User's Manual

32

RX72M Group

Renesas Starter Kit+ for RX72M
Tutorial Manual
For e^2 studio

RENESAS 32-Bit MCU
RX Family / RX700 Series

Renesas Electronics
www.renesas.com

Rev. 1.00 Jul 2019

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

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 e^2 studio 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 e^2 studio, but does not intend to be a complete guide to software development on the RSK+ platform. Further details regarding operating the RX72M microcontroller may be found in the RX72M Group Hardware Manual and within the provided sample code. The setup procedure for the RSK+ Web 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 RX72M 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 RX72M User's Manual	R20UT4391EG
Tutorial Manual	Provides a guide to setting up RSK+ environment, running sample code and debugging programs.	Renesas Starter Kit+ for RX72M Tutorial Manual	R20UT4387EG
Quick Start Guide	Provides simple instructions to setup the RSK+ and run the first sample, on a single A4 sheet.	Renesas Starter Kit+ for RX72M Quick Start Guide	R20UT4388EG
Smart Configurator Tutorial	Provides a guide to code generation and importing into the e^2 studio IDE.	Renesas Starter Kit+ for RX72M Smart Configurator Tutorial Manual	R20UT4389EG
Schematics	Full detail circuit schematics of the RSK+.	Renesas Starter Kit+ for RX72M Schematics	R20UT4390EG
Hardware Manual	Provides technical details of the RX72M microcontroller.	RX72M Group Hardware Manual	R01UH0804EJ

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
- Sample application
- Sample peripheral device initialization code

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?

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 e^2 studio, the 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 plugin for the RX72M has been used to generate the sample code discussed in this document. Smart Configurator for e^2 studio is a plugin tool for generating template 'C' source code and project settings for the RX72M. 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

e² studio is an open source integrated development tool that allows the user to write, compile, program and debug a software product on many of the Renesas microcontrollers.

3.2 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 ‘E1/E2 Lite’.
- Connect the +5V center-positive PSU to the PWR connector on the RSK+.

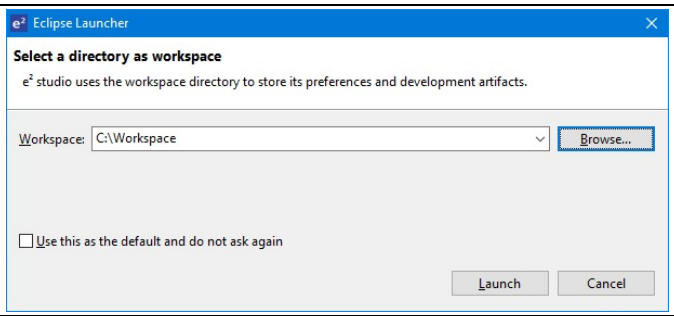
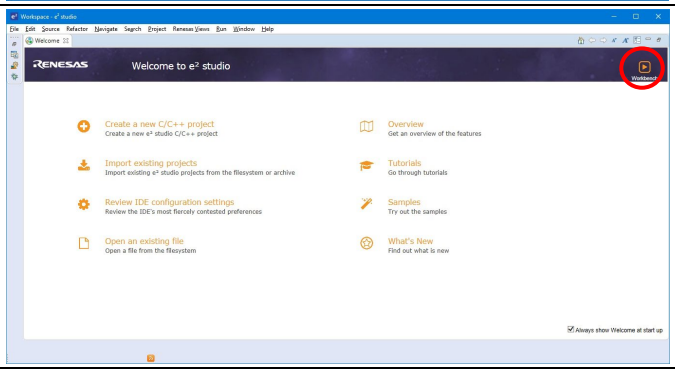
3.3 Starting e² studio and Importing Sample Code

To use the program, start e²studio:

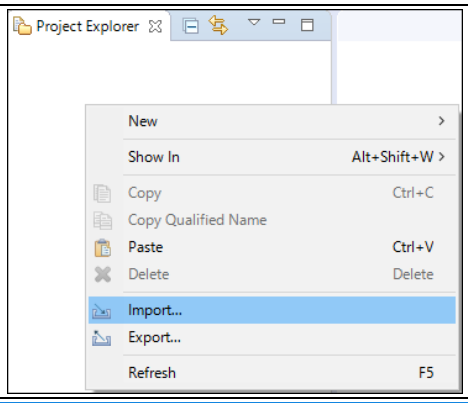
Windows™ 7: Start Menu > All Programs > Renesas Electronics e2studio > e2 studio

