User's Manual

32

RX671 Group

Renesas Starter Kit+ for RX671
Tutorial Manual
For CS+

RENESAS 32-Bit MCU
RX Family / RX600 Series

Renesas Electronics
www.renesas.com

Rev. 1.00 May 2021

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

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/.

Trademarks

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

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.

Disclaimer

By using this Renesas Starter Kit+ (RSK+), the user accepts the following terms:

The RSK+ is not guaranteed to be error free, and the entire risk as to the results and performance of the RSK+ is assumed by the User. The RSK+ is provided by Renesas on an “as is” basis without warranty of any kind whether express or implied, including but not limited to the implied warranties of satisfactory quality, fitness for a particular purpose, title and non-infringement of intellectual property rights with regard to the RSK+. Renesas expressly disclaims all such warranties. Renesas or its affiliates shall in no event be liable for any loss of profit, loss of data, loss of contract, loss of business, damage to reputation or goodwill, any economic loss, any reprogramming or recall costs (whether the foregoing losses are direct or indirect) nor shall Renesas or its affiliates be liable for any other direct or indirect special, incidental or consequential damages arising out of or in relation to the use of this RSK+, even if Renesas or its affiliates have been advised of the possibility of such damages.

Precautions

The following precautions should be observed when operating any RSK+ product:

This Renesas Starter Kit+ is only intended for use in a laboratory environment under ambient temperature and humidity conditions. A safe separation distance should be used between this and any sensitive equipment. Its use outside the laboratory, classroom, study area or similar such area invalidates conformity with the protection requirements of the Electromagnetic Compatibility Directive and could lead to prosecution.

The product generates, uses, and can radiate radio frequency energy and may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment causes harmful interference to radio or television reception, which can be determined by turning the equipment off or on, you are encouraged to try to correct the interference by one or more of the following measures;

- ensure attached cables do not lie across the equipment
- reorient the receiving antenna
- increase the distance between the equipment and the receiver
- connect the equipment into an outlet on a circuit different from that which the receiver is connected
- power down the equipment when not in use
- consult the dealer or an experienced radio/TV technician for help NOTE: It is recommended that wherever possible shielded interface cables are used.

The product is potentially susceptible to certain EMC phenomena. To mitigate against them it is recommended that the following measures be undertaken;

- The user is advised that mobile phones should not be used within 10m of the product when in use.
- The user is advised to take ESD precautions when handling the equipment.

The Renesas Starter Kit+ does not represent an ideal reference design for an end product and does not fulfil the regulatory standards for an end product.

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of how to use the CS+ IDE to develop and debug software for the RSK+ platform. It is intended for users designing sample code on the RSK+ platform, using the many different incorporated peripheral devices.

The manual comprises of step-by-step instructions to load and debug a project in CS+, but does not intend to be a complete guide to software development on the RSK+ platform. Further details regarding operating the RX671 microcontroller may be found in 'RX671 Group User's Manual: Hardware' and within the provided sample code. The setup procedure for the RSK+ installer is described in the Quick Start Guide.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

In this manual, the display may differ slightly from screen shots. There is no problem in reading this manual.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the RX671 Group. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Electronics Web site.

Document Type	Description	Document Title	Document No.
User's Manual	Describes the technical details of the RSK+ hardware.	Renesas Starter Kit+ for RX671 User's Manual	R20UT4879EG
Tutorial Manual	Provides a guide to setting up RSK+ environment, running sample code and debugging programs.	Renesas Starter Kit+ for RX671 Tutorial Manual	R20UT4880EG
Quick Start Guide	Provides simple instructions to setup the RSK+ and run the first sample, on a single A4 sheet.	Renesas Starter Kit+ for RX671 Quick Start Guide	R20UT4881EG
Smart Configurator Tutorial Manual	Provides a guide to code generation and importing into the CS+ IDE.	Renesas Starter Kit+ for RX671 Smart Configurator Tutorial Manual	R20UT4882EG
Schematics	Full detail circuit schematics of the RSK+.	Renesas Starter Kit+ for RX671 Schematics	R20UT4878EG
Hardware Manual	Provides technical details of the RX671 microcontroller.	RX671 Group User's Manual: Hardware	R01UH0899EJ

2. List of Abbreviations and Acronyms

Abbreviation	Full Form
ADC	Analog-to-Digital Converter
API	Application Programming Interface
bps	bits per second
CMT	Compare Match Timer
COM	COMmunications port referring to PC serial port
CPU	Central Processing Unit
E1/E2 Lite	Renesas On-chip Debugging Emulator
GUI	Graphical User Interface
IDE	Integrated Development Environment
IRQ	Interrupt Request
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LSB	Least Significant Bit
LVD	Low Voltage Detect
MCU	Micro-controller Unit
MSB	Most Significant Bit
PC	Personal Computer
PLL	Phase-locked Loop
Pmod™	This is a Digilent Pmod™ Compatible connector. Pmod™ is registered to Digilent Inc. Digilent-Pmod_Interface_Specification
PSU	Power Supply Unit
RAM	Random Access Memory
ROM	Read Only Memory
RSK+	Renesas Starter Kit+
RTC	Real Time Clock
SCI	Serial Communications Interface
SPI	Serial Peripheral Interface
TFT	Thin Film Transistor
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WDT	Watchdog Timer

Table of Contents

1. Overview

1.1 Purpose

This RSK+ is an evaluation tool for Renesas microcontrollers. This manual describes how to get the RSK+ tutorial started, and basic debugging operations.

1.2 Features

This RSK+ provides an evaluation of the following features:

- Renesas microcontroller programming
- User code debugging
- User circuitry such as switches, LEDs and a potentiometer

  Through the provided set of sample applications.

The RSK+ board contains all the circuitry required for microcontroller operation.

2. Introduction

This manual is designed to answer, in tutorial form, the most common questions asked about using a Renesas Starter Kit+ (RSK+). The tutorials help explain the following:

- How do I compile, link, download and run a simple program on the RSK+?
- How do I build an embedded application?
- How do I use Renesas' tools?

