

# RX220 Group

## Renesas Starter Kit Tutorial Manual

RENESAS MCU  
RX Family / RX200 Series

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corporation without notice. Please review the latest information published by Renesas Electronics Corporation through various means, including the Renesas Electronics Corporation website (<http://www.renesas.com>).

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.  
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.  

Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be 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, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

## 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 High-performance Embedded Workshop (HEW) 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 HEW, but does not intend to be a complete guide to software development on the RSK platform. Further details regarding operating the RX220 microcontroller may be found in the Hardware Manual and within the provided sample code.

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.

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 RX220 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.	RSKRX220 User's Manual	R20UT2230EG
Tutorial Manual	Provides a guide to setting up RSK environment, running sample code and debugging programs.	RSKRX220 Tutorial Manual	R20UT2231EG
Quick Start Guide	Provides simple instructions to setup the RSK and run the first sample, on a single A4 sheet.	RSKRX220 Quick Start Guide	R20UT2232EG
Schematics	Full detail circuit schematics of the RSK.	RSKRX220 Schematics	R20UT2229EG
Hardware Manual	Provides technical details of the RX220 microcontroller.	RX220 Group Hardware Manual	R01UH0292EJ

## 2. List of Abbreviations and Acronyms

Abbreviation	Full Form
ADC	Analog-to-Digital Converter
API	Application Programming Interface
CMT	Compare Match Timer
CPU	Central Processing Unit
DVD	Digital Versatile Disc
E1	Renesas On-chip Debugging Emulator
GUI	Graphical User Interface
GPT	General PWM Timer
IRQ	Interrupt Request
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MTU	Multi-Function Timer Pulse Unit
PC	Personal Computer
RAM	Random Access Memory
ROM	Read Only Memory
RSK	Renesas Starter Kit
RTC	Realtime Clock
SAU	Serial Array Unit
SCI	Serial Communications Interface
SFR	Special Function Registers
TAU	Timer Array Unit
TPU	Timer Pulse Unit
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WDT	Watchdog timer

# Table of Contents

1. Overview.....	7
1.1 Purpose.....	7
1.2 Features.....	7
2. Introduction.....	8
2.1 Note Regarding Source Code.....	8
3. Tutorial Project Workspace.....	9
4. Project Workspace.....	10
4.1 Introduction.....	10
4.2 Starting HEW.....	10
4.3 Build Configurations and Debug Sessions.....	11
4.3.1 Build Configuration.....	11
4.3.2 Debug Session.....	11
5. Building the Tutorial Program.....	12
5.1 Building Code.....	12
5.2 Connecting the Debugger.....	13
5.3 Connecting to the Target with the E1 Debugger.....	13
6. Downloading and Running the Tutorial.....	17
6.1 Downloading the Program Code.....	17
6.2 Running the Tutorial.....	17
7. Reviewing the Tutorial Program.....	18
7.1 Program Initialisation.....	18
7.2 Main Functions.....	20
8. Additional Information.....	24

## 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 initialisation 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?

The project generator will create a tutorial project with two selectable build configurations.

- 'Debug' is a project built with the debugger support included.
- 'Release' is a project built with optimised compile options, 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.

These tutorials are designed to show you how to use the RSK and are not intended as a comprehensive introduction to the High-performance Embedded Workshop (HEW) debugger, compiler toolchains or the E1 emulator. Please refer to the relevant user manuals for more in-depth information.

### 2.1 Note Regarding Source Code

During the project generator, it is possible that the line numbers for source code illustrated in this document does not match exactly with that in the actual source files. It is also possible that the source address of instructions illustrated in this manual differs from a user's code compiled from the same source. These differences are minor, and do not affect the functionality of the sample code or the validity of this accompanying manual.



### 3. Tutorial Project Workspace

The workspace includes all of the files for two build configurations, 'Debug' and 'Release'. The tutorial code is common to both build configurations; and is designed to show how code can be written, debugged and then downloaded without the debug monitor in a 'Release' situation.

The build configuration menu in High-performance Embedded Workshop allows the project to be configured such that certain files may be excluded from each of the build configurations. This allows the inclusion of the debug monitor within the Debug build, and its exclusion in the Release build. Contents of common C files are controlled with defines set up in the build configuration options and #ifdef statements within the source files. Maintaining only one set of project files means that projects are more controllable.

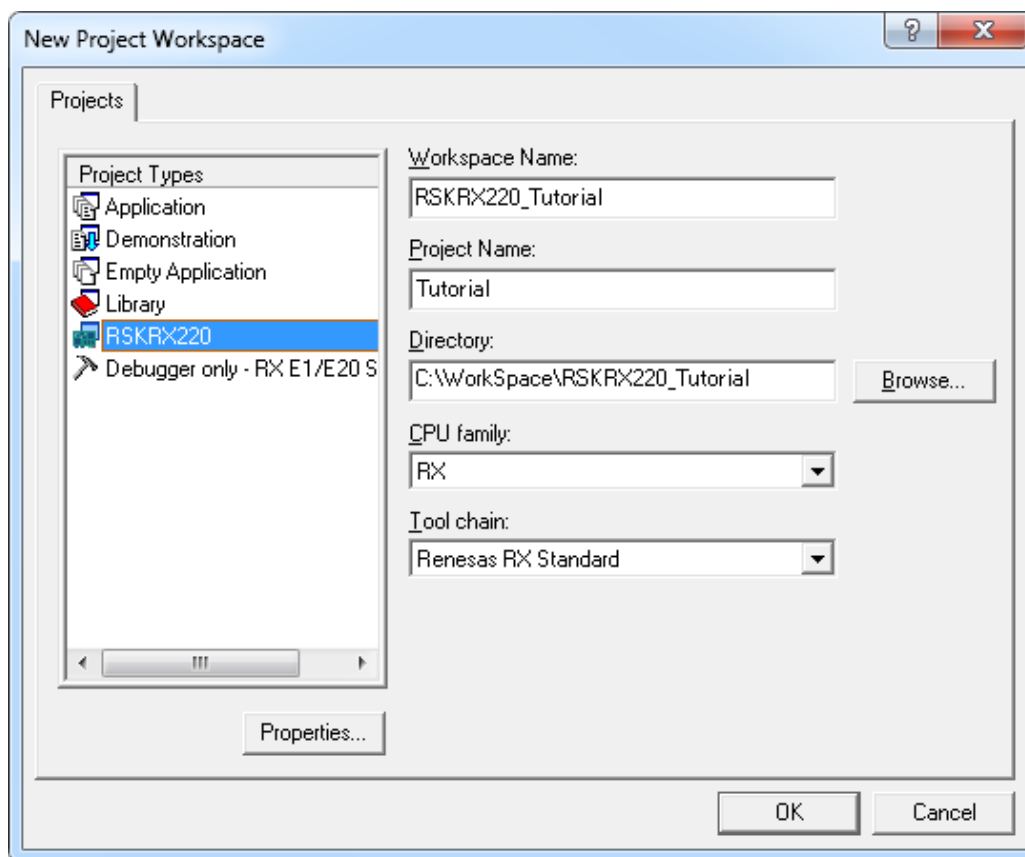
## 4. Project Workspace

### 4.1 Introduction

High-performance Embedded Workshop is an integrated development tool that allows the user to write, compile, program and debug a software project on any of the Renesas microcontrollers. High-performance Embedded Workshop 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.

### 4.2 Starting HEW

To look at the program, start High-performance Embedded Workshop from the Windows Start Menu. Open a new tutorial workspace from the [File > New Workspace...] menu or select 'Create a new project workspace' when presented with the 'Welcome!' dialog box.



The example above shows the 'New Project Workspace' dialog box with the RSKRX220 selected.

- Select the RX CPU family and 'Renesas RX Standard' toolchain.
- Select the 'RSKRX220' project type from the left-hand projects list.
- Enter a name for the workspace – all your files will be stored under a directory with this name.
- The project name field will be pre-filled to match the workspace name above, but this name may be changed manually.
- Note: High-performance Embedded Workshop allows you to add multiple projects to a workspace. You may add the sample code projects later so you may wish to choose a suitable name for the tutorial project now.
- Click [OK] to start the Renesas Starter Kit project generator wizard.

The next dialog box presents the three types of example project available:

- Tutorial: this is the one of interest at this time – the code is explained later in this manual.
- Sample Code: This provides examples for using various peripherals. If you select this and click <Next> it will open a new dialog box, allowing the selection of many code examples for the peripheral modules of the device.
- Application: where the debugger is configured but there is no program code. This project is suitable for the user to add code without having to configure the debugger.

The project generator wizard will display a confirmation dialog box. Press [OK] to create the project and insert the necessary files. A tree showing all the files in this project will appear in High-performance Embedded Workshop.

To view the file 'main.c', double click on the file in the Workspace window. A new window will open showing the code.

### 4.3 Build Configurations and Debug Sessions

The workspace that has been created contains two build configurations and two debug sessions. The build configuration allows the same project to be built but with different compiler options. The options available to the user are described fully in the High-performance Embedded Workshop Manual.

#### 4.3.1 Build Configuration

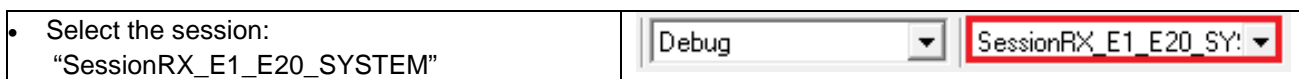
The build configurations are selected from the left hand drop down list on the toolbar. The options available are Debug and Release. The Debug build is configured for use with the debugger. The Release build is configured for final ROM programmable code.

A common difference between the two builds may be the optimisation settings. 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.



#### 4.3.2 Debug Session

The debug sessions are selected from the right hand drop down list on the toolbar. The options vary between Renesas Starter Kit types however one will always be 'DefaultSession' and the other will include the type of debug interface, in this case 'SessionRX\_E1\_E20\_SYSTEM'. The purpose of the debug sessions is to allow the use of different debugger tools or different debugger settings on the same project.



