

RX210 Group (B Mask)

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; anticrime 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 RX210 (B Mask) 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 this RSK and the RX210 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.	RSKRX210B User's Manual	R20UT2604EG
Software Manual	Describes the functionality of the sample code, and its interaction with the Renesas Peripheral Driver Library (RPDL)	RSKRX210B Software Manual	R20UT2607EG
Tutorial	Provides a guide to setting up RSK environment, running sample code and debugging programs.	RSKRX210B Tutorial Manual	R20UT2605EG
Quick Start Guide	Provides simple instructions to setup the RSK and run the first sample, on a single A4 sheet.	RSKRX210B Quick Start Guide	R20UT2606EG
Schematics	Full detail circuit schematics of the RSK.	RSKRX210B Schematics	R20UT2603EG
Hardware Manual	Provides technical details of the RX210 microcontroller.	RX210 Group Hardware Manual	R01UH0037EJ

2. List of Abbreviations and Acronyms

Abbreviation	Full Form
ADC	Analog to Digital Converter
API	Application Programming Interface
CD	Compact Disc
CPU	Central Processing Unit
E1	E1 Emulator
E20	E20 Emulator
HEW	High-performance Embedded Workshop
LCD	Liquid Crystal Display
LED	Light Emitting Diode
ROM	Read-Only Memory
RSK	Renesas Starter Kit
USB	Universal Serial Bus
ADC	Analog to Digital Converter

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 and Connecting the E1 Debugger.....	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.....	21
8. Additional Information.....	25

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 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.

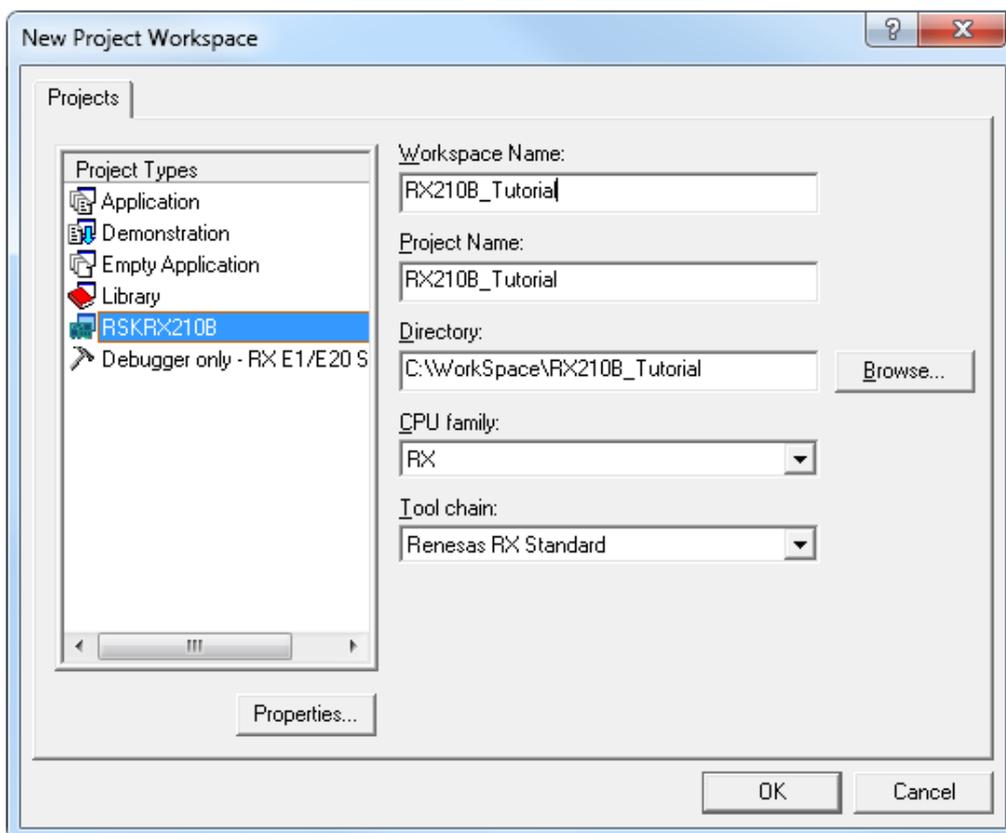
4. Project Workspace

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 and Connecting the E1 Debugger

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.