The project generator will create a tutorial project with three selectable build configurations:

- 'DefaultBuild' is a project with debug support and optimisation level set to two.
- 'Debug' is a project built with the debugger support included. Optimisation level is set to zero.
- 'Release' is a project with optimised compile options (level two) and no outputs debugging information options selected, producing code suitable for release in a product.

Files referred to in this manual are installed using the project generator as you work through the tutorials. The tutorial examples in this manual assume that installation procedures described in the RSK+ Quick Start Guide have been completed. Please refer to the Quick Start Guide for details of preparing the configuration.

Due to the project generator, it is possible that line numbers for source code illustrated in this document do not match exactly with that in the actual source files. It is also possible that the source address of instructions illustrated in this manual differ from those in user code compiled from the same source. These differences are minor, and do not affect the functionality of the sample code nor the validity of this manual.

> These tutorials are designed to show you how to use the RSK+ and are not intended as a comprehensive introduction to the CS+ debugger, compiler toolchains or the E2 emulator Lite. Please refer to the relevant user manuals for more in-depth information.

2.1 Smart Configurator Plugin

The Smart Configurator plug in for the RX671 has been used to generate the sample code discussed in this document. Smart Configurator for CS+ is a plugin tool for generating template 'C' source code and project settings for the RX671. When using Smart Configurator, it supports the user with a visual way of configuring the target device, clocks, software components, hardware resources and interrupts for the project; thereby bypassing the need, in most cases, to refer to sections of the Hardware Manual.

Once the user has configured the project, the 'Smart Configurator' function is used to generate three code modules for each specific MCU feature selected. These code modules are name 'Config_xxx.h', 'Config_xxx.c', and 'Config_xxx_user.c', where 'xxx' is an acronym for the relevant MCU feature, for example 'CMT'. Within these code modules, the user is then free to add custom code to meet their specific requirement. However, these files require custom code to be added between the following comment delimiters:

```
/* Start user code for adding. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
```

Smart Configurator will locate these comment delimiters, and preserve any custom code inside the delimiters on subsequent code generation operations. This is useful if, after adding custom code, the user needs to re-visit Smart Configurator to change any MCU operating parameters.
Note: If code is added outside the above user code area, it will be lost if code generation is executed again with Smart Configurator.

In this RSK+ sample project, only some functions are used.
For other useful features, refer to the https://www.renesas.com/smart-configurator.

3. Tutorial Project Workspace

3.1 Introduction

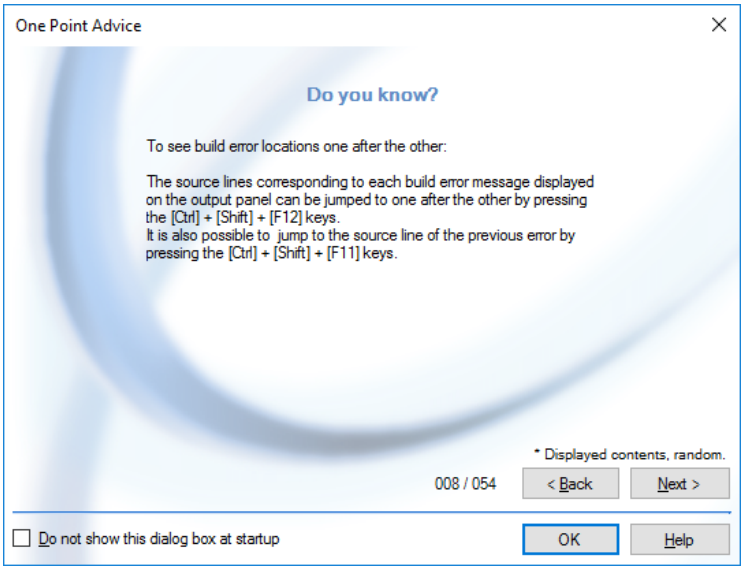
CS+ is an integrated development tool that allows the user to write, compile, program and debug a software project on the RX, RL78 and RH850 family of Renesas microcontrollers. CS+ will have been installed during the installation of the software support for the Renesas Starter Kit+ product. This manual will describe the stages required to create and debug the supplied tutorial code.

3.2 Starting CS+

To use the program, start CS+:
Windows™ 8.1: From Apps View, click 'CS+ for CC (RL78,RX, RH850)' icon

Windows™ 10: Start Menu > All Apps > Renesas Electronics CS+ > CS+ for CC (RL78,RX,RH850)

The first time CS+ is started, the One Point Advice dialog box will be shown:

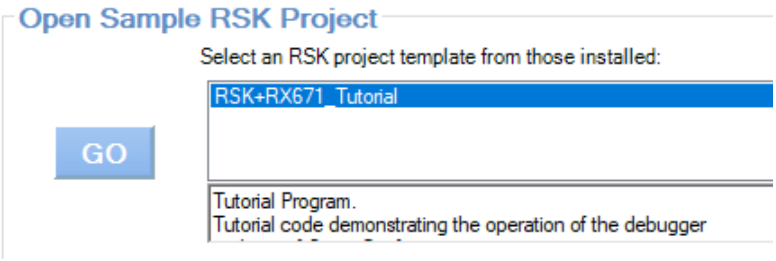


The One Point Advice dialog box provides some useful tips when using CS+. Press 'OK' to skip the advice and close the One Point Advice dialog.