## 5. Building the Tutorial Program

The tutorial project build settings have been pre-configured in the toolchain options. To view the toolchain options select the 'Build' menu item and the relevant toolchain. This should be the first option on the drop down menu. The dialog box that is displayed will be specific to the toolchain selected.

The Configuration pane on the left hand side will exist on all the toolchain options. It is important when changing any setting to be aware of the current configuration that is being modified. If you wish to modify multiple or all build configurations this is possible by selecting 'All Configurations' or 'Multiple Configurations...' from the 'Configuration' drop down list.

<ul style="list-style-type: none"> <li>Review the options on each of the tabs and 'Category' drop down lists to be aware of the options available. For the purposes of the tutorial, leave all options at default.</li> <li>When complete close the dialog box by clicking [OK]</li> </ul>	
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

### 5.1 Building Code

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

<ul style="list-style-type: none"> <li>Selecting the 'Build All' toolbar button. This will build everything in the project that has not been excluded from the build. The standard library is built only once.</li> </ul>	
<ul style="list-style-type: none"> <li>Selecting the 'Build' toolbar button. This will build all files that have changed since the last build. The standard library will not be built unless an option has been changed.</li> </ul>	
<ul style="list-style-type: none"> <li>Pressing [F7]. This is equivalent to pressing the 'Build' button described above.</li> </ul>	

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.

### 5.2 Connecting the Debugger

For this tutorial it is not necessary to provide an external power supply to the board. The power will be obtained from the USB port. Please be aware that if you have too many devices connected to your USB port it may be shut down by Windows. If this happens remove some devices and try again. Alternatively provide an external power source taking care to ensure the correct polarity and voltage.

Other sample code supplied with this RSK will require a variable power supply; in which case an external 0-5V variable power supply should be used. Refer to the RSKRX220 User’s Manual for further details.

The Quick Start Guide provided with the Renesas Starter Kit board gives detailed instructions on how to connect the E1 to the host computer. The following assumes that the steps in the Quick Start Guide have been followed and the E1 drivers have been installed.


- Fit the LCD module to LCD connector on the board, via the header marked ‘LCD’. Ensure all the pins of the connector are correctly inserted in the socket.
- Connect the E1 Debugger to a free USB port on your computer.
- Connect the E1 Debugger to the target hardware ensuring that it is plugged into the connector marked ‘E1’.
- If supplying external power to the board, it can be turned on now.

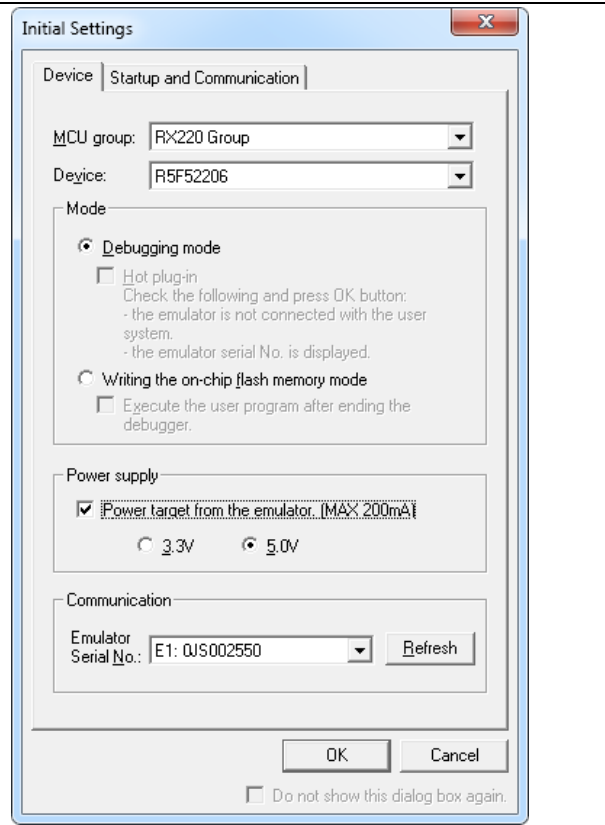
### 5.3 Connecting to the Target with the E1 Debugger

This section will take you through the process of connecting to the device, programming the Flash and executing the code.

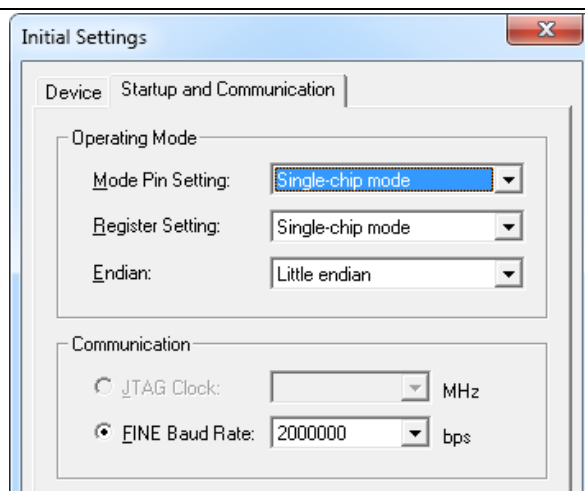
Please note that the “Emulator Mode” wizard shown here will only appear the FIRST time you connect to the target within a project. On subsequent connections the “Emulator Setting” dialog box will appear please choose the same options to connect.

- Select the ‘SessionRX\_E1\_E20\_SYSTEM’ debug session.
- Click the [Connect] button on the debug toolbar.
 

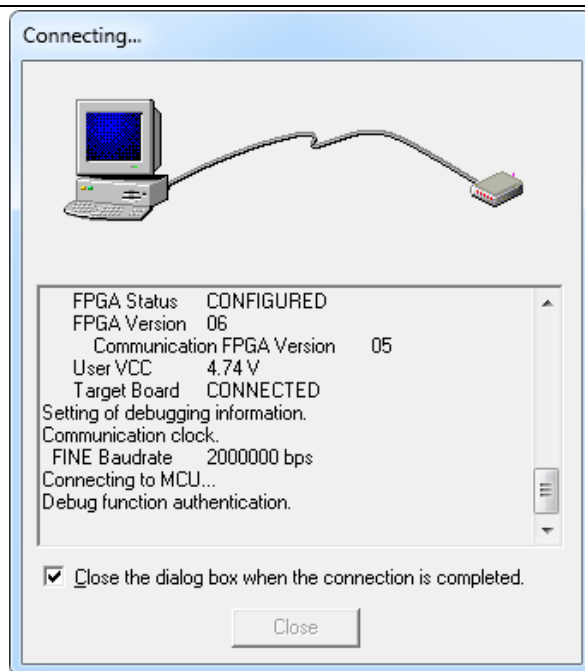

- The ‘Initial Settings’ configuration dialog box will appear. Ensure the follow configurations are set:
  - MCU group: RX220 Group
  - Device: R5F52206
  - Mode: Debugging mode
- If the E1 is to provide power to the CPU board, select ‘Power Target from Emulator’ and choose the “5.0V” option. Otherwise connect a suitable power supply (refer to the RSKRX220 User’s Manual for details).



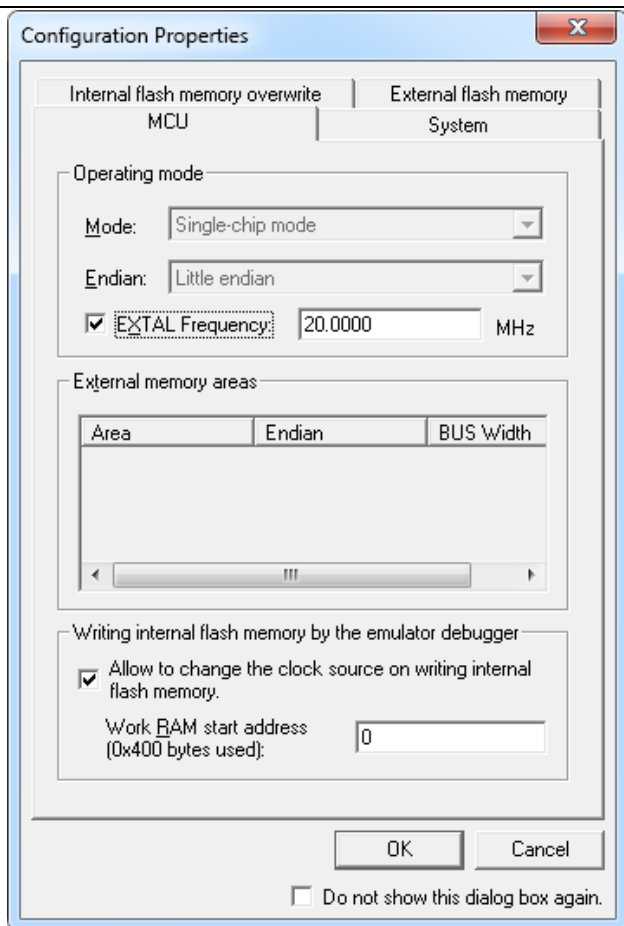
- Click the 'Startup and Communication' tab and ensure the Mode Pin, Register & Endian settings match the screenshot opposite. Ensure the FINE Baud Rate is set to 2000000bps. Once these settings have been confirmed, click the [OK] button to continue.
- The Flash Memory write program will be downloaded to the target.



- The 'Connecting...' dialog box will appear, showing the status of the connection process. Under default settings, this dialog box will disappear once the connection is complete.