The example above shows the 'New Project Workspace' dialog with the RSKRX210B selected.

- Select the RX CPU family and 'Renesas RX Standard' toolchain.
- Select the 'RSKRX210B' 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 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, 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. 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.

- Select the 'Debug' build configuration



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 'Default' 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.

- Select the session:
"SessionRX_E1_E20_SYSTEM"

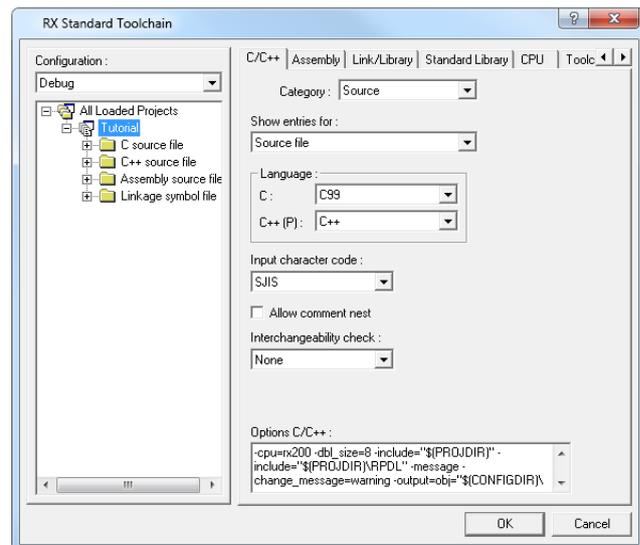


5. Building the Tutorial Program

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' or 'Multiple' from the 'Configuration' drop down list.

- 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.
- When complete close the dialog box by clicking [OK]



5.1 Building Code

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

- 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. 
- Selecting the 'Build' tool bar 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. 
- Pressing [F7]. This is equivalent to pressing the 'Build' button described above. 

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 5V variable power supply should be used. Refer to the RSKRX210B User 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.

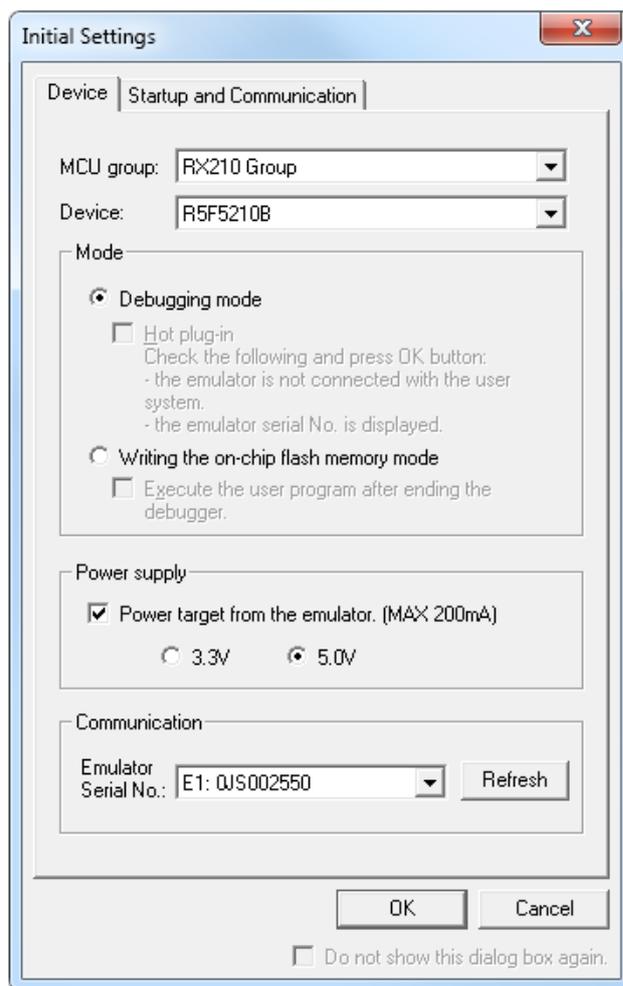
- Fit the LCD module to LCD connector on the board, via the header marked 'LCD'. Ensure all the pins of the connector are correctly inserted in the socket.
- Connect the E1 Debugger to a free USB port on your computer.
- Connect the E1 Debugger to the target hardware ensuring that it is plugged into 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