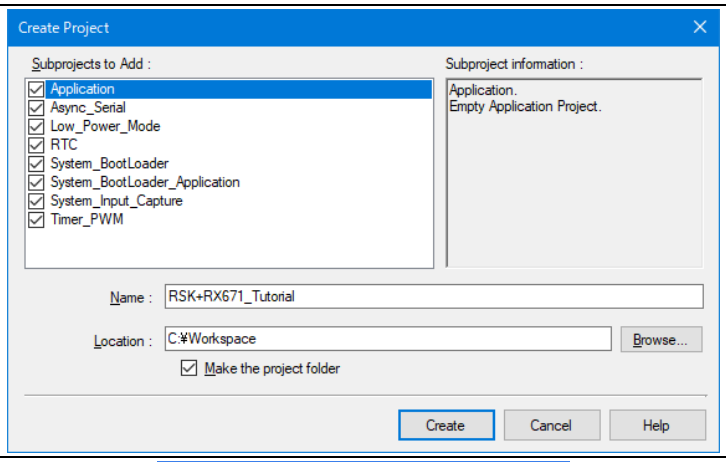
When start up for the first time, the "Confirm an agreement" dialog is displayed. After reading the contents, check the tick box and click ‘Yes (Y)’ or ‘No (N)’ button.

The user will then be presented with the Start panel.

Under the ‘Open Sample RSK Project’, open a new Tutorial project by selecting the RSK+RX671_Tutorial project template and click on ‘Go’ as shown below. This will save a copy of the RSK+RX671_Tutorial project.

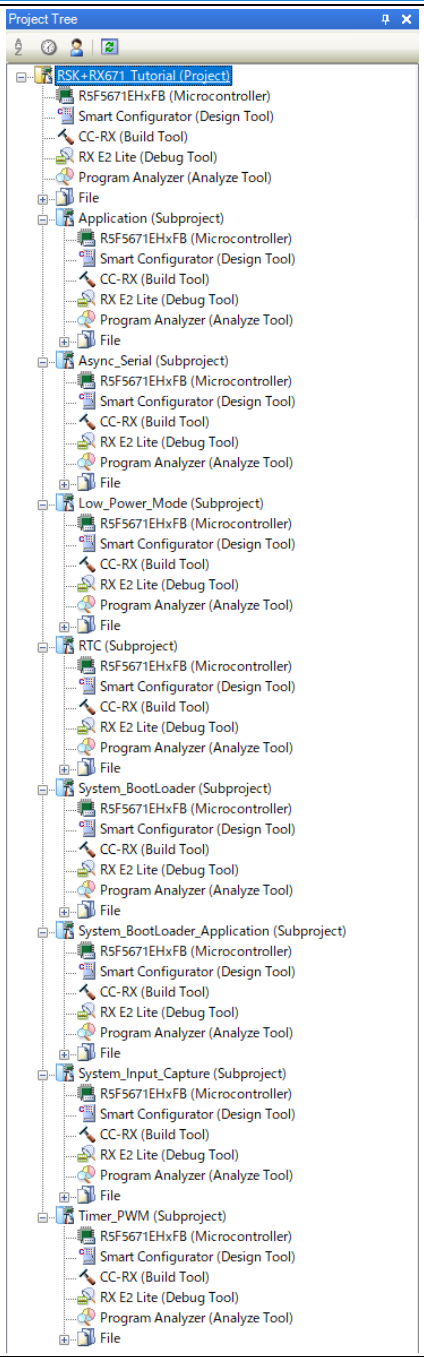


- CS+ will present a ‘Create Project’ dialog box.
- Select all sub-projects by clicking on each checkbox and observe the information displayed under the ‘Subproject information’ heading as you select each project.
- Specify a name and location for the new project and click on ‘Create’
- A dialog box will appear if the location specified does not exist; asking to create the folder specified. Click ‘OK’.

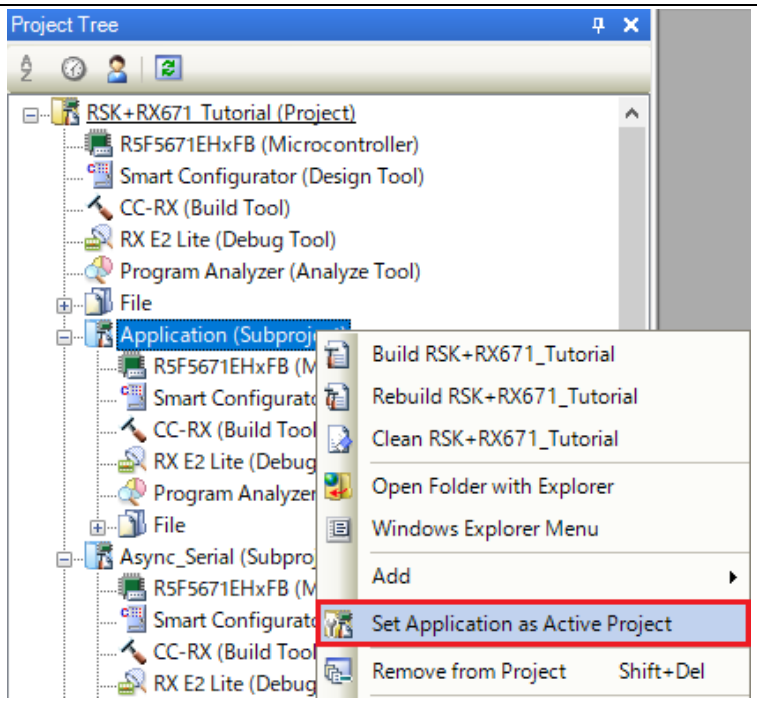


- CS+ will create and open the project showing the Project Tree as seen in the screenshot opposite.
- RSK+RX671_Tutorial (Project) is the master project and includes the tools to modify, build and debug the code.
- The File folder seen in the screenshot belongs to the master project, RSK+RX671_Tutorial.
- This folder contains and lists all project source and header files including text files arranged in separate folder structures.
- Folders containing the subprojects, indicated by “(Subproject)”, are listed below the File folder.
- Each subproject folder, when expanded, reveals an identical tools and folder structure to that of the master project, RSK+RX671_Tutorial.
- By default the RSK+RX671_Tutorial project is set as the active project, indicated by the line under the project name.