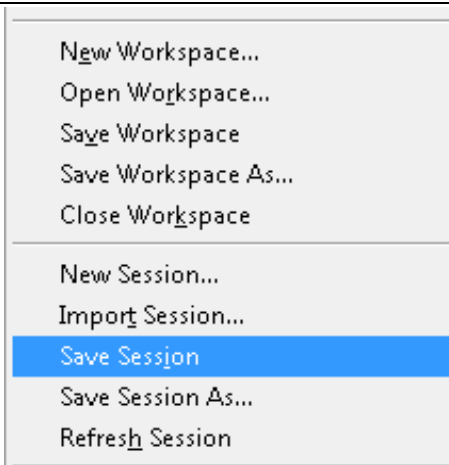


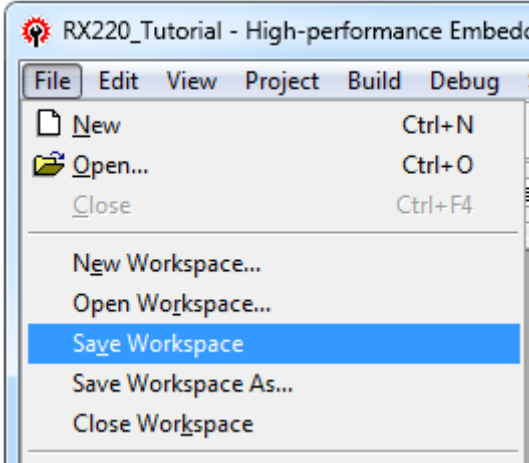
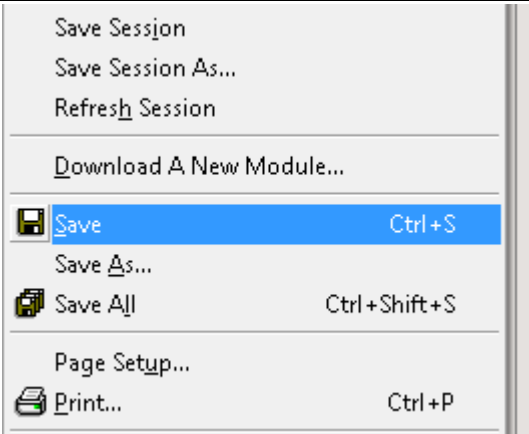

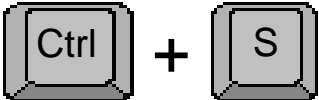
- Once the debugger has connected, the 'Configuration Properties' dialog box will appear.
- Ensure the following configurations are set:
  - Input Clock (EXTAL): 20.0000 MHz
  - Clock (flash memory): Allow to change
  - Work RAM Start Address: 0
- Once the settings have been reviewed, click [OK] to proceed. The Output Window in HEW will show 'Connected'.
- The connection to the target will activate the debugger buttons on the HEW toolbar. The function of these buttons will be explained in subsequent sections of this tutorial.



Now is a good time to save the High-performance Embedded Workshop session.

- Select 'File' | 'Save Session'.



<p>If you have changed any workspace settings now is a good time to save the workspace.</p> <ul style="list-style-type: none"> <li>• Select 'File'   'Save Workspace'.</li> </ul>	
<p>If you make any changes to files in HEW and want to preserve these change, you can save them by:</p> <ul style="list-style-type: none"> <li>• Select 'File'   'Save'.</li> </ul>	
<p>You can also save files by clicking the 'Save' or 'Save All' buttons from the HEW toolbar.</p>	
<p>You can also save files using the [CTRL+S] shortcut.</p>	



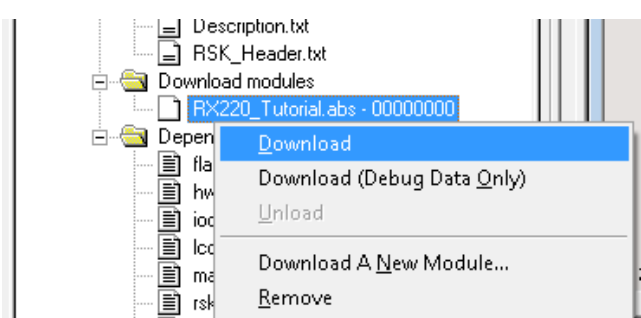
## 6. Downloading and Running the Tutorial

### 6.1 Downloading the Program Code

Now the code has been built in HEW it needs to be downloaded to the RSK.

As you are now connected to the target you should see an additional category in the workspace view called 'Download Modules'.

- Right click on the download module listed and select 'Download'.
- On completion the debugger and code are ready to be executed.



### 6.2 Running the Tutorial

Once the program has been downloaded onto the RSK device, the program can be executed. Click the 'Reset Go' button to begin the program. It is recommended that you run through the program once first, and then continue to the review section.







## 7. Reviewing the Tutorial Program



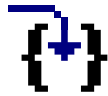
This section will look at each section of the tutorial code, how it works, and how it could be altered to be implemented into more complex code.

### 7.1 Program Initialisation

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

Ensuring the tutorial program has been downloaded onto the RX220; press the 'Reset CPU' button on the Debug toolbar.	
----------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

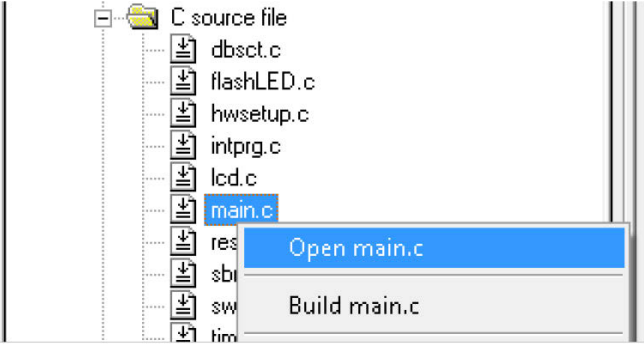

<ul style="list-style-type: none"> <li>The File window will open the tutorial code at the entry point. An arrow and a yellow highlight marks the current position of the program counter.</li> <li>Use these buttons to switch between 'source, disassembly and mixed modes'.</li> </ul>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">Line</th> <th style="width: 15%;">Source Ad...</th> <th style="width: 5%;">O</th> <th style="width: 5%;">S</th> <th style="width: 70%;">Source</th> </tr> </thead> <tbody> <tr><td>88</td><td></td><td></td><td></td><td><code>void PowerON_Reset_PC (void);</code></td></tr> <tr><td>89</td><td></td><td></td><td></td><td></td></tr> <tr><td>90</td><td></td><td></td><td></td><td><code>/* *****</code></td></tr> <tr><td>91</td><td></td><td></td><td></td><td><code>* Function Name: PowerON_Reset_PC</code></td></tr> <tr><td>92</td><td></td><td></td><td></td><td><code>* Description : This program is the MCU's entry point from a power-on reset.</code></td></tr> <tr><td>93</td><td></td><td></td><td></td><td><code>* The function configures the MCU stack, then calls the</code></td></tr> <tr><td>94</td><td></td><td></td><td></td><td><code>* HardwareSetup function and main function sequentially.</code></td></tr> <tr><td>95</td><td></td><td></td><td></td><td><code>*</code></td></tr> <tr><td>96</td><td></td><td></td><td></td><td><code>* Arguments : none</code></td></tr> <tr><td>97</td><td></td><td></td><td></td><td><code>* Return Value : none</code></td></tr> <tr><td>98</td><td></td><td></td><td></td><td><code>*****</code></td></tr> <tr style="background-color: yellow;"><td>98</td><td>FFFF8000</td><td></td><td></td><td><code>void PowerON_Reset_PC (void)</code></td></tr> <tr><td>99</td><td></td><td></td><td></td><td><code>{</code></td></tr> <tr><td>100</td><td></td><td></td><td></td><td><code>/* Initialise the MCU processor word */</code></td></tr> <tr><td>101</td><td>FFFF800E</td><td></td><td></td><td><code>set_intb({__sectop("C\$VECT")}));</code></td></tr> <tr><td>102</td><td></td><td></td><td></td><td></td></tr> <tr><td>103</td><td></td><td></td><td></td><td><code>/* Initialise the MCU stack area */</code></td></tr> <tr><td>104</td><td>FFFF8017</td><td></td><td></td><td><code>_INITSCT();</code></td></tr> <tr><td>105</td><td></td><td></td><td></td><td></td></tr> <tr><td>106</td><td></td><td></td><td></td><td><code>/* Configure the MCU and RSK hardware */</code></td></tr> <tr><td>107</td><td>FFFF801B</td><td></td><td></td><td><code>HardwareSetup();</code></td></tr> <tr><td>108</td><td></td><td></td><td></td><td></td></tr> <tr><td>109</td><td></td><td></td><td></td><td><code>/* Execute a NOP instruction */</code></td></tr> <tr><td>110</td><td>FFFF801F</td><td></td><td></td><td><code>nop();</code></td></tr> <tr><td>111</td><td></td><td></td><td></td><td></td></tr> <tr><td>112</td><td></td><td></td><td></td><td><code>/* Set Ubit and Ibit for PSW */</code></td></tr> <tr><td>113</td><td>FFFF8020</td><td></td><td></td><td><code>set_psw(PSW_init);</code></td></tr> <tr><td>114</td><td></td><td></td><td></td><td></td></tr> <tr><td>115</td><td></td><td></td><td></td><td><code>/* Change the MCU's usermode from supervisor to user */</code></td></tr> <tr><td>116</td><td>FFFF8028</td><td></td><td></td><td><code>change_psw_pm_to_user_mode();</code></td></tr> <tr><td>117</td><td></td><td></td><td></td><td></td></tr> <tr><td>118</td><td></td><td></td><td></td><td><code>/* Call the main program function */</code></td></tr> <tr><td>119</td><td>FFFF803D</td><td></td><td></td><td><code>main();</code></td></tr> <tr><td>120</td><td></td><td></td><td></td><td></td></tr> <tr><td>121</td><td></td><td></td><td></td><td><code>/* Invoke a break interrupt */</code></td></tr> <tr><td>122</td><td>FFFF8041</td><td></td><td></td><td><code>brk();</code></td></tr> </tbody> </table>	Line	Source Ad...	O	S	Source	88				<code>void PowerON_Reset_PC (void);</code>	89					90				<code>/* *****</code>	91				<code>* Function Name: PowerON_Reset_PC</code>	92				<code>* Description : This program is the MCU's entry point from a power-on reset.</code>	93				<code>* The function configures the MCU stack, then calls the</code>	94				<code>* HardwareSetup function and main function sequentially.</code>	95				<code>*</code>	96				<code>* Arguments : none</code>	97				<code>* Return Value : none</code>	98				<code>*****</code>	98	FFFF8000			<code>void PowerON_Reset_PC (void)</code>	99				<code>{</code>	100				<code>/* Initialise the MCU processor word */</code>	101	FFFF800E			<code>set_intb({__sectop("C\$VECT")}));</code>	102					103				<code>/* Initialise the MCU stack area */</code>	104	FFFF8017			<code>_INITSCT();</code>	105					106				<code>/* Configure the MCU and RSK hardware */</code>	107	FFFF801B			<code>HardwareSetup();</code>	108					109				<code>/* Execute a NOP instruction */</code>	110	FFFF801F			<code>nop();</code>	111					112				<code>/* Set Ubit and Ibit for PSW */</code>	113	FFFF8020			<code>set_psw(PSW_init);</code>	114					115				<code>/* Change the MCU's usermode from supervisor to user */</code>	116	FFFF8028			<code>change_psw_pm_to_user_mode();</code>	117					118				<code>/* Call the main program function */</code>	119	FFFF803D			<code>main();</code>	120					121				<code>/* Invoke a break interrupt */</code>	122	FFFF8041			<code>brk();</code>
Line	Source Ad...	O	S	Source																																																																																																																																																																																						
88				<code>void PowerON_Reset_PC (void);</code>																																																																																																																																																																																						
89																																																																																																																																																																																										
90				<code>/* *****</code>																																																																																																																																																																																						
91				<code>* Function Name: PowerON_Reset_PC</code>																																																																																																																																																																																						
92				<code>* Description : This program is the MCU's entry point from a power-on reset.</code>																																																																																																																																																																																						
93				<code>* The function configures the MCU stack, then calls the</code>																																																																																																																																																																																						
94				<code>* HardwareSetup function and main function sequentially.</code>																																																																																																																																																																																						
95				<code>*</code>																																																																																																																																																																																						
96				<code>* Arguments : none</code>																																																																																																																																																																																						
97				<code>* Return Value : none</code>																																																																																																																																																																																						
98				<code>*****</code>																																																																																																																																																																																						
98	FFFF8000			<code>void PowerON_Reset_PC (void)</code>																																																																																																																																																																																						
99				<code>{</code>																																																																																																																																																																																						
100				<code>/* Initialise the MCU processor word */</code>																																																																																																																																																																																						
101	FFFF800E			<code>set_intb({__sectop("C\$VECT")}));</code>																																																																																																																																																																																						
102																																																																																																																																																																																										
103				<code>/* Initialise the MCU stack area */</code>																																																																																																																																																																																						
104	FFFF8017			<code>_INITSCT();</code>																																																																																																																																																																																						
105																																																																																																																																																																																										
106				<code>/* Configure the MCU and RSK hardware */</code>																																																																																																																																																																																						
107	FFFF801B			<code>HardwareSetup();</code>																																																																																																																																																																																						
108																																																																																																																																																																																										
109				<code>/* Execute a NOP instruction */</code>																																																																																																																																																																																						
110	FFFF801F			<code>nop();</code>																																																																																																																																																																																						
111																																																																																																																																																																																										
112				<code>/* Set Ubit and Ibit for PSW */</code>																																																																																																																																																																																						
113	FFFF8020			<code>set_psw(PSW_init);</code>																																																																																																																																																																																						
114																																																																																																																																																																																										
115				<code>/* Change the MCU's usermode from supervisor to user */</code>																																																																																																																																																																																						
116	FFFF8028			<code>change_psw_pm_to_user_mode();</code>																																																																																																																																																																																						
117																																																																																																																																																																																										
118				<code>/* Call the main program function */</code>																																																																																																																																																																																						
119	FFFF803D			<code>main();</code>																																																																																																																																																																																						
120																																																																																																																																																																																										
121				<code>/* Invoke a break interrupt */</code>																																																																																																																																																																																						
122	FFFF8041			<code>brk();</code>																																																																																																																																																																																						
<div style="margin-bottom: 10px;">  ← Source mode Button                 </div> <div style="margin-bottom: 10px;">  ← Mixed mode Button                 </div> <div style="margin-bottom: 10px;">  ← Disassembly mode Button                 </div>																																																																																																																																																																																										
Ensure the view is switched to 'source' before continuing.																																																																																																																																																																																										