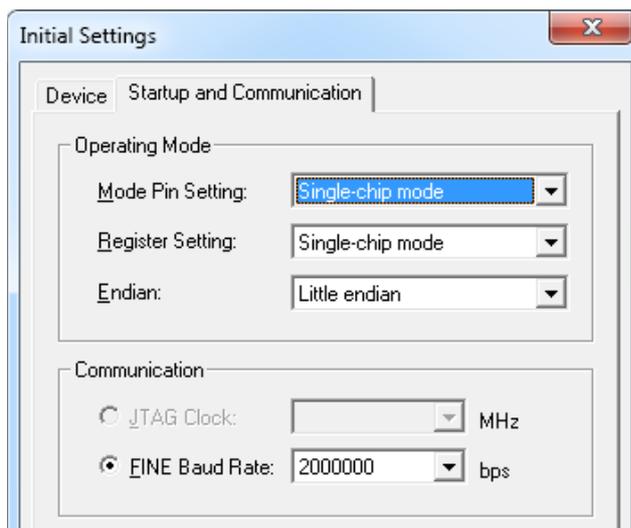
This section will take you through the process of connecting to the device, programming the Flash and executing the code.

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.

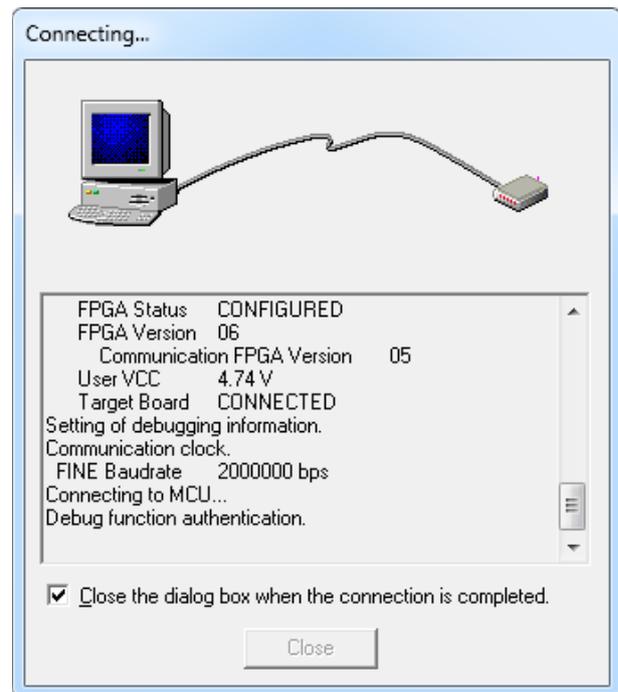
- Select the 'SessionRX_E1_E20_SYSTEM' debug session.
- Click the [Connect] button on the debug toolbar. 
- The 'Initial Settings' configuration dialog box will appear. Ensure the follow configurations are set:
 - MCU group: RX210 Group
 - Device: R5F5210B
 - Mode: Debugging mode
- If the E1 is to provide power to the CPU board, select 'Power target from the emulator' and choose the "5.0V" option. Otherwise connect a suitable power supply (refer to the RSKRX210B User Manual for details).