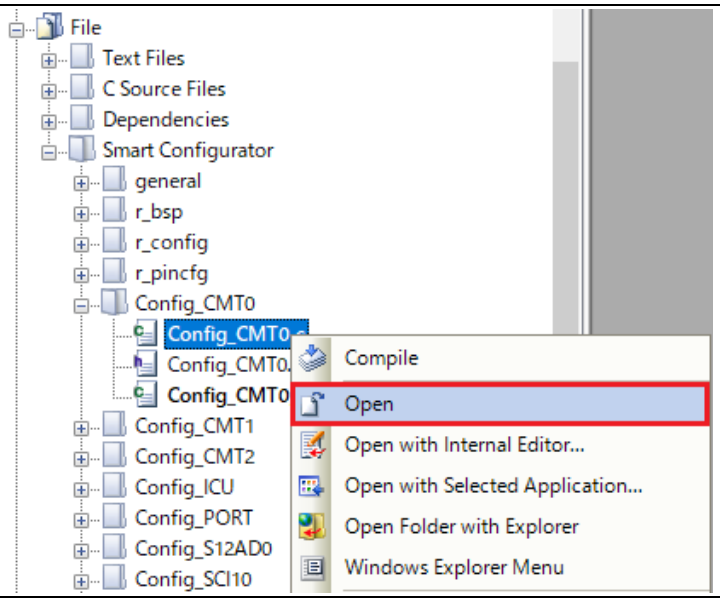
Note: ‘Smart Configurator (Design Tool)’ node is shown in the ‘Project Tree’ and indicates an optional plug-in has been enabled previously.



- To change the active project, right-click on the project/subproject name and select “Set x as Active Project” (x represents the project name).
- The opposite screen-shot is an example of changing the Application subproject to an active project.
- This is a demonstration of how to set the active project, please ensure that the "RSK+RX671_Tutorial " project is changed back to being the active project before continuing with this tutorial.



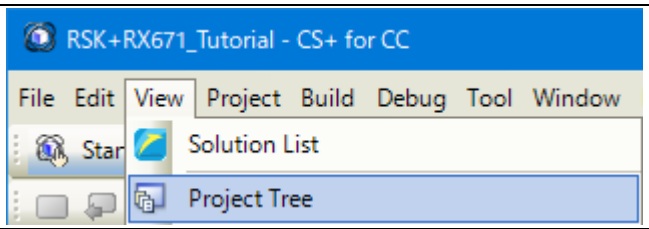
- The File folder contains four subfolders. This structure is common to all projects.
- Some of the source files were generated by Smart Configurator, which are grouped under the ‘Smart Configurator’ folder which itself is listed under the File folder in the Project Tree. These files are prefixed with ‘Config_’ to indicate that they were generated by a Smart Configurator. All other user-generated source files are contained in the ‘C Source Files’ folder.
- To open a file for viewing, right-click on the file and select ‘Open’. Alternatively, double-click on the file.



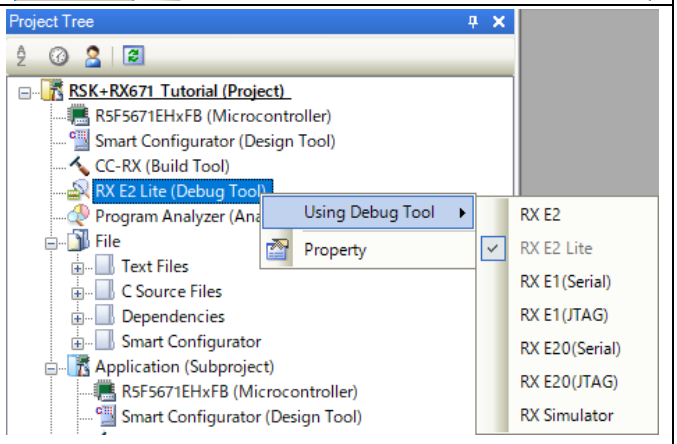
3.3 Configuring the Debug Tool (E2 Lite)

Note: The Tutorial sample project's settings are pre-configured. This section is intended to familiarise the user with the debug tool settings for when they create their own project.

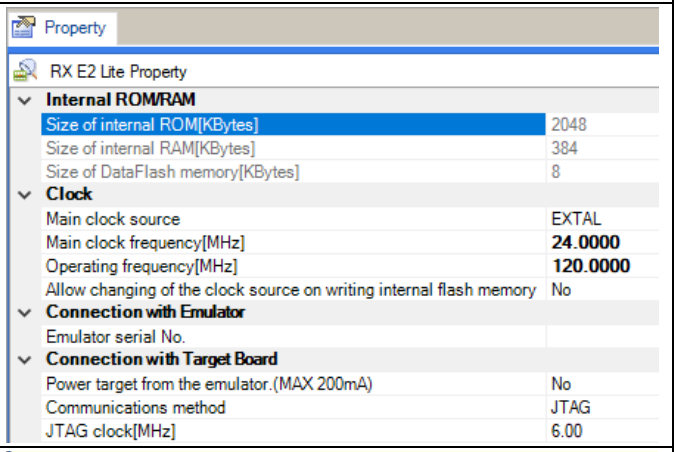
- The Project Tree will be displayed on the left-hand pane of CS+.
- This can also be invoked from the menu bar [View > Project Tree].



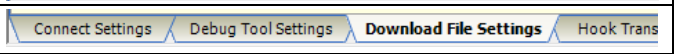
- The opposite screen-shot indicates that the selected Debug Tool is RX E2 Lite.



- Right click on RX E2 Lite (DebugTool).
- Click on Property.
- View the Connect Settings.
- Verify that the settings match the opposite screen-shot.