<ul style="list-style-type: none"> <li>Highlight the 'HardwareSetup()' function call by double clicking in the center of the text.</li> </ul>	<pre> Line Source Ad... 0 S Source 88 void PowerON_Reset_PC (void); 89 90 91 92 93 94 95 96 97 98 FFFF8000 void PowerON_Reset_PC (void) 99 { 100 /* Initialise the MCU processor word */ 101 set_intb({_sectop("CSVECT")}); 102 103 104 /* Initialise the MCU stack area */ 105 _INITSCT(); 106 107 /* Configure the MCU and RSK hardware */ 108 HardwareSetup(); 109 110 /* Execute a NOP instruction */ 111 nop(); 112 113 /* Set Ubit and Ibit for PSW */ 114 set_psw(PSW_init); 115 116 /* Change the MCU's usermode from supervisor to user */ 117 change_psw_pm_to_user_mode(); 118 119 /* Call the main program function */ 120 main();             </pre>
<ul style="list-style-type: none"> <li>Click the 'Go to Cursor' button to run the program up to this point.</li> </ul> 	<pre> Line Source Ad... 0 S Source 108 HardwareSetup();             </pre>
<ul style="list-style-type: none"> <li>Click 'Step In' to enter the HardwareSetup function.</li> </ul> 	<pre> Line Source Ad... 0 S Source 108 HardwareSetup();             </pre>
<ul style="list-style-type: none"> <li>The program counter should now move to the HardwareSetup function definition. This function groups together several key functions that are used to ensure the device is setup correctly before the main program is executed.</li> </ul>	<pre> Line Source Ad... 0 S Source 83 84 85 86 87 FFFF85DE void HardwareSetup (void) 88 { 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103             </pre>
<ul style="list-style-type: none"> <li>Click 'Step In' again to enter the configure_operating_frequency function.</li> </ul> 	<pre> Line Source Ad... 0 S Source 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103             </pre>
<ul style="list-style-type: none"> <li>The configure_operating_frequency function is used to set the speed of the system clocks.</li> <li>We will now skip past the hardware setup functions to look at the tutorial's main program code.</li> </ul>	<pre> Line Source Ad... 0 S Source 98 99 100 101 102 103 104 FFFF85EA static void configure_operating_frequency (void) 105 { 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132             </pre>


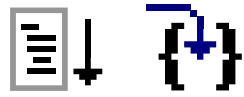
For further details regarding hardware configuration, please refer to the RSKRX220 User's Manual and the RX220 Hardware Manual.


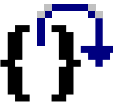

## 7.2 Main Function



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

<ul style="list-style-type: none"> <li>Find the main.c file from the file tree on the left hand side, then double click it to open the file, or right-click the filename and select 'Open main.c'.</li> </ul>																																																																																																																									
<ul style="list-style-type: none"> <li>Place an event at the call to main(); by double clicking in the On-Chip Breakpoint column next to the line to stop at.</li> </ul> <p>Note that two event points will appear because they share the same source address.</p>	<table border="1"> <thead> <tr> <th>Line</th> <th>Source Ad..</th> <th>O</th> <th>S</th> <th>Source</th> </tr> </thead> <tbody> <tr><td>123</td><td></td><td></td><td></td><td>* Function Name: main</td></tr> <tr><td>124</td><td></td><td></td><td></td><td>* Description : The main program function. Displays the Renesas</td></tr> <tr><td>125</td><td></td><td></td><td></td><td>onto the LCD display, then calls the 'flashLED'</td></tr> <tr><td>126</td><td></td><td></td><td></td><td>functions. The function then calls the statics</td></tr> <tr><td>127</td><td></td><td></td><td></td><td>before waiting in an infinite while loop.</td></tr> <tr><td>128</td><td></td><td></td><td></td><td>* Arguments : none</td></tr> <tr><td>129</td><td></td><td></td><td></td><td>* Return Value : none</td></tr> <tr><td>130</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>131</td><td>FFFF87AE</td><td>●</td><td></td><td>void main (void)</td></tr> <tr><td>132</td><td></td><td></td><td></td><td></td></tr> <tr><td>133</td><td>FFFF87AE</td><td>●</td><td></td><td>/* Initialise the debug LCD */</td></tr> <tr><td>134</td><td></td><td></td><td></td><td>Init_LCD();</td></tr> <tr><td>135</td><td></td><td></td><td></td><td></td></tr> <tr><td>136</td><td></td><td></td><td></td><td>/* Displays the Renesas splash screen */</td></tr> <tr><td>137</td><td>FFFF87B2</td><td></td><td></td><td>Display_LCD(LCD_LINE1, "Renesas");</td></tr> <tr><td>138</td><td>FFFF87BE</td><td></td><td></td><td>Display_LCD(LCD_LINE2, NICKNAME);</td></tr> <tr><td>139</td><td></td><td></td><td></td><td></td></tr> <tr><td>140</td><td></td><td></td><td></td><td>/* Begins the initial LED flash sequence */</td></tr> <tr><td>141</td><td>FFFF87CB</td><td></td><td></td><td>Flash_LED();</td></tr> <tr><td>142</td><td></td><td></td><td></td><td></td></tr> <tr><td>143</td><td></td><td></td><td></td><td>/* Begins the ADC-varying flash Sequence */</td></tr> <tr><td>144</td><td>FFFF87CF</td><td></td><td></td><td>Timer_ADC();</td></tr> <tr><td>145</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	Line	Source Ad..	O	S	Source	123				* Function Name: main	124				* Description : The main program function. Displays the Renesas	125				onto the LCD display, then calls the 'flashLED'	126				functions. The function then calls the statics	127				before waiting in an infinite while loop.	128				* Arguments : none	129				* Return Value : none	130				*****	131	FFFF87AE	●		void main (void)	132					133	FFFF87AE	●		/* Initialise the debug LCD */	134				Init_LCD();	135					136				/* Displays the Renesas splash screen */	137	FFFF87B2			Display_LCD(LCD_LINE1, "Renesas");	138	FFFF87BE			Display_LCD(LCD_LINE2, NICKNAME);	139					140				/* Begins the initial LED flash sequence */	141	FFFF87CB			Flash_LED();	142					143				/* Begins the ADC-varying flash Sequence */	144	FFFF87CF			Timer_ADC();	145				
Line	Source Ad..	O	S	Source																																																																																																																					
123				* Function Name: main																																																																																																																					
124				* Description : The main program function. Displays the Renesas																																																																																																																					
125				onto the LCD display, then calls the 'flashLED'																																																																																																																					
126				functions. The function then calls the statics																																																																																																																					
127				before waiting in an infinite while loop.																																																																																																																					
128				* Arguments : none																																																																																																																					
129				* Return Value : none																																																																																																																					
130				*****																																																																																																																					
131	FFFF87AE	●		void main (void)																																																																																																																					
132																																																																																																																									
133	FFFF87AE	●		/* Initialise the debug LCD */																																																																																																																					
134				Init_LCD();																																																																																																																					
135																																																																																																																									
136				/* Displays the Renesas splash screen */																																																																																																																					
137	FFFF87B2			Display_LCD(LCD_LINE1, "Renesas");																																																																																																																					
138	FFFF87BE			Display_LCD(LCD_LINE2, NICKNAME);																																																																																																																					
139																																																																																																																									
140				/* Begins the initial LED flash sequence */																																																																																																																					
141	FFFF87CB			Flash_LED();																																																																																																																					
142																																																																																																																									
143				/* Begins the ADC-varying flash Sequence */																																																																																																																					
144	FFFF87CF			Timer_ADC();																																																																																																																					
145																																																																																																																									
<ul style="list-style-type: none"> <li>It is also possible to set breakpoints by selecting the line to stop at and pressing the [F9] key.</li> </ul> 																																																																																																																									