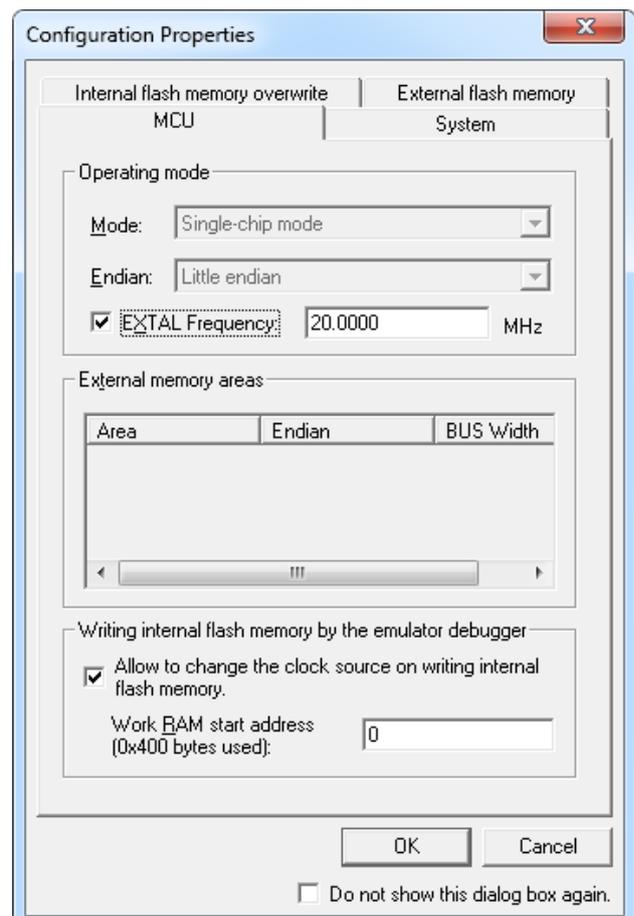
- Click the 'Startup and Communication' tab and ensure the Mode Pin, Register & Endian settings match the screenshot opposite. Ensure the FINE Baud Rate is set to 2000000bps. Once these settings have been confirmed, click the [OK] button to continue.
- The Flash Memory write program will be downloaded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

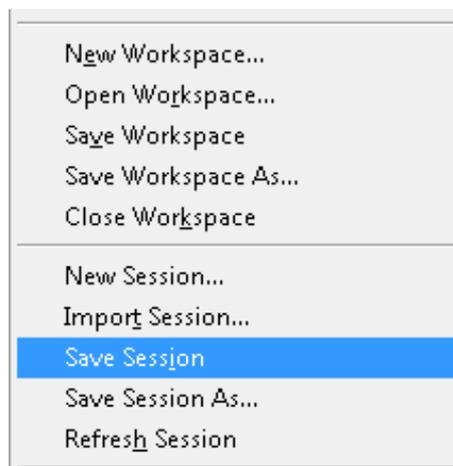


- Once the debugger has connected, the 'Configuration Properties' dialog box will appear.
- Ensure the following configurations are set:
 - Mode: Single-chip mode
 - Endian: Little Endian
 - Input Clock (EXTAL): 20.0000 MHz
 - Work RAM Start Address: 0
- Once the settings have been reviewed, click [OK] 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.



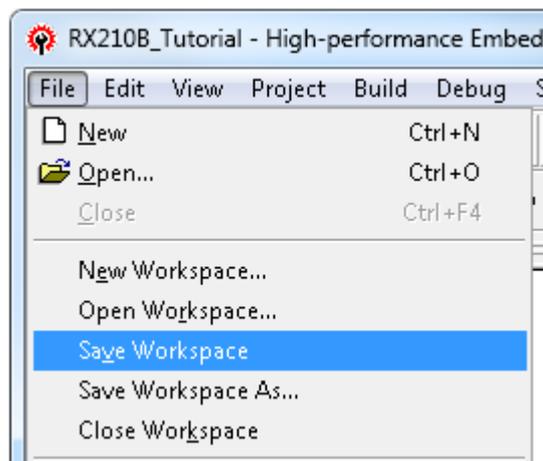
Now is a good time to save the High-performance Embedded Workshop session.

- Select 'File' | 'Save Session'.



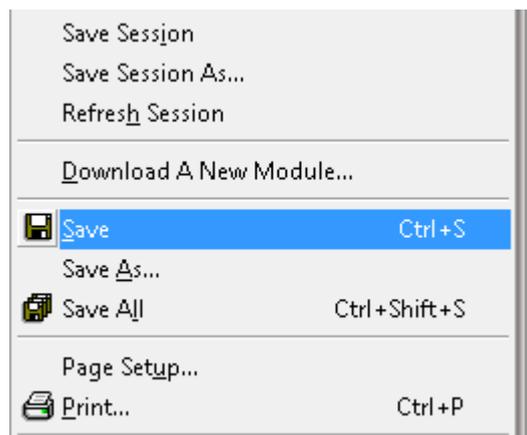
If you have changed any workspace settings now is a good time to save the workspace.