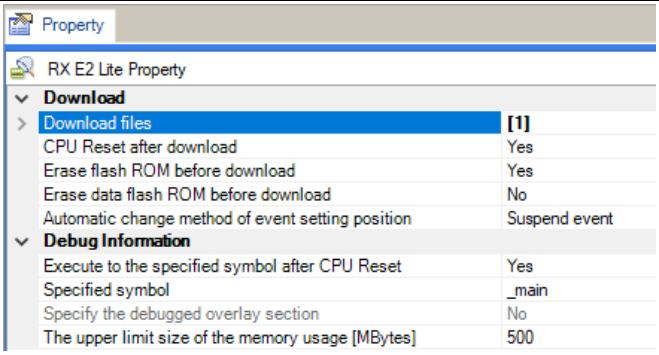
- Click on the 'Download File Settings' tab.



The project is configured to halt code execution on the first instruction of the main function after programming the micro controller. To specify another function as an entry point:

- Change the 'specified symbol' to another available function.
- Ensure to prefix the function name with an underscore ("_").

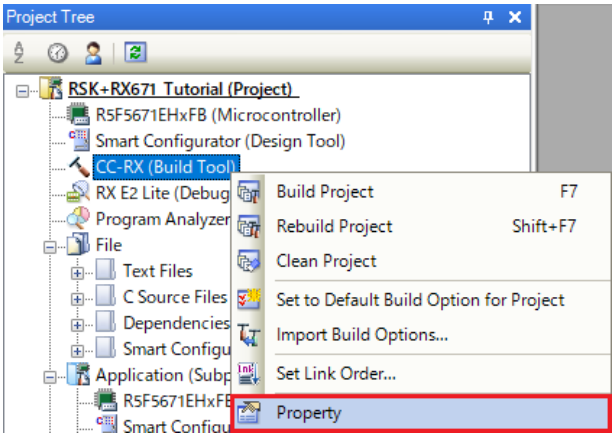
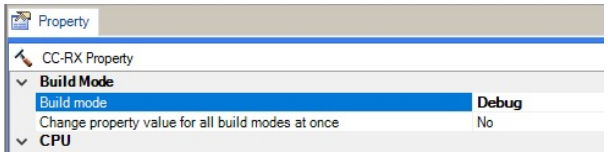
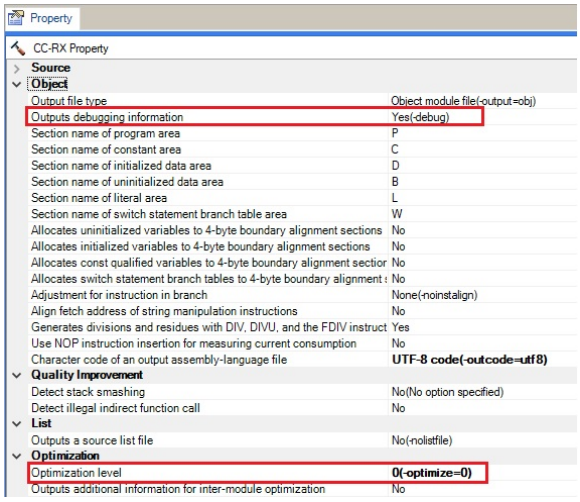
Note: Do not specify an interrupt handler as the entry point.



3.4 Build Configuration

The build configurations are selected from the build tool's Property panel. The options available are 'DefaultBuild', 'Debug' and 'Release'. 'DefaultBuild' and 'Debug' are configured for use with the debugger. 'Release' is configured for the final ROM programmable code.

A common difference between the two builds is the optimisation setting and the addition of debug information. With optimisation turned on, the debugger may seem to execute code in an unexpected order. To assist in debugging it is often helpful to turn optimisation off on the code being debugged.

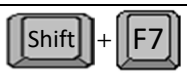
• Right-click on CC-RX (Build Tool) from the Project Tree. • Select 'Property'.	
• The 'Common Options' tab will open by default. • Verify that the Build Mode is set to 'Debug'.	
• Click on the 'Compile Options' tab to view compiler options.	
• Ensure the 'Outputs debug information' entry is set to 'Yes(-debug)'. • Ensure the 'Optimization' entry is set to '0(-optimize=0)'.	
• Review the options on each of the tabs to be aware of the options available. For the purposes of the tutorial, leave all options at default. • When complete, click the [x] in the right -hand corner of the properties window to close the properties panel.	

4. Building the Tutorial Program

The tutorial project build settings have been pre-configured in the toolchain options. To view the toolchain options double-click on CC-RX(Build Tool) from the Project Tree and select the available tabs. It is important when changing settings to be aware of the current configuration before modifying the settings.

4.1 Building the Code

There is a choice of three shortcuts available for building the project:

• Selecting the 'Build Project' toolbar button will build all projects listed in the project tree.	
• Pressing [F7]. This is equivalent to pressing the 'Build Project' toolbar button.	
• Selecting the 'Rebuild Project' toolbar button will rebuild all project files.	
• Pressing [Shift] + [F7]. This is equivalent to pressing the 'Rebuild Project' toolbar button.	
• Selecting the 'Build & Download' toolbar button will only build the active project and download the code to the target device after a successful build.	
• Pressing [F6]. This is equivalent to pressing the 'Build & Download' toolbar button.	

Build the project now by pressing [F7] or pressing one of the build icons as shown above. During the build each stage will be reported in the Output Window. The build will complete with an indication of any errors and warnings encountered during the build.

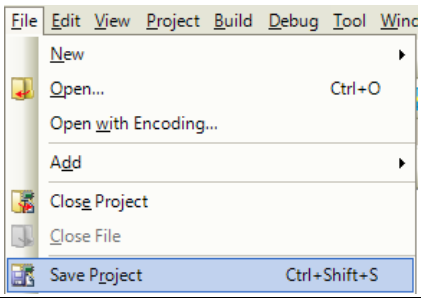
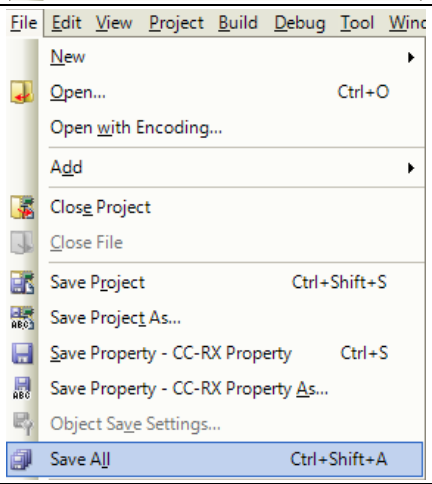
4.2 Connecting the Debugger

For this tutorial it is necessary to provide an external power supply to the board. Use the +5V center-positive PSU supplied with this RSK+ to power the board.