The E1 emulator 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 RX Family E1/E20 Emulator User's Manual

<ul style="list-style-type: none"> <li>Press 'Reset Go' on the Debug toolbar.</li> </ul> 	<table border="1"> <thead> <tr> <th>Line</th> <th>Source Ad...</th> <th>D</th> <th>S</th> <th>Source</th> </tr> </thead> <tbody> <tr><td>123</td><td></td><td></td><td></td><td>* Function Name: main</td></tr> <tr><td>124</td><td></td><td></td><td></td><td>* Description : The main program function. Displays the Renesas splash screen</td></tr> <tr><td>125</td><td></td><td></td><td></td><td>* onto the LCD display, then calls the 'flashLED' and 'TimerADC'</td></tr> <tr><td>126</td><td></td><td></td><td></td><td>* functions. The function then calls the statics test routine,</td></tr> <tr><td>127</td><td></td><td></td><td></td><td>* before waiting in an infinite while loop.</td></tr> <tr><td>128</td><td></td><td></td><td></td><td>* Arguments : none</td></tr> <tr><td>129</td><td></td><td></td><td></td><td>* Return Value : none</td></tr> <tr><td>130</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>131</td><td>FFFF87AE</td><td></td><td></td><td>void main (void)</td></tr> <tr><td>132</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>133</td><td></td><td></td><td></td><td>/* Initialise the debug LCD */</td></tr> <tr><td>134</td><td>FFFF87AE</td><td></td><td></td><td>Init_LCD();</td></tr> <tr><td>135</td><td></td><td></td><td></td><td></td></tr> <tr><td>136</td><td></td><td></td><td></td><td>/* Displays the Renesas splash screen */</td></tr> <tr><td>137</td><td>FFFF87B2</td><td></td><td></td><td>Display_LCD(LCD_LINE1, "Renesas");</td></tr> <tr><td>138</td><td>FFFF87BE</td><td></td><td></td><td>Display_LCD(LCD_LINE2, NICKNAME);</td></tr> <tr><td>139</td><td></td><td></td><td></td><td></td></tr> <tr><td>140</td><td></td><td></td><td></td><td>/* Begins the initial LED flash sequence */</td></tr> <tr><td>141</td><td>FFFF87CB</td><td></td><td></td><td>Flash_LED();</td></tr> <tr><td>142</td><td></td><td></td><td></td><td></td></tr> <tr><td>143</td><td></td><td></td><td></td><td>/* Begins the ADC-varying flash Sequence */</td></tr> <tr><td>144</td><td>FFFF87CF</td><td></td><td></td><td>Timer_ADC();</td></tr> <tr><td>145</td><td></td><td></td><td></td><td></td></tr> <tr><td>146</td><td></td><td></td><td></td><td>/* Begins the static variable test */</td></tr> <tr><td>147</td><td>FFFF87D3</td><td></td><td></td><td>static_test();</td></tr> <tr><td>148</td><td></td><td></td><td></td><td></td></tr> <tr><td>149</td><td></td><td></td><td></td><td>/* Infinite while loop */</td></tr> <tr><td>150</td><td>FFFF87D6</td><td></td><td></td><td>while (1)</td></tr> <tr><td>151</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>152</td><td></td><td></td><td></td><td>/* Sample operation is performed by interrupt routines */</td></tr> <tr><td>153</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>154</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>155</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>156</td><td></td><td></td><td></td><td>* End of function main</td></tr> <tr><td>157</td><td></td><td></td><td></td><td>*****</td></tr> </tbody> </table>	Line	Source Ad...	D	S	Source	123				* Function Name: main	124				* Description : The main program function. Displays the Renesas splash screen	125				* onto the LCD display, then calls the 'flashLED' and 'TimerADC'	126				* functions. The function then calls the statics test routine,	127				* before waiting in an infinite while loop.	128				* Arguments : none	129				* Return Value : none	130				*****	131	FFFF87AE			void main (void)	132				{	133				/* Initialise the debug LCD */	134	FFFF87AE			Init_LCD();	135					136				/* Displays the Renesas splash screen */	137	FFFF87B2			Display_LCD(LCD_LINE1, "Renesas");	138	FFFF87BE			Display_LCD(LCD_LINE2, NICKNAME);	139					140				/* Begins the initial LED flash sequence */	141	FFFF87CB			Flash_LED();	142					143				/* Begins the ADC-varying flash Sequence */	144	FFFF87CF			Timer_ADC();	145					146				/* Begins the static variable test */	147	FFFF87D3			static_test();	148					149				/* Infinite while loop */	150	FFFF87D6			while (1)	151				{	152				/* Sample operation is performed by interrupt routines */	153				}	154				}	155				*****	156				* End of function main	157				*****
Line	Source Ad...	D	S	Source																																																																																																																																																																																	
123				* Function Name: main																																																																																																																																																																																	
124				* Description : The main program function. Displays the Renesas splash screen																																																																																																																																																																																	
125				* onto the LCD display, then calls the 'flashLED' and 'TimerADC'																																																																																																																																																																																	
126				* functions. The function then calls the statics test routine,																																																																																																																																																																																	
127				* before waiting in an infinite while loop.																																																																																																																																																																																	
128				* Arguments : none																																																																																																																																																																																	
129				* Return Value : none																																																																																																																																																																																	
130				*****																																																																																																																																																																																	
131	FFFF87AE			void main (void)																																																																																																																																																																																	
132				{																																																																																																																																																																																	
133				/* Initialise the debug LCD */																																																																																																																																																																																	
134	FFFF87AE			Init_LCD();																																																																																																																																																																																	
135																																																																																																																																																																																					
136				/* Displays the Renesas splash screen */																																																																																																																																																																																	
137	FFFF87B2			Display_LCD(LCD_LINE1, "Renesas");																																																																																																																																																																																	
138	FFFF87BE			Display_LCD(LCD_LINE2, NICKNAME);																																																																																																																																																																																	
139																																																																																																																																																																																					
140				/* Begins the initial LED flash sequence */																																																																																																																																																																																	
141	FFFF87CB			Flash_LED();																																																																																																																																																																																	
142																																																																																																																																																																																					
143				/* Begins the ADC-varying flash Sequence */																																																																																																																																																																																	
144	FFFF87CF			Timer_ADC();																																																																																																																																																																																	
145																																																																																																																																																																																					
146				/* Begins the static variable test */																																																																																																																																																																																	
147	FFFF87D3			static_test();																																																																																																																																																																																	
148																																																																																																																																																																																					
149				/* Infinite while loop */																																																																																																																																																																																	
150	FFFF87D6			while (1)																																																																																																																																																																																	
151				{																																																																																																																																																																																	
152				/* Sample operation is performed by interrupt routines */																																																																																																																																																																																	
153				}																																																																																																																																																																																	
154				}																																																																																																																																																																																	
155				*****																																																																																																																																																																																	
156				* End of function main																																																																																																																																																																																	
157				*****																																																																																																																																																																																	
<ul style="list-style-type: none"> <li>The code will execute to the event point. At this point all the device initialisation will have been completed. The code window will open 'main.c' and show the new position of the program counter.</li> </ul>	<table border="1"> <tbody> <tr><td>140</td><td></td><td></td><td></td><td>/* Begins the initial LED flash sequence */</td></tr> <tr><td>141</td><td>FFFF87CB</td><td></td><td></td><td>Flash_LED();</td></tr> <tr><td>142</td><td></td><td></td><td></td><td></td></tr> <tr><td>143</td><td></td><td></td><td></td><td>/* Begins the ADC-varying flash Sequence */</td></tr> <tr><td>144</td><td>FFFF87CF</td><td></td><td></td><td>Timer_ADC();</td></tr> <tr><td>145</td><td></td><td></td><td></td><td></td></tr> <tr><td>146</td><td></td><td></td><td></td><td>/* Begins the static variable test */</td></tr> <tr><td>147</td><td>FFFF87D3</td><td></td><td></td><td>static_test();</td></tr> </tbody> </table>	140				/* Begins the initial LED flash sequence */	141	FFFF87CB			Flash_LED();	142					143				/* Begins the ADC-varying flash Sequence */	144	FFFF87CF			Timer_ADC();	145					146				/* Begins the static variable test */	147	FFFF87D3			static_test();																																																																																																																																												
140				/* Begins the initial LED flash sequence */																																																																																																																																																																																	
141	FFFF87CB			Flash_LED();																																																																																																																																																																																	
142																																																																																																																																																																																					
143				/* Begins the ADC-varying flash Sequence */																																																																																																																																																																																	
144	FFFF87CF			Timer_ADC();																																																																																																																																																																																	
145																																																																																																																																																																																					
146				/* Begins the static variable test */																																																																																																																																																																																	
147	FFFF87D3			static_test();																																																																																																																																																																																	
<ul style="list-style-type: none"> <li>Support for the LCD display is included in the tutorial code. We do not need to be concerned about the details of the LCD interface – except that the interface is write-only and so is not affected if the LCD display is attached or not.</li> </ul>	<table border="1"> <thead> <tr> <th>Line</th> <th>Source Ad...</th> <th>D</th> <th>S</th> <th>Source</th> </tr> </thead> <tbody> <tr><td>66</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>67</td><td></td><td></td><td></td><td>* Function Name: Flash_LED</td></tr> <tr><td>68</td><td></td><td></td><td></td><td>* Description : The LED flash function used at the beginning of the program</td></tr> <tr><td>69</td><td></td><td></td><td></td><td>* Arguments : none</td></tr> <tr><td>70</td><td></td><td></td><td></td><td>* Return Value : none</td></tr> <tr><td>71</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>72</td><td>FFFF858C</td><td></td><td></td><td>void Flash_LED (void)</td></tr> <tr><td>73</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>74</td><td></td><td></td><td></td><td>/* Declare a delay count variable */</td></tr> <tr><td>75</td><td>FFFF8590</td><td></td><td></td><td>volatile uint32_t ulLed_Delay = 0;</td></tr> <tr><td>76</td><td></td><td></td><td></td><td></td></tr> <tr><td>77</td><td></td><td></td><td></td><td>/* Declare a flash count variable */</td></tr> <tr><td>78</td><td>FFFF8593</td><td></td><td></td><td>uint16_t Flash_Count = LEDFLASHCOUNT;</td></tr> <tr><td>79</td><td></td><td></td><td></td><td></td></tr> <tr><td>80</td><td></td><td></td><td></td><td>/* Flash the LEDs for 200 times or until a user switch is pressed */</td></tr> <tr><td>81</td><td>FFFF8598</td><td></td><td></td><td>while ((0x00 == g_switch_flag) &amp;&amp; (0 &lt; --Flash_Count))</td></tr> <tr><td>82</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>83</td><td>FFFF85A0</td><td></td><td></td><td>for (ulLed_Delay = 0; ulLed_Delay &lt; LEDFLASHDELAY; ++ulLed_Delay)</td></tr> <tr><td>84</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>85</td><td></td><td></td><td></td><td>/* Delay briefly for LED flash */</td></tr> <tr><td>86</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>87</td><td></td><td></td><td></td><td></td></tr> <tr><td>88</td><td></td><td></td><td></td><td>/* Toggles the LEDs after a specific delay. */</td></tr> <tr><td>89</td><td>FFFF85B3</td><td></td><td></td><td>Toggle_LED();</td></tr> <tr><td>90</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>91</td><td></td><td></td><td></td><td></td></tr> <tr><td>92</td><td></td><td></td><td></td><td>/* Reset the g_switch_flag flag variable */</td></tr> <tr><td>93</td><td>FFFF85C2</td><td></td><td></td><td>g_switch_flag = 0x00;</td></tr> <tr><td>94</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>95</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>96</td><td></td><td></td><td></td><td>* End of function Flash_LED</td></tr> <tr><td>97</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>98</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	Line	Source Ad...	D	S	Source	66				*****	67				* Function Name: Flash_LED	68				* Description : The LED flash function used at the beginning of the program	69				* Arguments : none	70				* Return Value : none	71				*****	72	FFFF858C			void Flash_LED (void)	73				{	74				/* Declare a delay count variable */	75	FFFF8590			volatile uint32_t ulLed_Delay = 0;	76					77				/* Declare a flash count variable */	78	FFFF8593			uint16_t Flash_Count = LEDFLASHCOUNT;	79					80				/* Flash the LEDs for 200 times or until a user switch is pressed */	81	FFFF8598			while ((0x00 == g_switch_flag) && (0 < --Flash_Count))	82				{	83	FFFF85A0			for (ulLed_Delay = 0; ulLed_Delay < LEDFLASHDELAY; ++ulLed_Delay)	84				{	85				/* Delay briefly for LED flash */	86				}	87					88				/* Toggles the LEDs after a specific delay. */	89	FFFF85B3			Toggle_LED();	90				}	91					92				/* Reset the g_switch_flag flag variable */	93	FFFF85C2			g_switch_flag = 0x00;	94				}	95				*****	96				* End of function Flash_LED	97				*****	98														
Line	Source Ad...	D	S	Source																																																																																																																																																																																	
66				*****																																																																																																																																																																																	
67				* Function Name: Flash_LED																																																																																																																																																																																	
68				* Description : The LED flash function used at the beginning of the program																																																																																																																																																																																	
69				* Arguments : none																																																																																																																																																																																	