- Select 'File' | 'Save Workspace'.



If you make any changes to files in HEW and want to preserve these changes, you can save them by:

- Select 'File' | 'Save'.



You can also save files by clicking the 'Save' or 'Save All' buttons from the HEW toolbar.



You can also save files using the following keyboard shortcut:



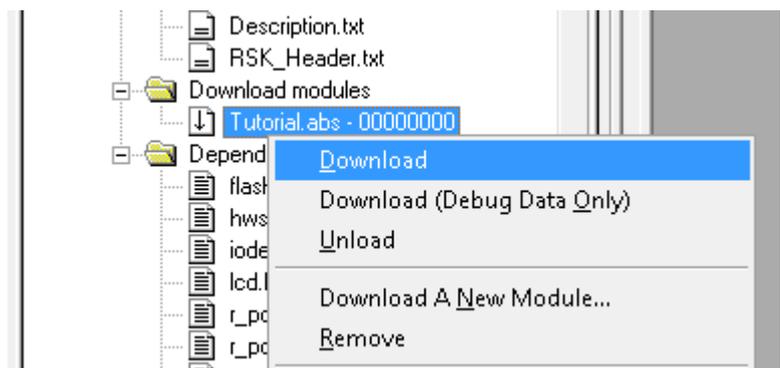
6. Downloading and Running the Tutorial

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.

It is recommended that a copy of the RX210 API Manual is made available, as the tutorial program uses RPD and it is outside the scope of this manual to fully document the API system.

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 RX210 (B Mask); press the 'Reset CPU' button on the Debug toolbar.



- 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.
- Use these buttons to switch between 'source, disassembly and mixed modes'.



Ensure the view is switched to 'source' before continuing.

Line	Source Ad..	O.	S.	Source
75				void PowerON_Reset_PC(void);
76				/* Main program function declaration */
77				void main(void);
78				
79				/* ===== */
80				* Outline : PowerON_Reset_PC
81				* Description : This program is the MCU's entry point from a power-on reset.
82				* The function configures the MCU stack, then calls the
83				* HardwareSetup function and main function sequentially.
84				* Argument : none
85				* Return value : none
86				*****
87	FFFF000D			void PowerON_Reset_PC(void)
88				{
89				/* Initialise the MCU processor word */
90	FFFF000E			set_intb(__sectop("CS\$VECT"));
91				
92				/* Initialise the MCU stack area */
93	FFFF0017			_INIT\$CT();
94				
95				/* Configure the MCU and RSK hardware */
96	FFFF001B			HardwareSetup();
97				
98				/* Change the MCU's usermode from supervisor to user */
99	FFFF001F			nop();
100	FFFF0020			set_psw(PSW_init);
101	FFFF0028			Change_PSW_PM_to_UserMode();
102				
103				/* Call the main program function */
104	FFFF003D			main();
105				
106				/* Invoke a break interrupt */
107	FFFF0041			brk();

- Highlight the 'HardwareSetup()' function call by double clicking in the center of the text.

Line	Source Ad...	O.	S.	Source
75				void PowerON_Reset_PC(void);
76				/* Main program function declaration */
77				void main(void);
78				
79				/*-----*/
80				* Outline : PowerON_Reset_PC
81				* Description : This program is the MCU's entry point from a power-on reset.
82				* The function configures the MCU stack, then calls the
83				* HardwareSetup function and main function sequentially.
84				* Argument : none
85				* Return value : none
86				/*-----*/
87	FFFF0000			void PowerON_Reset_PC(void)
88				{
89				/* Initialise the MCU processor word */
90	FFFF000E			set_intb(__sectop("C\$VECT"));
91				
92				/* Initialise the MCU stack area */
93	FFFF0017			_INITSTC();
94				
95				/* Configure the MCU and RSK hardware */
96	FFFF001B			HardwareSetup();
97				
98				/* Change the MCU's usermode from supervisor to user */
99	FFFF001F			nop();
100	FFFF0020			set_psw(PSW_init);
101	FFFF0028			Change_PSW_PM_to_UserMode();
102				
103				/* Call the main program function */
104	FFFF003D			main();
105				}

- Click the 'Go to Cursor' button to run the program up to this point.