Windows™ 8.1 & 8: From Apps View, click ‘Renesas Electronics e2studio > e2 studio icon

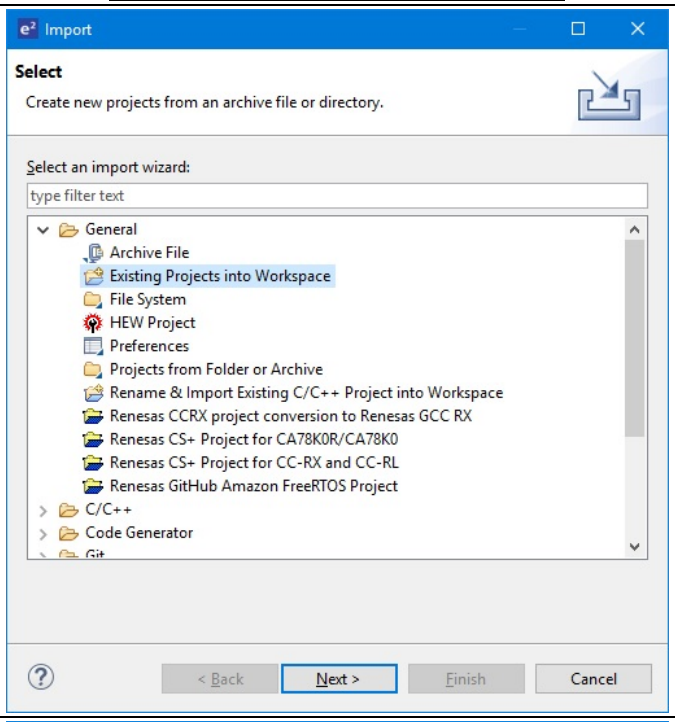
Windows™ 10: Start Menu > All Apps > Renesas Electronics e2studio > e2 studio

• Start e² studio by selecting it from the Windows™ Start Menu. The first dialog box to appear will be the Workspace Launcher. • Click ‘Browse’ and select a suitable location to store your workspace, using the ‘Make New Folder’ option as necessary. Click ‘Launch’.	
• The e² studio Welcome splash screen will appear. Click the ‘Workbench’ arrow button on the far right (circled in the screenshot opposite).	

- Once the environment has initialized, right click in the ‘Project Explorer’ window and select ‘Import…’



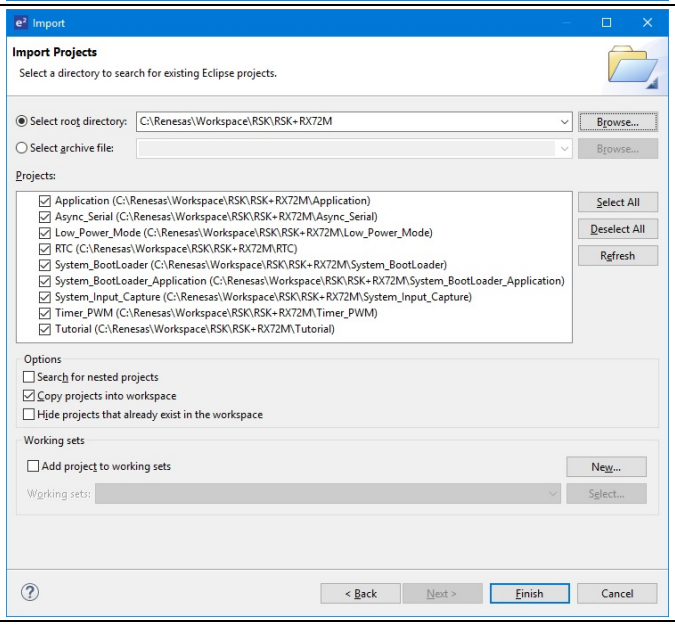
- The Import dialog box will be shown. Expand the ‘General’ folder icon, and select ‘Existing Projects into Workspace’, then click ‘Next’.