70				* Return Value : none																																																																																																																																																																																	
71				*****																																																																																																																																																																																	
72	FFFF858C			void Flash_LED (void)																																																																																																																																																																																	
73				{																																																																																																																																																																																	
74				/* Declare a delay count variable */																																																																																																																																																																																	
75	FFFF8590			volatile uint32_t ulLed_Delay = 0;																																																																																																																																																																																	
76																																																																																																																																																																																					
77				/* Declare a flash count variable */																																																																																																																																																																																	
78	FFFF8593			uint16_t Flash_Count = LEDFLASHCOUNT;																																																																																																																																																																																	
79																																																																																																																																																																																					
80				/* Flash the LEDs for 200 times or until a user switch is pressed */																																																																																																																																																																																	
81	FFFF8598			while ((0x00 == g_switch_flag) && (0 < --Flash_Count))																																																																																																																																																																																	
82				{																																																																																																																																																																																	
83	FFFF85A0			for (ulLed_Delay = 0; ulLed_Delay < LEDFLASHDELAY; ++ulLed_Delay)																																																																																																																																																																																	
84				{																																																																																																																																																																																	
85				/* Delay briefly for LED flash */																																																																																																																																																																																	
86				}																																																																																																																																																																																	
87																																																																																																																																																																																					
88				/* Toggles the LEDs after a specific delay. */																																																																																																																																																																																	
89	FFFF85B3			Toggle_LED();																																																																																																																																																																																	
90				}																																																																																																																																																																																	
91																																																																																																																																																																																					
92				/* Reset the g_switch_flag flag variable */																																																																																																																																																																																	
93	FFFF85C2			g_switch_flag = 0x00;																																																																																																																																																																																	
94				}																																																																																																																																																																																	
95				*****																																																																																																																																																																																	
96				* End of function Flash_LED																																																																																																																																																																																	
97				*****																																																																																																																																																																																	
98																																																																																																																																																																																					
<ul style="list-style-type: none"> <li>Insert event points on the Flash_LED, Timer_ADC and static_test function calls.</li> </ul>	<table border="1"> <tbody> <tr><td>84</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>85</td><td></td><td></td><td></td><td>/* Delay briefly for LED flash */</td></tr> <tr><td>86</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>87</td><td></td><td></td><td></td><td></td></tr> <tr><td>88</td><td></td><td></td><td></td><td>/* Toggles the LEDs after a specific delay. */</td></tr> <tr><td>89</td><td>FFFF85B3</td><td></td><td></td><td>Toggle_LED();</td></tr> <tr><td>90</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>91</td><td></td><td></td><td></td><td></td></tr> <tr><td>92</td><td></td><td></td><td></td><td>/* Reset the g_switch_flag flag variable */</td></tr> <tr><td>93</td><td>FFFF85C2</td><td></td><td></td><td>g_switch_flag = 0x00;</td></tr> <tr><td>94</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>95</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>96</td><td></td><td></td><td></td><td>* End of function Flash_LED</td></tr> <tr><td>97</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>98</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	84				{	85				/* Delay briefly for LED flash */	86				}	87					88				/* Toggles the LEDs after a specific delay. */	89	FFFF85B3			Toggle_LED();	90				}	91					92				/* Reset the g_switch_flag flag variable */	93	FFFF85C2			g_switch_flag = 0x00;	94				}	95				*****	96				* End of function Flash_LED	97				*****	98																																																																																																													
84				{																																																																																																																																																																																	
85				/* Delay briefly for LED flash */																																																																																																																																																																																	
86				}																																																																																																																																																																																	
87																																																																																																																																																																																					
88				/* Toggles the LEDs after a specific delay. */																																																																																																																																																																																	
89	FFFF85B3			Toggle_LED();																																																																																																																																																																																	
90				}																																																																																																																																																																																	
91																																																																																																																																																																																					
92				/* Reset the g_switch_flag flag variable */																																																																																																																																																																																	
93	FFFF85C2			g_switch_flag = 0x00;																																																																																																																																																																																	
94				}																																																																																																																																																																																	
95				*****																																																																																																																																																																																	
96				* End of function Flash_LED																																																																																																																																																																																	
97				*****																																																																																																																																																																																	
98																																																																																																																																																																																					
<ul style="list-style-type: none"> <li>Press 'Go' to run the program up to the Flash_LED event point, then press 'Step In', to move the program counter to the beginning of the Flash_LED function definition.</li> </ul> 	<table border="1"> <tbody> <tr><td>84</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>85</td><td></td><td></td><td></td><td>/* Delay briefly for LED flash */</td></tr> <tr><td>86</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>87</td><td></td><td></td><td></td><td></td></tr> <tr><td>88</td><td></td><td></td><td></td><td>/* Toggles the LEDs after a specific delay. */</td></tr> <tr><td>89</td><td>FFFF85B3</td><td></td><td></td><td>Toggle_LED();</td></tr> <tr><td>90</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>91</td><td></td><td></td><td></td><td></td></tr> <tr><td>92</td><td></td><td></td><td></td><td>/* Reset the g_switch_flag flag variable */</td></tr> <tr><td>93</td><td>FFFF85C2</td><td></td><td></td><td>g_switch_flag = 0x00;</td></tr> <tr><td>94</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>95</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>96</td><td></td><td></td><td></td><td>* End of function Flash_LED</td></tr> <tr><td>97</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>98</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	84				{	85				/* Delay briefly for LED flash */	86				}	87					88				/* Toggles the LEDs after a specific delay. */	89	FFFF85B3			Toggle_LED();	90				}	91					92				/* Reset the g_switch_flag flag variable */	93	FFFF85C2			g_switch_flag = 0x00;	94				}	95				*****	96				* End of function Flash_LED	97				*****	98																																																																																																													
84				{																																																																																																																																																																																	
85				/* Delay briefly for LED flash */																																																																																																																																																																																	
86				}																																																																																																																																																																																	
87																																																																																																																																																																																					
88				/* Toggles the LEDs after a specific delay. */																																																																																																																																																																																	
89	FFFF85B3			Toggle_LED();																																																																																																																																																																																	
90				}																																																																																																																																																																																	
91																																																																																																																																																																																					
92				/* Reset the g_switch_flag flag variable */																																																																																																																																																																																	
93	FFFF85C2			g_switch_flag = 0x00;																																																																																																																																																																																	
94				}																																																																																																																																																																																	
95				*****																																																																																																																																																																																	
96				* End of function Flash_LED																																																																																																																																																																																	
97				*****																																																																																																																																																																																	
98																																																																																																																																																																																					
<ul style="list-style-type: none"> <li>The Flash_LED function uses a delay loop to toggle the LEDs periodically.</li> <li>The 'while' statement checks the value of the Flash_Count variable, which counts down with every LED flash. Once this reaches zero (or a switch is pressed) the function exits the 'while' loop.</li> </ul>	<table border="1"> <tbody> <tr><td>84</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>85</td><td></td><td></td><td></td><td>/* Delay briefly for LED flash */</td></tr> <tr><td>86</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>87</td><td></td><td></td><td></td><td></td></tr> <tr><td>88</td><td></td><td></td><td></td><td>/* Toggles the LEDs after a specific delay. */</td></tr> <tr><td>89</td><td>FFFF85B3</td><td></td><td></td><td>Toggle_LED();</td></tr> <tr><td>90</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>91</td><td></td><td></td><td></td><td></td></tr> <tr><td>92</td><td></td><td></td><td></td><td>/* Reset the g_switch_flag flag variable */</td></tr> <tr><td>93</td><td>FFFF85C2</td><td></td><td></td><td>g_switch_flag = 0x00;</td></tr> <tr><td>94</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>95</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>96</td><td></td><td></td><td></td><td>* End of function Flash_LED</td></tr> <tr><td>97</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>98</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	84				{	85				/* Delay briefly for LED flash */	86				}	87					88				/* Toggles the LEDs after a specific delay. */	89	FFFF85B3			Toggle_LED();	90				}	91					92				/* Reset the g_switch_flag flag variable */	93	FFFF85C2			g_switch_flag = 0x00;	94				}	95				*****	96				* End of function Flash_LED	97				*****	98																																																																																																													
84				{																																																																																																																																																																																	
85				/* Delay briefly for LED flash */																																																																																																																																																																																	
86				}																																																																																																																																																																																	
87																																																																																																																																																																																					
88				/* Toggles the LEDs after a specific delay. */																																																																																																																																																																																	
89	FFFF85B3			Toggle_LED();																																																																																																																																																																																	
90				}																																																																																																																																																																																	
91																																																																																																																																																																																					
92				/* Reset the g_switch_flag flag variable */																																																																																																																																																																																	
93	FFFF85C2			g_switch_flag = 0x00;																																																																																																																																																																																	
94				}																																																																																																																																																																																	
95				*****																																																																																																																																																																																	
96				* End of function Flash_LED																																																																																																																																																																																	
97				*****																																																																																																																																																																																	
98																																																																																																																																																																																					