- Click 'Step In' to enter the HardwareSetup function.



- 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.

Line	Source Ad...	O.	S.	Source
52				* Description : Contains all the setup functions called at device restart
53				* Argument : none
54				* Return value : none
55				/*-----*/
56	FFFF16F1			void HardwareSetup(void)
57				{
58	FFFF16F1			ConfigureOperatingFrequency();
59	FFFF16F4			ConfigureOutputPorts();
60	FFFF16F7			ConfigureInterrupts();
61	FFFF16FA			EnablePeripheralModules();
62				}
63				/*-----*/
64				* End of function HardwareSetup
65				/*-----*/
66				
67				/*-----*/
68				* Outline : ConfigureOperatingFrequency
69				* Description : Configures the clock settings for each of the device clocks
70				* Argument : none
71				* Return value : none
72				/*-----*/

- Click 'Step In' again to enter the ConfigureOperatingFrequency function.



- The `ConfigureOperatingFrequency` function is used to set the speed of the system clocks.
- Several RPDL APIs are called within this function - their exact function can be found by referring to the RPDL manual.
- We will now skip past the hardware setup functions to look at the tutorial's main program code.

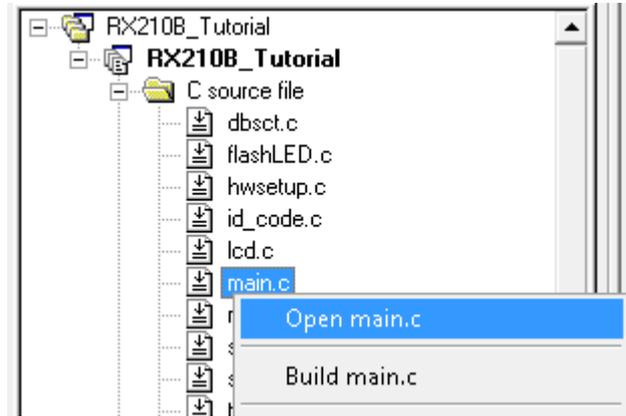
Line	S...	O.	S.	Source
69				/* Outline : ConfigureOperatingFrequency
70				*/
71				/* Description : Configures the clock settings for each of the device clocks
72				*/
73				/* Argument : none
74				*/
75				/* Return value : none
76				*/
77	FF1			void ConfigureOperatingFrequency(void)
78				{
79				/* Declare error flag */
80				bool err = true;
81	FF1			/* Configure the PLL to use the 20.0 MHz crystal as clock input */
82				err &= R_CGC_Set
83				{
84				PDL_CGC_CLK_MAIN,
85				PDL_CGC_BCLK_DIV_1 PDL_CGC_MAIN_RESONATOR,
86				20E6,
87				20E6,
88				20E6,
89				20E6,
90				PDL_NO_DATA
91				};
92				
93				/* Configure the clocks as follows -
94				Clock Description Frequency
95				-----
96				PLL Clock frequency.....100MHz
97				System Clock Frequency.....50MHz
98				Peripheral Module Clock B.....50MHz
99				Peripheral Module Clock D.....25MHz
100				FlashIF Clock.....25MHz
101				External Bus Clock.....25MHz */
102				
103	FF1			err &= R_CGC_Set
104				{
105				PDL_CGC_CLK_PLL,
106				PDL_CGC_BCLK_DIV_2,
107				100E6,
108				50E6,
109				50E6,
110				25E6,
111				25E6,
112				25E6,
113				};

For further details regarding hardware configuration, please refer to the RSKRX210B User's Manual and the RX210 Hardware Manual.

7.2 Main Functions

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

- 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'.



- Place an event at the call to main(); by double clicking in the On-Chip Breakpoint column next to the line to stop at.

Note that two event points will appear because they share the same source address.