The Quick Start Guide provided with the Renesas Starter Kit board gives detailed instructions on how to connect the E2 Lite to the host computer. The following assumes that the steps in the Quick Start Guide have been followed and the E2 Lite drivers have been installed.

- Fit the PMOD LCD display to the board. Ensure all the pins of the connector are correctly inserted in the socket.
- Connect the E2 Lite Debugger to a free USB port on your computer.
- Connect the E2 Lite Debugger to the target hardware ensuring that it is plugged into the connector marked ‘E2 Lite’.
- Connect the +5V center-positive PSU to the PWR connector on the RSK+.

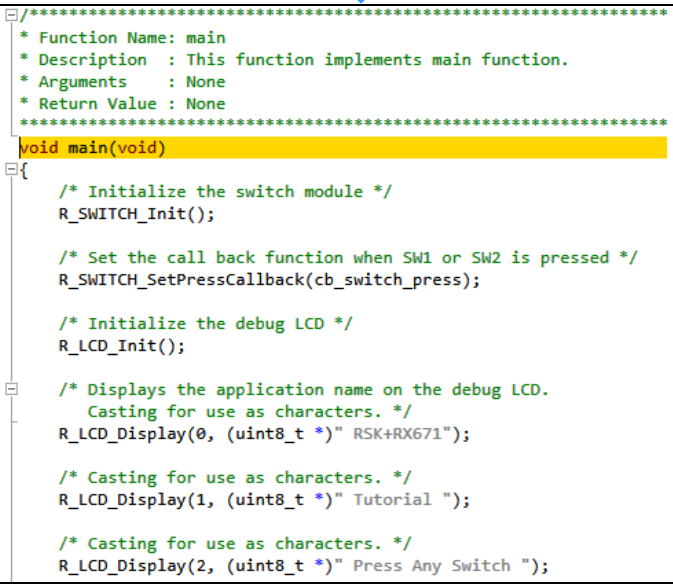
4.3 Saving Project Settings

If you have changed any project settings this is a good time to save the project. • Select ‘File’ | ‘Save Project’.	
If you make any changes to files in CS+ and want to preserve these change, you can save them by: • Select ‘File’ | ‘Save All’.	
You can also save files by clicking the ‘Save’ or ‘Save All’ buttons from the CS+ toolbar. In addition, keyboard shortcut keys can be saved with 'Ctrl + S' for 'Save' and 'Ctrl + Shift + A' for 'Save All'.	

5. Downloading and Running the Tutorial

5.1 Downloading the Program Code

Now that the code has been built in CS+ it needs to be downloaded to the RSK+.

• Click on the program download button. Alternatively, select Debug from the Menu bar and click on Download.	
• On completion of program download, the debugger and code are ready to be executed. The program counter indicator will point to first line of code inside the 'main' function; this is the program's entry point.	

```
/***********************************************************************
* Function Name: main
* Description  : This function implements main function.
* Arguments    : None
* Return Value : None
***********************************************************************/
void main(void)
{
    /* Initialize the switch module */
    R_SWITCH_Init();

    /* Set the call back function when SW1 or SW2 is pressed */
    R_SWITCH_SetPressCallback(cb_switch_press);

    /* Initialize the debug LCD */
    R_LCD_Init();

    /* Displays the application name on the debug LCD.
       Casting for use as characters. */
    R_LCD_Display(0, (uint8_t *)" RSK+RX671");

    /* Casting for use as characters. */
    R_LCD_Display(1, (uint8_t *)" Tutorial ");

    /* Casting for use as characters. */
    R_LCD_Display(2, (uint8_t *)" Press Any Switch ");
```

Before proceeding, it is necessary to connect to the RSK+ G1CUSB0 port to a PC using a USB Type A to mini B cable. The first time this port is connected to the PC an 'Installing Device Driver Software' pop-up will appear and the device driver will be automatically installed. Open Device Manager, the virtual COM port will appear under 'Port (COM & LPT)' as 'RSK+ USB Serial Port (COMx)', where x is a number. Open a terminal emulation program, such as HyperTerminal, with the settings:
Baud Rate: 19200, Data Length: 8, Parity Bit: None, Stop Bit: 1, Flow Control: None

5.2 Running the Tutorial

Once the program has been downloaded onto the RSK+ device, the program can be executed. Click the 'Go' button or press F5 to begin the program from the current program counter position. It is recommended that you run through the program once first, and then continue to the review section. Operating instructions for the program can be found in the file 'Description.txt', under the 'Text Files' folder in the CS+ Project Tree.	

6. Reviewing the Tutorial Program

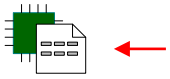
This section will explain the basic debugging techniques of the tutorial code.

6.1 Program Initialization

Before the main program can run, the microcontroller must be configured. Due to the debugger configuration used for the Tutorial project and the rest of the sample projects, the user will not be able to step through the hardware initialization code. Please refer to Section 3.3 to change the entry point after programming the microcontroller. Specify '_R_Systeminit' as the function name if viewing of hardware initialization is desired. The initialization code is executed every time the device is reset via the reset switch or from a power reboot. The user is advised not to use the 'step' feature of the debugger to exit the 'R_Systeminit' function.

Ensuring the Tutorial program has been downloaded onto the RX671; press the 'CPU Reset' button on the Debug Toolbar.	