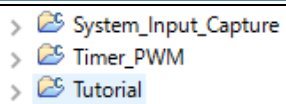
- The Import dialog box will allow you to specify a project to import. Click the ‘Browse’ button and locate the following directory:

  C:\Renesas\Workspace\RSK\RSK+RX72M

- Ensure that the ‘Copy projects into workspace’ option is ticked, and then click ‘Finish’.



- Click on Tutorial from the list of projects in the ‘Project Explorer’ on the left-hand side.



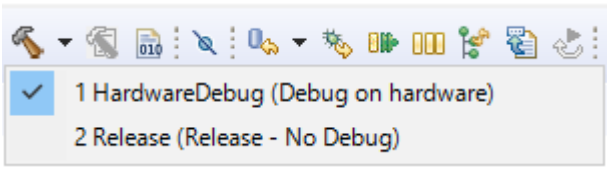
3.4 Build Configurations and Debug Sessions

3.4.1 Build Configuration

The e² studio workspace will be created with two build configurations: ‘HardwareDebug’ and ‘Release’.

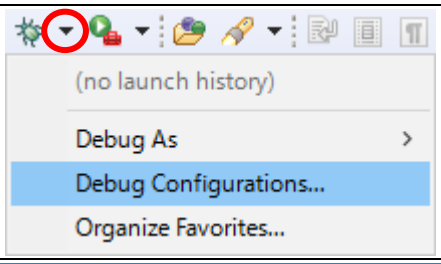
Release
This Build Mode has optimisation turned on, and provides little debug information. The C code execution may appear to be out of order, due to the way the compiler optimises the code. This build configuration is intended for final ROM-programmable code.

HardwareDebug
This Build Mode has all optimisation turned off, and provides full debug information. This is the best configuration to use whilst developing code as C code execution will be linear.

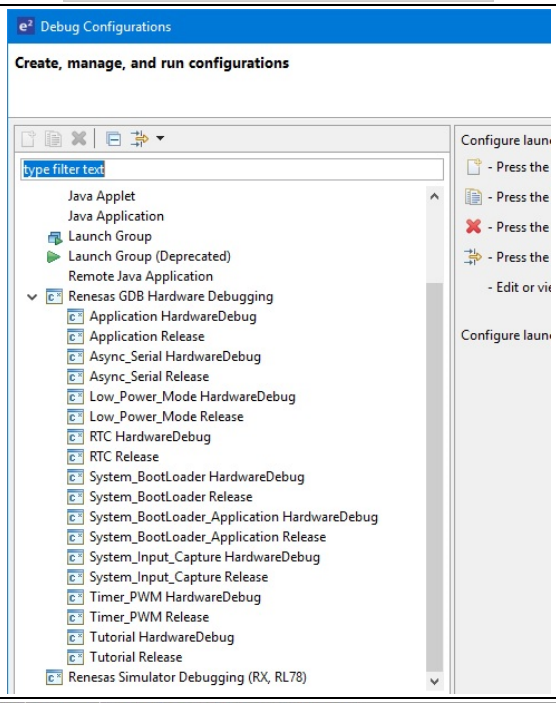
• Click the top level ‘Tutorial’ folder again, and then the arrow next to the build button (hammer icon) and select the ‘HardwareDebug’ option.  • e² studio will be built the code.	

3.4.2 Debug Configuration

- Click the arrow next to the debug button (bug icon), as highlighted by the red circle. Select ‘Debug Configurations’.