<ul style="list-style-type: none"> <li>Click 'Go' to resume the program, and then press a switch to proceed. The program should halt at the event point set on the Timer_ADC function call.</li> </ul>	<pre> Line   Source Ad...  O  S  Source 102 103 104      /****** 105      * Function Name: start_timer 106      * Description  : Configures CMT channel 1 to call the 'CB_TimerADC' callback 107      *                function which starts the AD conversion 108      * Arguments   : none 109      * Return Value: none 110      ****** 111      static void start_timer (void) 112      { 113      /* Protection off */ 114      SYSTEM.PCR.WORD = PROTECTOFF; 115 116      /* Cancel the CMT1 module clock stop mode */ 117      MSTP_CMT1 = 0x0; 118 119      /* Cancel the CMT2 module clock stop mode */ 120      MSTP_CMT2 = 0x0; 121 122      /* Protection on */ 123      SYSTEM.PCR.WORD = PROTECTON; 124 125      /* Set CMT1 interrupt priority level to 10 */ 126      IPR(CMT1, CH11) = 0xA; 127      ..                     </pre>
<ul style="list-style-type: none"> <li>Press 'Step In' twice to step into the start_timer function.</li> </ul> 	<pre> 113      SYSTEM.PCR.WORD = PROTECTOFF; 114 115      /* Cancel the CMT1 module clock stop mode */ 116      MSTP_CMT1 = 0x0; 117 118      /* Cancel the CMT2 module clock stop mode */ 119      MSTP_CMT2 = 0x0; 120 121      /* Protection on */ 122      SYSTEM.PCR.WORD = PROTECTON; 123 124      /* Set CMT1 interrupt priority level to 10 */ 125      IPR(CMT1, CH11) = 0xA; 126      ..                     </pre>
<ul style="list-style-type: none"> <li>The start_timer function configures the timer used to periodically flash the LEDs.</li> </ul>	<pre> 126      ..                     </pre>
<ul style="list-style-type: none"> <li>Press the 'Step Out' button to exit the start_timer function.</li> </ul> 	<pre> Line   Source Ad...  O  S  Source 158 159      /****** 160      * Function Name: start_adc 161      * Description  : Initialises the S12AD ADC module's channel AN000 for 162      *                continuous scan operations, then starts the conversions. 163      * Arguments   : none 164      * Return Value: none 165      ****** 166      static void start_adc (void) 167      { 168      /* Protection off */ 169      SYSTEM.PCR.WORD = PROTECTOFF; 170 171      /* Cancel the S12AD module clock stop mode */ 172      MSTP_S12AD = 0x0; 173 174      /* Protection on */ 175      SYSTEM.PCR.WORD = PROTECTON; 176 177      /* Clear the S12AD interrupt flag */ 178      IR(S12AD, S12ADIO) = 0x0; 179 180      /* Set the S12AD interrupt level to 5 */ 181      IPR(S12AD, S12ADIO) = 0x5; 182 183      /* Enable S12AD interrupt requests */ 184      IEN(S12AD, S12ADIO) = 0x1; 185 186      /* Use the AN000 (Potentiometer) pin 187      as an I/O for peripheral functions */ 188      PORT4.PMR.BYTE = 0x01; 189 190      /* Set continuous scan mode */ 191      S12AD.ADCSR.BIT.ADCS = 0x2; 192 193      /* Selects AN000 */ 194      ..                     </pre>
<ul style="list-style-type: none"> <li>Press 'Step In'. The program should now reach the start_adc function.</li> </ul> 	<pre> 168      SYSTEM.PCR.WORD = PROTECTOFF; 169 170      /* Cancel the S12AD module clock stop mode */ 171      MSTP_S12AD = 0x0; 172 173      /* Protection on */ 174      SYSTEM.PCR.WORD = PROTECTON; 175 176      /* Clear the S12AD interrupt flag */ 177      IR(S12AD, S12ADIO) = 0x0; 178 179      /* Set the S12AD interrupt level to 5 */ 180      IPR(S12AD, S12ADIO) = 0x5; 181 182      /* Enable S12AD interrupt requests */ 183      IEN(S12AD, S12ADIO) = 0x1; 184 185      /* Use the AN000 (Potentiometer) pin 186      as an I/O for peripheral functions */ 187      PORT4.PMR.BYTE = 0x01; 188 189      /* Set continuous scan mode */ 190      S12AD.ADCSR.BIT.ADCS = 0x2; 191 192      /* Selects AN000 */ 193      ..                     </pre>
<ul style="list-style-type: none"> <li>The start_adc function configures the ADC unit to make repeat conversions of the voltage from the potentiometer RV1.</li> </ul>	<pre> 193      ..                     </pre>