- From the Menu bar select View > Disassemble > Disassemble1. Alternatively, use the Display Disassemble button to open and view the 'source and disassembly'.
- To make the Display Disassemble button available on the toolbar, right-click on the toolbar and select 'View Panels'.



Disassemble button

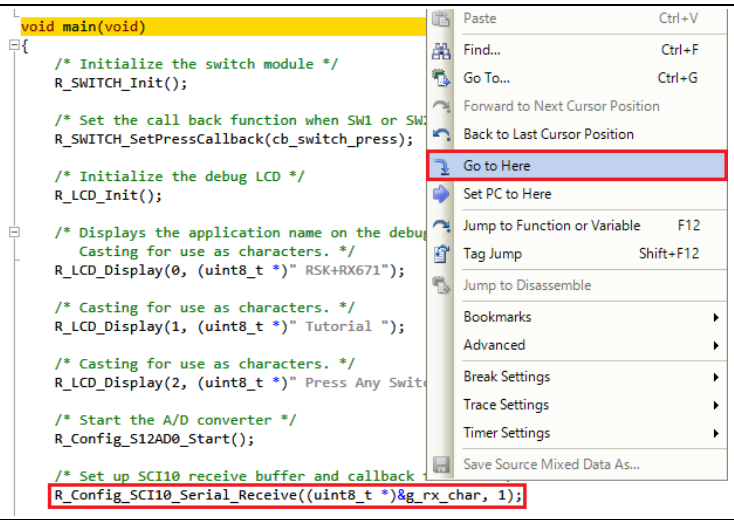
```
91:             void main(void)
                _main:
ffe0092c                6040            SUB             #4H,R0
94:                 R_SWITCH_Init();
 ffe0092e               05d50700        BSR.A           _R_SWITCH_Init
97:                 R_SWITCH_SetPressCallback(cb_switch_press);
 ffe00932               fb12280ae0ff    MOV.L           #-001FF5D8H,R1
 ffe00938               05d10700        BSR.A           _R_SWITCH_SetPressCallback
100:                R_LCD_Init();
 ffe0093c               05b00200        BSR.A           _R_LCD_Init
104:                R_LCD_Display(0, (uint8_t *)" RSK+RX671");
 ffe00940               fb228c08e0ff    MOV.L           #-001FF774H,R2
 ffe00946               6601            MOV.L           #0H,R1
 ffe00948               05b70200        BSR.A           _R_LCD_Display
107:                R_LCD_Display(1, (uint8_t *)" Tutorial ");
 ffe0094c               fb229708e0ff    MOV.L           #-001FF769H,R2
 ffe00952               6611            MOV.L           #1H,R1
 ffe00954               05ab0200        BSR.A           _R_LCD_Display
110:                R_LCD_Display(2, (uint8_t *)" Press Any Switch ");
 ffe00958               fb22a208e0ff    MOV.L           #-001FF75EH,R2
 ffe0095e               6621            MOV.L           #2H,R1
 ffe00960               059f0200        BSR.A           _R_LCD_Display
```

Revert back to the source by clicking on the file containing the function pointed to by the program counter indicator. Alternatively, right click in the Disassemble1 window and click "Jump to Source".

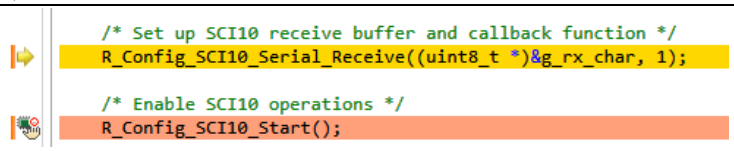
6.2 Main Functions

This section will look at the program code called from with the ‘main’ function, and how it works.

- Right click the ‘R_Config_SCI10_Serial_Receive’ function call and select ‘Go to Here’ to execute the program up to this line.
- The ‘R_LCD_Init’ function call enables and configures the LCD panel, and ‘R_LCD_Display’ will write “ RSK+RX671 ” onto the top line, “ Tutorial ” on the second line and “ Press Any Switch ” on the third.

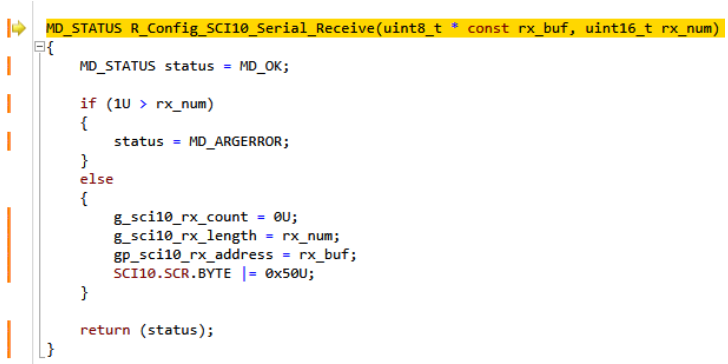


- Set a breakpoint on the ‘R_Config_SCI10_Start’ function call by clicking on the On-Chip Breakpoint column to the left of the number column.

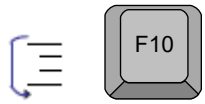


- Click the ‘Step In’ button to step into the ‘R_Config_SCI10_Serial_Receive’ function. Alternatively, press [F11].

- The program counter should now move into the ‘R_Config_SCI10_Serial_Receive’ function definition. This function is an API function provided by the Smart Configurator. It sets up the SCI10 interrupt handler code to receive a specified number of bytes into a receive buffer. Once the specified number of bytes has been received, the interrupt handler code calls a callback function as shown later on in this section.
- For full details on how to configure a project using Smart Configurator refer to the Smart Configurator Tutorial Manual.
- Press the button to resume program execution.



- The program counter should come to a halt at the 'R_Config_SCI10_Start' function.
- Step over the function by clicking the 'Step Over' button. Alternatively, press F10.