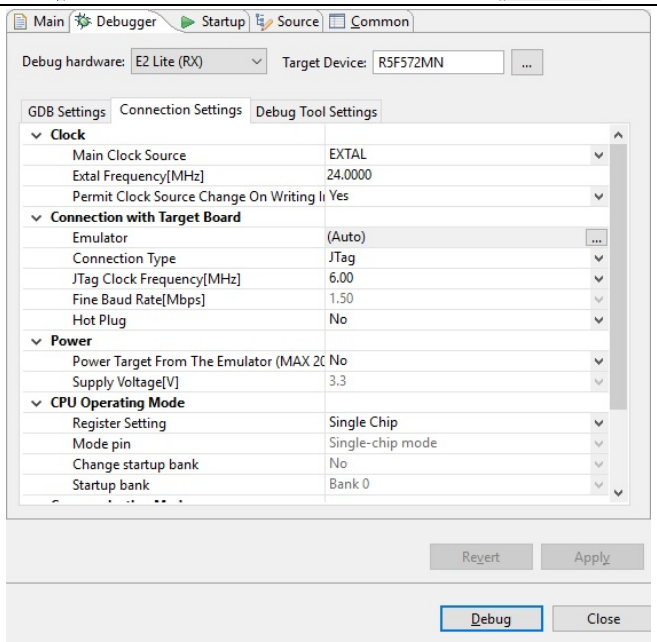
- The ‘Debug Configurations’ dialog box will appear. Click the small arrow next to the ‘Renesas GDB Hardware Debugging’ option.
- The debug configurations for each project will appear. Select the entry for the ‘Tutorial HardwareDebug’.

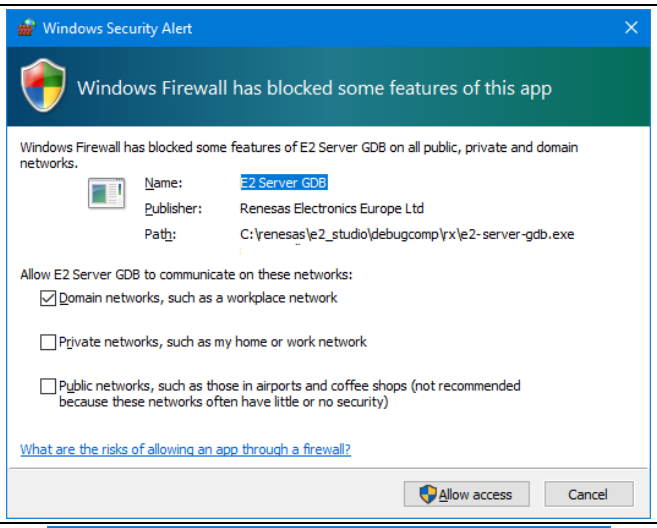
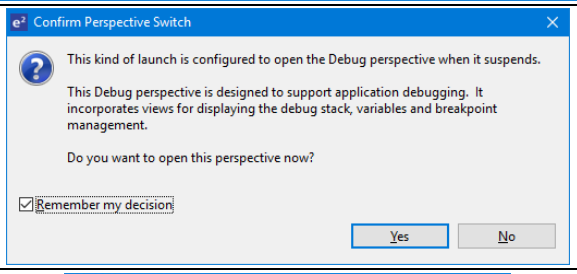
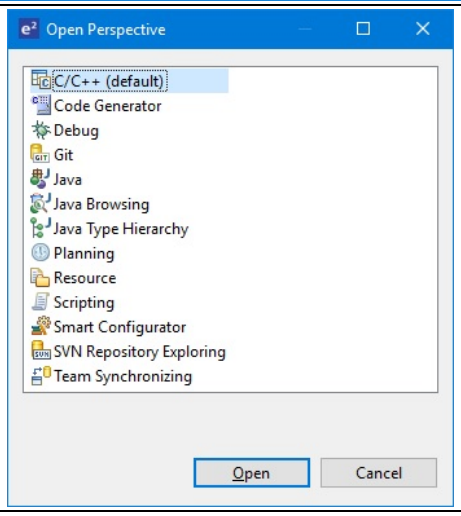


- The debug configurations control page will then show for the Tutorial project. Change the main tab to ‘Debugger’ and then select ‘Connection Settings’ on the secondary tab bar that appears.
- There is no need to change the debugger settings as they are preconfigured with the Tutorial project. Please check “Power Target From The Emulator (MAX 200mA)” is “No”.
- Refer to the RSK+RX72M User’s Manual for details of power supply configuration.

Note: e² studio will display a warning if you attempt to connect with an incorrect power supply setting.

- Click the ‘Debug’ button to continue. e² studio will be connected to the debugger and download the code to the target.