Line	S...	O...	S...	Source
102				/* *****
103				* Outline : main
104				* Description : The main program function. Displ
105				* onto the LCD display, then calls
106				* functions. The function then cal
107				* before waiting in an infinite w
108				* Argument : none
109				* Return value : none
110				*****
111	FF1	●		void main(void)
112				{
113				/* Initialise the debug LCD */
114	FF1	●		Init_LCD();
115				
116				/* Displays the Renesas splash screen */
117	FF1			Display_LCD(LCD_LINE1, "Renesas");
118	FF1			Display_LCD(LCD_LINE2, NICKNAME);
119				
120				/* Begins the initial LED flash sequence */
121	FF1			Flash_LED();
122				
123				/* Begins the ADC-varying flash Sequence */
124	FF1			Timer_ADC();

- It is also possible to set breakpoints by selecting the line to stop at and pressing the [F9] key.



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

- Press 'Reset Go' on the Debug toolbar. 
- 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.
- 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.

Line	S...	O..	S..	Source
102				/* *****
103				* Outline : main
104				* Description : The main program function. Displays the Renesas splash screen
105				* onto the LCD display, then calls the 'flashLED' and 'TimerADC'
106				* functions. The function then calls the statics test routine,
107				* before waiting in an infinite while loop.
108				* Argument : none
109				* Return value : none
110				*****
111	FF1	●		void main(void)
112				{
113				/* Initialise the debug LCD */
114	FF1	●	→	Init_LCD();
115				
116				/* Displays the Renesas splash screen */
117	FF1			Display_LCD(LCD_LINE1, "Renesas");
118	FF1			Display_LCD(LCD_LINE2, NICKNAME);
119				
120				/* Begins the initial LED flash sequence */
121	FF1			Flash_LED();
122				
123				/* Begins the ADC-varying flash Sequence */
124	FF1			Timer_ADC();
125				
126				/* Begins the static variable test */
127	FF1			Static_Test();
128				
129				/* Infinite while loop */
130				/* Defines an infinite loop to keep the MCU running */
131	FF1			while(1);
132				}
133				/* *****
134				* End of function main
135				*****

- Insert event points on the Flash_LED, Timer_ADC and Static_Test function calls.

120				/* Begins the initial LED flash sequence */
121	FF1	●		Flash_LED();
122				
123				/* Begins the ADC-varying flash Sequence */
124	FF1	●		Timer_ADC();
125				
126				/* Begins the static variable test */
127	FF1	●		Static_Test();

- 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.



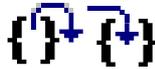
- The Flash_LED function uses RPDL functions to create a periodic CMT callback, which toggles the LEDs at regular intervals.
- The 'while' statement checks the value of the gFlashCount variable, which counts down with every LED flash. Once this reaches zero the function destroys the CMT timer and exits the 'while' loop.

Line	S...	O..	S..	Source
64				/* *****
65				* Outline : Flash LED
66				* Description : The LED flash function used at the beginning of the program
67				* Argument : none
68				* Return value : none
69				*****
70	FF1			void Flash_LED (void)
71				{
72				/* Declare error flag */
73	FF1			bool err = true;
74				
75				/* Configure compare match timer */
76	FF1			err &= R_CMT_Create
77				{
78				0,
79				PDL_CMT_PCLK_DIV_512,
80				0xFEO,
81				CB_TimerLED,
82				3
83				};
84				
85				/* Checks if the flash count has been reached,
86				or if a button has been pressed */
87	FF1			while((gSwitchFlag ==0) &&(gFlashCount > 0))
88				{
89				/* Waits for a switch press or 200 flashes to complete */
90				}
91				
92				/* Reset the gSwitchFlag flag variable */
93	FF1			gSwitchFlag = 0;
94				
95				/* Stop Timer */
96	FF1			err &= R_CMT_Destroy
97				{
98				0
99				};
100				
101				/* Halt in while loop when RPDL errors detected */
102	FF1			while(!err);

- 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.
- Press 'Step In' twice to step into the StartTimer function.
- The StartTimer function configures the timer used to periodically flash the LEDs.