The 'R_Config_SCI10_Start' function enables the SCI interrupts. The program then proceeds to the main while(1U) loop. The code inside the loop waits for user input from either SCI reception or RSK+ switches, and then performs an A/D conversion.

```
    /* Set up SCI10 receive buffer and callback function */
    R_Config_SCI10_Serial_Receive((uint8_t *)&g_rx_char, 1);

    /* Enable SCI10 operations */
    R_Config_SCI10_Start();

    while (1U)
    {
```

- Locate the function call to 'lcd_display_adc' inside the while loop.
- Set a breakpoint by clicking in the main column of the "lcd_display_adc" function call line.
- In the Project Tree pane, locate the file 'Config_SCI10_user.c' and double-click to open the source file. Scroll down to the function 'r_Config_SCI10_callback_receiveend' .

```
    while (1U)
    {
        uint16_t adc_result;

        /* Wait for user requested A/D conversion flag to be set (SW1
        if (TRUE == g_adc_trigger)
        {
            /* Call the function to perform an A/D conversion */
            adc_result = get_adc();

            /* Display the result on the LCD */
            lcd_display_adc(adc_result);
```

- Set a breakpoint on the line of code inside the 'r_Config_SCI10_callback_receiveend' function as shown opposite.
- Continue to execute the program by pressing the  button.

```
static void r_Config_SCI10_callback_receiveend(void)
{
    /* Start user code for r_Config_SCI10_callback_receiveend. Do

    /* Check the contents of g_rx_char */
    if (('c' == g_rx_char) || ('C' == g_rx_char))
    {
        g_adc_trigger = TRUE;
    }
```

- In the terminal; emulation window, press the 'c' button on the keyboard.
- The program will halt at the breakpoint in the 'r_Config_SCI10_callback_receiveend' function as shown opposite. Remove the breakpoint by clicking on the breakpoint column.
- Continue to execute the program by pressing the  button.

```
static void r_Config_SCI10_callback_receiveend(void)
{
    /* Start user code for r_Config_SCI10_callback_receiveend. Do

    /* Check the contents of g_rx_char */
    if (('c' == g_rx_char) || ('C' == g_rx_char))
    {
        g_adc_trigger = TRUE;
    }

    /* Set up SCI10 receive buffer and callback function again */
    R_Config_SCI10_Serial_Receive((uint8_t *)&g_rx_char, 1);

    /* End user code. Do not edit comment generated here */
```

- The program will halt at the breakpoint in the main while loop.
- Remove the breakpoint by clicking on the breakpoint column.
- Continue to execute the program by pressing the  button.

```
    while (1U)
    {
        uint16_t adc_result;

        /* Wait for user requested A/D conversion flag to be set (SW1
        if (TRUE == g_adc_trigger)
        {
            /* Call the function to perform an A/D conversion */
            adc_result = get_adc();

            /* Display the result on the LCD */
            lcd_display_adc(adc_result);
```

The program proceeds to display the result of the A/D conversion on the LCD and in the terminal window. In addition, the running count of A/D conversions performed is displayed in binary form using LEDs 0-3 on the RSK+. Adjust the potentiometer and press any switch on the RSK+ and an additional A/D conversion will be performed.

• Press the 'Stop' button to halt program execution. • This is the extent of the tutorial code.	

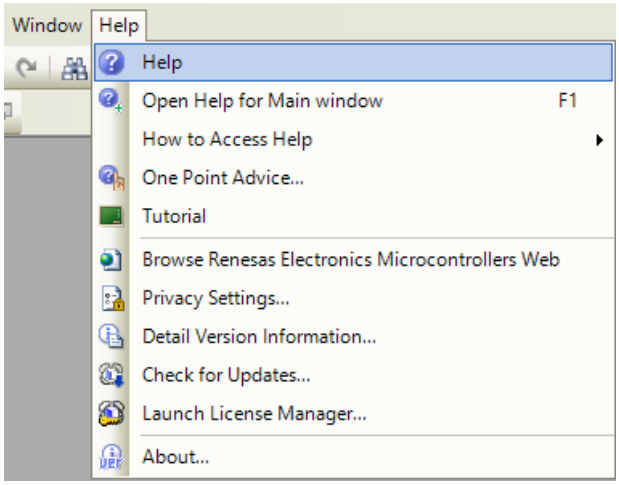
For further details regarding hardware configuration, please refer to 'RX Family User's Manual: Software' and ' RX671 Group User's Manual: Hardware'.

The E2 emulator Lite features advanced logic-based event point trigger system, and full instruction on its use is outside the scope of this tutorial. For further details, please refer to the E2 Emulator Lite User's Manual

7. Additional Information

Technical Support

For details on how to use CS+, refer to the help file by opening CS+, then selecting Help > Help Contents from the menu bar.



Parts of the sample code provided with the RSK+RX671 can be reproduced using the Smart Configurator plug in tool.
Source files and functions generated by Smart Configurator are prefixed with ‘r_’ and ‘R_’ or ‘Config_’, respectively.
For information about the RX671 Group microcontrollers refer to ‘RX671 Group User’s Manual: Hardware’.

For information about the RX assembly language, refer to ‘RX Family User’s Manual: Software’.

Technical Contact Details

Please refer to the contact details listed in section 8 of the “*Quick Start Guide*”

General information on Renesas microcontrollers can be found on the Renesas website at:
https://www.renesas.com/

Trademarks

All brand or product names used in this manual are trademarks or registered trademarks of their respective companies or organisations.

Copyright

REVISION HISTORY	RX671 Group Renesas Starter Kit+ for RX671 Tutorial Manual For CS+

Rev.	Date	Description	
		Page	Summary
1.00	May.10.21	—	First Edition issued

RX671 Group

Renesas Starter Kit+ for RX671 Tutorial Manual For CS+

Publication Date: Rev. 1.00 May.10.21

Published by: Renesas Electronics Corporation

RX671 Group

Renesas Electronics Corporation

R20UT4880EG0100