• A firewall warning may be displayed for 'e2-server-gdb.exe'. Check the 'Private networks, such as my home or work network' box and click 'Allow access'. • A user account control dialog may be displayed. Enter the administrator password and click 'Yes'.	
• After downloading the code a dialog box will appear asking if you would like to switch to the 'Debug perspective'. Click 'Remember my decision' to prevent this dialog box from appearing in future, then click 'Yes'. • e² studio will load the new perspective, which is optimised for debugging.	
• To change back to the default 'C/C++' perspective, from the menu bar select Window > Perspective >Open Perspective > Other or click the Open Perspective button.  • The 'Open Perspective' dialog box will appear. Click on the desired perspective to select it then 'OK'.	

3.5 Running the Tutorial

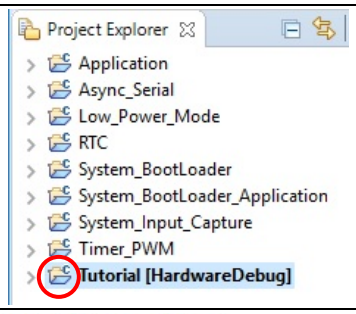
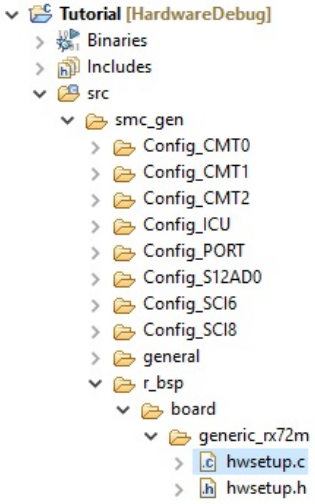
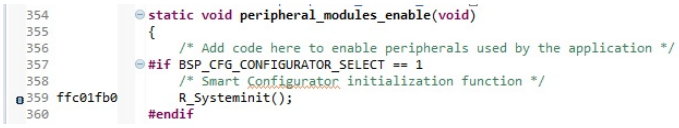
- Refer to the description.txt file in doc folder of Tutorial project for instructions on how to configure the RSK+ and run the sample code.
- Once the code has been downloaded, click 'Resume' button  to run the code to the main function. The main function is set as the program entry point by default. The program counter will stop on the first instruction in the main function.
- Click the 'Resume' button  in the 'Debug' perspective to run the rest of the code.
- It is recommended that you run the entire tutorial demo first, before continuing to debug it.

4. Reviewing the Tutorial Program

This section will look at each section of the tutorial code and basic debugging functionality in e² studio.

4.1 Program Initialization

Before the main program can run, the microcontroller must be configured. The following parts of the tutorial program are used exclusively for initializing the RSK+ device so that the main function can execute correctly. The initialization code is run every time the device is reset via the reset switch or from a power cycle.

• Build and download the code as shown in Section 3.4. • On the Project Explorer tab expand the 'Tutorial' folder by clicking on the arrow next to the folder icon, as highlighted by the red circle.	
• Click the arrow next to the 'src' folder to show the source files. • Expand the folder path 'smc_gen' -> 'r_bsp' -> 'board' -> 'generic_rx72m' as shown and double click on the file 'hwsetup.c'.	
• Breakpoints can be set by double clicking at the left-hand edge of the source window. On the line with instruction 'R_Systeminit();', double click next to the vertical line to set a breakpoint. Note: As an alternative, breakpoints may be set in the C/C++ perspective by selecting a line and using Run > Toggle Breakpoint.	

- Click the ‘Resume’ button in the Debug perspective (or press [F8]) to run the code up to this breakpoint.

Note: The program counter is indicated by the blue arrow next to the breakpoint.

```
354          static void peripheral_modules_enable(void)
355          {
356              /* Add code here to enable peripherals used by the application */
357          #if BSP_CFG_CONFIGURATOR_SELECT == 1
358              /* Smart Configurator initialization function */
359 ffc01fb0     R_Systeminit();
360          #endif
```

- Click the ‘Step Into’ button (or press [F5]), to step into the ‘R_Systeminit’ function.