<ul style="list-style-type: none"> <li>Press F5 to resume the code, where it will then halt at the break point on the static_test function call.</li> </ul> 	<table border="1"> <thead> <tr> <th>Line</th> <th>Source Ad...</th> <th>D</th> <th>S</th> <th>Source</th> </tr> </thead> <tbody> <tr><td>159</td><td></td><td></td><td></td><td>/******</td></tr> <tr><td>160</td><td></td><td></td><td></td><td>* Function Name: static_test</td></tr> <tr><td>161</td><td></td><td></td><td></td><td>* Description : Static variable test routine. The function replaces the</td></tr> <tr><td>162</td><td></td><td></td><td></td><td>contents of the string 'ucStr' with that of 'ucReplace', one</td></tr> <tr><td>163</td><td></td><td></td><td></td><td>element at a time. Right-click the variable 'ucStr', and</td></tr> <tr><td>164</td><td></td><td></td><td></td><td>select 'instant watch' - click add in the subsequent dialog.</td></tr> <tr><td>165</td><td></td><td></td><td></td><td>* If you step through the function, you can watch the string</td></tr> <tr><td>166</td><td></td><td></td><td></td><td>elements being overwritten with the new data.</td></tr> <tr><td>167</td><td></td><td></td><td></td><td>* Arguments : none</td></tr> <tr><td>168</td><td></td><td></td><td></td><td>* Return Value : none</td></tr> <tr><td>169</td><td></td><td></td><td></td><td>*****</td></tr> <tr><td>170</td><td>FFFF67D8</td><td></td><td></td><td>static void static_test (void)</td></tr> <tr><td>171</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>172</td><td></td><td></td><td></td><td>/* Declare loop count variable */</td></tr> <tr><td>173</td><td></td><td></td><td></td><td>uint8_t uicount = 0;</td></tr> <tr><td>174</td><td></td><td></td><td></td><td></td></tr> <tr><td>175</td><td></td><td></td><td></td><td>/* Write ucStr variable, "STATIC" to LCD */</td></tr> <tr><td>176</td><td>FFFF67DA</td><td></td><td></td><td>Display_LCD(LCD_LINE2, ucStr);</td></tr> <tr><td>177</td><td></td><td></td><td></td><td></td></tr> <tr><td>178</td><td></td><td></td><td></td><td>/* Begin for loop which writes one letter of ucReplace to the LCD at a time</td></tr> <tr><td>179</td><td></td><td></td><td></td><td>The nested while loops generate the delay between each letter change */</td></tr> <tr><td>180</td><td>FFFF67E9</td><td></td><td></td><td>for (uicount = 0; uicount &lt; 8u; uicount++)</td></tr> <tr><td>181</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>182</td><td></td><td></td><td></td><td>/* Start a one-shot timer to create a delay between each loop</td></tr> <tr><td>183</td><td></td><td></td><td></td><td>iteration */</td></tr> <tr><td>184</td><td>FFFF67EB</td><td></td><td></td><td>Timer_Delay(500u);</td></tr> <tr><td>185</td><td></td><td></td><td></td><td></td></tr> <tr><td>186</td><td></td><td></td><td></td><td>/* Replace letter number 'uiCount' of 'ucStr' from 'ucReplace' */</td></tr> <tr><td>187</td><td>FFFF67F3</td><td></td><td></td><td>ucStr[uicount] = ucReplace[uicount];</td></tr> <tr><td>188</td><td>FFFF67FC</td><td></td><td></td><td>Display_LCD(LCD_LINE2, ucStr);</td></tr> <tr><td>189</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>190</td><td></td><td></td><td></td><td></td></tr> <tr><td>191</td><td></td><td></td><td></td><td>/* Clear LCD Display */</td></tr> <tr><td>192</td><td>FFFF6810</td><td></td><td></td><td>ucStr[uicount] = '\0';</td></tr> <tr><td>193</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	Line	Source Ad...	D	S	Source	159				/******	160				* Function Name: static_test	161				* Description : Static variable test routine. The function replaces the	162				contents of the string 'ucStr' with that of 'ucReplace', one	163				element at a time. Right-click the variable 'ucStr', and	164				select 'instant watch' - click add in the subsequent dialog.	165				* If you step through the function, you can watch the string	166				elements being overwritten with the new data.	167				* Arguments : none	168				* Return Value : none	169				*****	170	FFFF67D8			static void static_test (void)	171				{	172				/* Declare loop count variable */	173				uint8_t uicount = 0;	174					175				/* Write ucStr variable, "STATIC" to LCD */	176	FFFF67DA			Display_LCD(LCD_LINE2, ucStr);	177					178				/* Begin for loop which writes one letter of ucReplace to the LCD at a time	179				The nested while loops generate the delay between each letter change */	180	FFFF67E9			for (uicount = 0; uicount < 8u; uicount++)	181				{	182				/* Start a one-shot timer to create a delay between each loop	183				iteration */	184	FFFF67EB			Timer_Delay(500u);	185					186				/* Replace letter number 'uiCount' of 'ucStr' from 'ucReplace' */	187	FFFF67F3			ucStr[uicount] = ucReplace[uicount];	188	FFFF67FC			Display_LCD(LCD_LINE2, ucStr);	189				}	190					191				/* Clear LCD Display */	192	FFFF6810			ucStr[uicount] = '\0';	193				
Line	Source Ad...	D	S	Source																																																																																																																																																																																	
159				/******																																																																																																																																																																																	
160				* Function Name: static_test																																																																																																																																																																																	
161				* Description : Static variable test routine. The function replaces the																																																																																																																																																																																	
162				contents of the string 'ucStr' with that of 'ucReplace', one																																																																																																																																																																																	
163				element at a time. Right-click the variable 'ucStr', and																																																																																																																																																																																	
164				select 'instant watch' - click add in the subsequent dialog.																																																																																																																																																																																	
165				* If you step through the function, you can watch the string																																																																																																																																																																																	
166				elements being overwritten with the new data.																																																																																																																																																																																	
167				* Arguments : none																																																																																																																																																																																	
168				* Return Value : none																																																																																																																																																																																	
169				*****																																																																																																																																																																																	
170	FFFF67D8			static void static_test (void)																																																																																																																																																																																	
171				{																																																																																																																																																																																	
172				/* Declare loop count variable */																																																																																																																																																																																	
173				uint8_t uicount = 0;																																																																																																																																																																																	
174																																																																																																																																																																																					
175				/* Write ucStr variable, "STATIC" to LCD */																																																																																																																																																																																	
176	FFFF67DA			Display_LCD(LCD_LINE2, ucStr);																																																																																																																																																																																	
177																																																																																																																																																																																					
178				/* Begin for loop which writes one letter of ucReplace to the LCD at a time																																																																																																																																																																																	
179				The nested while loops generate the delay between each letter change */																																																																																																																																																																																	
180	FFFF67E9			for (uicount = 0; uicount < 8u; uicount++)																																																																																																																																																																																	
181				{																																																																																																																																																																																	
182				/* Start a one-shot timer to create a delay between each loop																																																																																																																																																																																	
183				iteration */																																																																																																																																																																																	
184	FFFF67EB			Timer_Delay(500u);																																																																																																																																																																																	
185																																																																																																																																																																																					
186				/* Replace letter number 'uiCount' of 'ucStr' from 'ucReplace' */																																																																																																																																																																																	
187	FFFF67F3			ucStr[uicount] = ucReplace[uicount];																																																																																																																																																																																	
188	FFFF67FC			Display_LCD(LCD_LINE2, ucStr);																																																																																																																																																																																	
189				}																																																																																																																																																																																	
190																																																																																																																																																																																					
191				/* Clear LCD Display */																																																																																																																																																																																	
192	FFFF6810			ucStr[uicount] = '\0';																																																																																																																																																																																	
193																																																																																																																																																																																					
<ul style="list-style-type: none"> <li>Press F11 to step into the function.</li> </ul> 																																																																																																																																																																																					
<ul style="list-style-type: none"> <li>The static_test function initialises a character string with the contents of a static variable; then gradually replaces it, letter by letter, with another static string.</li> <li>Click 'Go' or press F5 to resume the program code. You should observe the word 'STATIC' appear on the second LCD line, to be gradually replaced with the string 'TESTTEST'. The program then changes the LCD back to the original message of 'RX220'.</li> </ul>																																																																																																																																																																																					

## 8. Additional Information

### Technical Support

For details on how to use High-performance Embedded Workshop (HEW), refer to the HEW manual available on the CD or from the web site.

For information about the RX220 Group microcontrollers refer to the RX220 Group Hardware Manual.

For information about the RX220 assembly language, refer to the RX Family Software Manual.

### Technical Contact Details

***Please refer to the contact details listed in section 7 of the “Quick Start Guide”***

General information on Renesas Microcontrollers can be found on the Renesas website at:

<http://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

This document may be, wholly or partially, subject to change without notice. All rights reserved. Duplication of this document, either in whole or part is prohibited without the written permission of Renesas Electronics Europe Limited.

© 2013 (2014) Renesas Electronics Europe Limited. All rights reserved.

© 2013 (2014) Renesas Electronics Corporation. All rights reserved.

© 2013 (2014) Renesas Solutions Corp. All rights reserved.



<b>REVISION HISTORY</b>	<b>RSKRX220 Tutorial Manual</b>
-------------------------	---------------------------------

Rev.	Date	Description	
		Page	Summary
1.00	Jan 15, 2013	—	First Edition issued
1.01	Apr 14, 2014	—	[2. List of Abbreviations and Acronyms] was updated.
		—	[Table of Contents] was updated.
		10	Section name of Section 4.2 was fixed.
		11 to 23	Frames were added to some explanations and figures.
		18	Source mode Button, Mixed mode Button and Disassembly mode Button were separated.

---

Renesas Starter Kit Tutorial Manual

Publication Date: Rev. 1.01 Apr 14, 2014

Published by: Renesas Electronics Corporation

---

**SALES OFFICES****Renesas Electronics Corporation**<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.**Renesas Electronics America Inc.**2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130**Renesas Electronics Canada Limited**1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220**Renesas Electronics Europe Limited**Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K  
Tel: +44-1628-585-100, Fax: +44-1628-585-900**Renesas Electronics Europe GmbH**Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-6503-0, Fax: +49-211-6503-1327**Renesas Electronics (China) Co., Ltd.**Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679**Renesas Electronics (Shanghai) Co., Ltd.**Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999**Renesas Electronics Hong Kong Limited**Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2265-6688, Fax: +852 2886-9022/9044**Renesas Electronics Taiwan Co., Ltd.**13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670**Renesas Electronics Singapore Pte. Ltd.**80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949  
Tel: +65-6213-0200, Fax: +65-6213-0300**Renesas Electronics Malaysia Sdn.Bhd.**Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510**Renesas Electronics Korea Co., Ltd.**12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

## RX220 Group