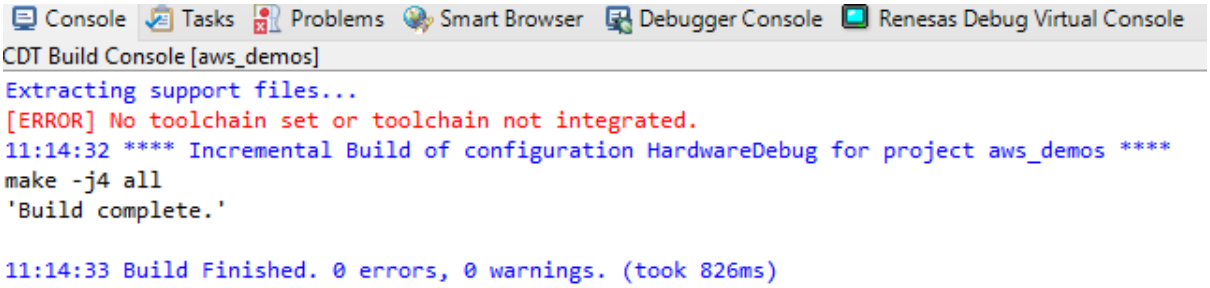
To enable build console in e²studio

1. The e²studio console should be enabled by default.
2. If it is not displayed after you launch e²studio, you may need to enable it manually.
3. From the top menu, choose **Window >> Show view >> Console**.
4. To show the relevant build console for troubleshooting, choose **CDT Build Console**. You can do this by clicking the  and choose **C/C++ Build Console**. It should look something as follows:

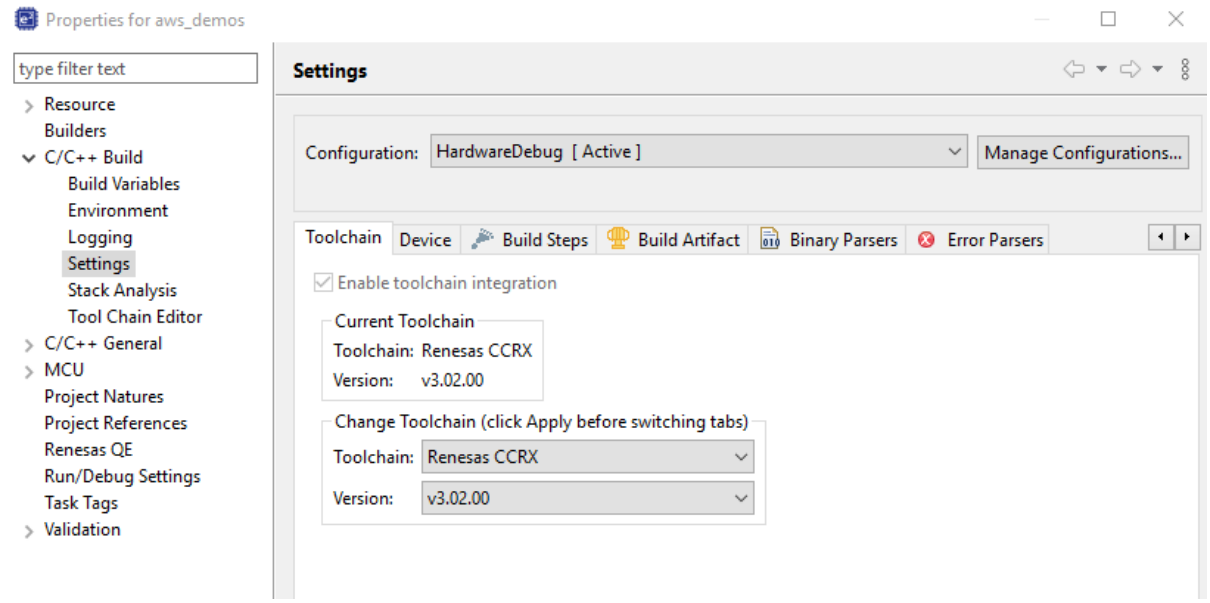


The following suggestions might help fix some common build errors you might encounter:

1. Make sure your aws_demos pathname does not exceed ~260 characters, this can cause build errors.
2. Make sure full pathname is used for pre-include file (-preinclude) implicitlyinclude.h, incomplete/partial pathname can cause build errors with `CC-RX v3.02.00`
3. If the build message is below



Make sure that the latest version of toolchain (**CC-RX v3.02.00**) is already installed on local PC. Then right click on aws_demos → Properties → C/C++ build → Settings → Toolchain tab and set parameters as following picture



If the steps did not solve your problem, please refer to e²studio build console log. It will provide further information regarding the cause of the problem.

For general troubleshooting information about Getting Started with FreeRTOS, see Troubleshooting getting started.