- The ‘R_Systeminit’ function calls several initialization functions which configure the MCU for normal operation. This includes input/output ports, and system clocks.
- The user can step through all the initialization code by clicking the ‘Step Into’ icon and reading the code however for the purpose of this manual, it will be skipped.
- Click the ‘Resume’ button, to run the code up to the main function.

```
78 ffc01ea3  void R_Systeminit(void)
79           {
80               /* Enable writing to registers related to operating modes, LPC, CGC and software reset */
81 ffc01ea8     SYSTEM.PRCR.WORD = 0xA50BU;
82
83               /* Enable writing to MPC pin function control registers */
84 ffc01eb1     MPC.PWPR.BIT.B0WI = 0U;
85 ffc01eb8     MPC.PWPR.BIT.PFSWE = 1U;
86
87               /* Initialize clocks settings */
88 ffc01eba     R_CGC_Create();
89
90               /* Set peripheral settings */
91 ffc01ebe     R_Config_PORT_Create();
92 ffc01ec2     R_Config_CMT0_Create();
93 ffc01ec6     R_Config_CMT1_Create();
94 ffc01eca     R_Config_S12AD0_Create();
95 ffc01ece     R_Config_ICU_Create();
96 ffc01ed2     R_Config_SCI6_Create();
97 ffc01ed6     R_Config_SCI8_Create();
98 ffc01eda     R_Config_CMT2_Create();
99
100              /* Set interrupt settings */
101 ffc01ede     R_Interrupt_Create();
102
103              /* Register undefined interrupt */
104 ffc01ee2     R_BSP_InterruptWrite(BSP_INT_SRC_UNDEFINED_INTERRUPT,(bsp_int_cb_t)r_undefined_exception);
105
106              /* Register group AL0 interrupt TEI8 (SCI8) */
107 ffc01eee     R_BSP_InterruptWrite(BSP_INT_SRC_AL0_SCI8_TEI8,(bsp_int_cb_t)r_Config_SCI8_transmitend_interrupt);
```

For further details regarding hardware configuration, please refer to the RSK+RX72M User’s Manual and the RX72M Group Hardware Manual.

4.2 Main Functions

This section will look at the program code called from within the main function, and how it works. It is necessary to connect the G1CUSB0 port on the RSK+ to a PC USB port and 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
For information on installation of the RSK+ virtual COM port driver, refer to the file 'description.txt' in doc folder of the e^2 studio Tutorial project.

<table>
<tr>
<td>

- Right click the 'R_Config_SCI6_Serial_Receive' function call and select 'Run to Line' 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+RX72M Tutorial Press Any Switch" onto the LCD.

</td>
<td>

```
85                  * Function Name: main
90  ffc00918        void main(void)
91                  {
92                      /* Initialize the switch module */
93  ffc0091a            R_SWITCH_Init();
94
95                      /* Set the call back function when SW1 or SW2 is pressed */
96  ffc0091e            R_SWITCH_SetPressCallback(cb_switch_press);
97
98                      /* Initialize the debug LCD */
99  ffc00928            R_LCD_Init();
100
101                     /* Displays the application name on the debug LCD.
102                        Casting for use as characters. */
103 ffc0092c            R_LCD_Display(0, (uint8_t *)" RSK+RX72M");
104
105                     /* Casting for use as characters. */
106 ffc00938            R_LCD_Display(1, (uint8_t *)" Tutorial ");
107
108                     /* Casting for use as characters. */
109 ffc00944            R_LCD_Display(2, (uint8_t *)" Press Any Switch ");
110
111                     /* Start the A/D converter */
112 ffc00950            R_Config_S12AD0_Start();
113
114                     /* Set up SCI6 receive buffer and callback function */
115 ffc00956            R_Config_SCI6_Serial_Receive((uint8_t *)&g_rx_char, 1);
```

</td>
</tr>
<tr>
<td>

- Set a breakpoint on the 'R_Config_SCI6_Start' function call by double-clicking in the breakpoint column.
- Click the 'Step Into' button to step into the 'R_Config_SCI6_Serial_Receive' function.

</td>
<td>

```
114                 /* Set up SCI6 receive buffer and callback function */
115 ffc00956         R_Config_SCI6_Serial_Receive((uint8_t *)&g_rx_char, 1);
116
117                 /* Enable SCI6 operations */
118 ffc00960         R_Config_SCI6_Start();
```

</td>
</tr>
<tr>
<td>

- The program counter should now move into the 'R_Config_SCI6_Serial_Receive' function definition. This function is an API function provided by the Smart Configurator. It sets up the SCI 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.
- Click the 'Resume' button to resume program execution.

</td>
<td>

```
163 ffc01ab8    MD_STATUS R_Config_SCI6_Serial_Receive(uint8_t * const rx_buf, uint16_t rx_num)
164             {
165 ffc01abe         MD_STATUS status = MD_OK;
166
167 ffc01ac1         if (1U > rx_num)
168                  {
169 ffc01ac6             status = MD_ARGERROR;
170                  }
171                  else
172                  {
173 ffc01acb             g_sci6_rx_count = 0U;
174 ffc01ad4             g_sci6_rx_length = rx_num;
175 ffc01add             gp_sci6_rx_address = rx_buf;
176 ffc01aeb             SCI6.SCR.BIT.RIE = 1U;
177 ffc01af2             SCI6.SCR.BIT.RE = 1U;
178                  }
179
180 ffc01af4         return (status);
181 ffc01afa     }
```

</td>
</tr>
</table>

- The program counter should come to a halt at the ‘R_Config_SCI6_Start’ function.
- Step over the function by clicking the ‘Step Over’ button. Alternatively, press [F6].

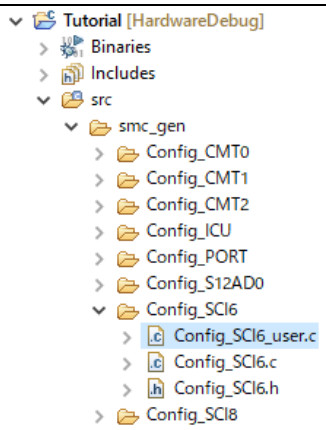
The ‘R_Config_SCI6_Start’ function enables the UART interrupts. The program then proceeds to the main while(1U) loop. The code inside the loop waits for user input from either the SCI or RSK+ switches, and then performs an A/D conversion.

```
114                 /* Set up SCI6 receive buffer and callback function */
115 ffc00956        R_Config_SCI6_Serial_Receive((uint8_t *)&g_rx_char, 1);
116
117                 /* Enable SCI6 operations */
118 ffc00960        R_Config_SCI6_Start();
```

- Locate the function call to ‘lcd_display_adc’ inside the while loop.
- Set a breakpoint on the ‘lcd_display_adc’ function call by double-clicking in the break point column.

```
120             while (1U)
121             {
122                 uint16_t adc_result;
123
124                 /* Wait for user requested A/D conversion flag t
125 ffc00a05        if (TRUE == g_adc_trigger)
126                 {
127                     /* Call the function to perform an A/D conve
128 ffc00967          adc_result = get_adc();
129
130                     /* Display the result on the LCD */
131 ffc0096c          lcd_display_adc(adc_result);
```

- In the Project Explorer pane, locate the file ‘Config_SCI6_user.c’ and double-click to open the source file. Scroll down to the function ‘r_Config_SCI6_callback_receiveend’.



- Set a breakpoint on the line of code inside the ‘r_Config_SCI6_callback_receiveend’ function as shown opposite.
- Continue to execute the program by clicking the ‘Resume’ button.

```
193 ffc01c54  static void r_Config_SCI6_callback_receiveend(void)
194           {
195               /* Start user code for r_Config_SCI6_callback_receiveend. Do not
196
197               /* Check the contents of g_rx_char */
198 ffc01c56      if (('c' == g_rx_char) || ('C' == g_rx_char))
199               {
200 ffc01c7b          g_adc_trigger = TRUE;
201               }
202
203               /* Set up SCI6 receive buffer and callback function again */
204 ffc01c86      R_Config_SCI6_Serial_Receive((uint8_t *)&g_rx_char, 1);
205
206               /* End user code. Do not edit comment generated here */
207           }
```

- In the terminal emulation window, press the ‘c’ button on the keyboard.
- The program will halt at the breakpoint in the ‘r_Config_SCI6_callback_receiveend’ function as shown opposite. Remove the breakpoint by double-clicking on the breakpoint column.
- Continue to execute the program by clicking the ‘Resume’ button.

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

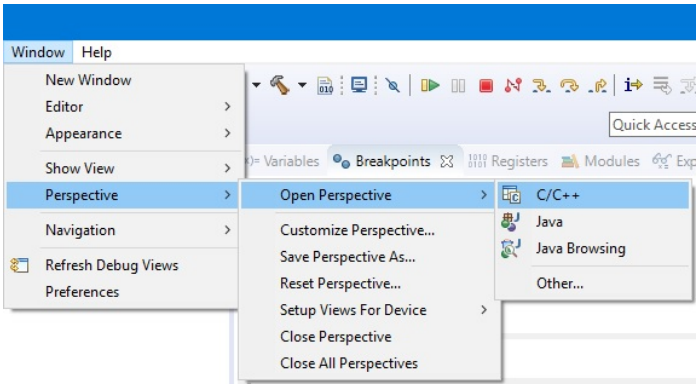
```
197                 /* Check the contents of g_rx_char */
198 ffc01c56        if (('c' == g_rx_char) || ('C' == g_rx_char))
199                 {
200 ffc01c7b            g_adc_trigger = TRUE;
201                 }
```

```
130                 /* Display the result on the LCD */
131 ffc0096c        lcd_display_adc(adc_result);
132
133                 /* Increment the adc count and display
134 ffc00971        if (16 == (++gs_adc_count))
135                 {
136 ffc00983            gs_adc_count = 0;
137                 }
138 ffc0098c        led_display_count(gs_adc_count);
```

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 SW1, SW2 or SW3 on the RSK+ and an additional A/D conversion will be performed.

- Press the ‘Suspend’ button to halt program execution.

- To change back to the default ‘C/C++’ perspective, from the menu bar select Window > Perspective >Open Perspective > ‘C/C++’.



- Alternatively, click on the ‘C/C++’ button in the top right corner of the screen, as shown opposite.

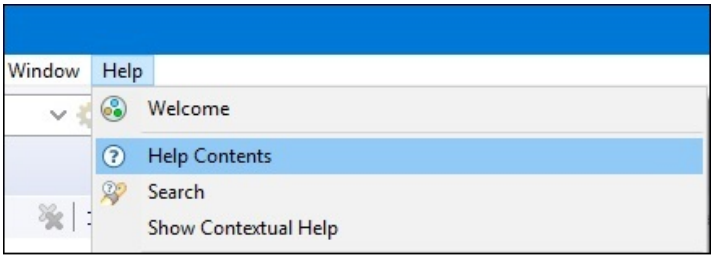
- This is the extent of the tutorial code.

For further details regarding hardware configuration, please refer to the RX72M User's Manual and the RX72M 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.

5. Additional Information

Technical Support

For details on how to use e^2 studio, refer to the help file by opening e^2 studio, then selecting Help > Help Contents from the menu bar.	

Parts of the sample code provided with the RSK+RX72M can be reproduced using the Smart Configurator tool. Smart Configurator is included as a plug in with e^2 studio.
Source files and functions generated by Smart Configurator are prefixed with ‘r_’ and ‘R_’ or ‘Config_’, respectively.

For information about the RX72M Group microcontrollers refer to the RX72M Group Hardware Manual.

For information about the RX assembly language, refer to the RX Family Software Manual.

Technical Contact Details

Please refer to the contact details listed in section 9 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	RX72M Group Renesas Starter Kit+ for RX72M Tutorial Manual For e^2 studio

Rev.	Date	Description	
		Page	Summary
1.00	Jul.31.19	—	First Edition issued

RX72M Group
Renesas Starter Kit+ for RX72M Tutorial Manual For e^2 studio

Publication Date: Rev. 1.00 Jul.31.19

Published by: Renesas Electronics Corporation

RX72M Group

Renesas Electronics Corporation

R20UT4387EG0100