Line	S...	O.	S.	Source
88				/* ***** */
89				* Outline : StartTimer
90				* Description : Configures CMT channel 0 to call the 'CB_TimerADC' callback
91				* function which starts the AD conversion
92				* Argument : none
93				/* ***** */
94	FF)			static void StartTimer(void)
95				{
96				/* Declare error flag */
97	FF)			bool err = true;
98				
99				/* Configure compare match timer */
100	FF)			err &= R_CMT_Create
101				{
102				0,
103				PDL_CMT_PCLK_DIV_512,
104				0x03FF,
105				CB_Timer,
106				3
107				};
108				
109				/* Halt in while loop when RPDL errors detected */
110	FF)			while(!err);

- Press the 'Step Out' button to exit the StartTimer function, then press 'Step In'. The program should now reach the StartADC function.



- The StartADC function configures the ADC unit to make repeat conversions of the voltage from the potentiometer RV1.

Line	S...	O.	S.	Source
116				/* ***** */
117				* Outline : StartADC
118				* Description : Initialises the ADC's channel 0 for one-shot operations and
119				* set the callback function.
120				* Argument : none
121				* Return value : none
122				/* ***** */
123	FF)			static void StartADC(void)
124				{
125				/* Declare error flag */
126	FF)			bool err = true;
127				
128				/* Set analog channel AN000 */
129	FF)			err &= R_ADC_12_Set
130				{
131				PDL_ADC_12_PIN_AN000_P40
132				};
133				
134				/* Configure the ADC in single mode, software triggered */
135	FF)			err &= R_ADC_12_CreateUnit
136				{
137				0,
138				PDL_ADC_12_SCAN_SINGLE,
139				PDL_NO_DATA,
140				PDL_NO_DATA,
141				PDL_NO_DATA,
142				PDL_NO_DATA,
143				PDL_NO_DATA,
144				CB_ADConversion,
145				6,
146				PDL_NO_FUNC,
147				PDL_NO_DATA
148				};
149				
150				/* Configure the ADC channel AN000 */
151	FF)			err &= R_ADC_12_CreateChannel
152				{
153				0,
154				0,
155				PDL_ADC_12_CH_GROUP_A1\
156				PDL_ADC_12_CH_SAMPLE_AND_HOLD_ENABLE,
157				SE-6
158				};

- Press F5 to resume the code, where it will then halt at the break point on the Static_Test function call.
- Press F11 to step into the function.



- 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.
- 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 reverts the LCD back to the original message of 'RX210B'.

Line	S...	D.	S.	Source
137				/* *****
138				* Outline : Static_Test
139				* Description : Static variable test routine. The function replaces the
140				* contents of the string 'ucStr' with that of 'ucReplace', one
141				* element at a time. Right-click the variable 'ucStr', and
142				* select 'instant watch' - click add in the subsequent dialog.
143				* If you step through the function, you can watch the string
144				* elements being overwritten with the new data.
145				* Argument : none
146				* Return value : none
147				***** */
148	FF1			void Static_Test(void)
149				{
150				/* Declare error flag */
151	FF1			bool err = true;
152				
153				/* Declare loop count variable */
154				uint8_t uiCount;
155				
156				/* Write ucStr variable, "STATIC" to LCD */
157	FF1			Display_LCD(LCD_LINE2,ucStr);
158				
159				/* Begin for loop which writes one letter of ucReplace to the LCD at a time
160				The nested while loops generate the delay between each letter change */
161	FF1			for (uiCount=0; uiCount<8; uiCount++)
162				{
163				/* Replace letter number 'uiCount' of 'ucStr' from 'ucReplace' */
164	FF1			ucStr[uiCount] = ucReplace[uiCount];
165	FF1			Display_LCD(LCD_LINE2,ucStr);
166				
167				/* Start a one-shot timer to create a delay between each loop
168				iteration */
169	FF1			err &= R_CMT_CreateOneShot
170				{
171				1,
172				PDL_NO_DATA,
173				600E-3,
174				PDL_NO_FUNC,
175				0
176				};
177				}
178				
179				/* Clear LCD Display */
180	FF1			ucStr[uiCount] = '\0';

- This is the extent of the tutorial code. For further information on the RPDL function calls used in the tutorial sample, please refer to *Renesas Peripheral Driver Library User's Manual*.

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 RX210 series microcontrollers refer to the RX210 Group Hardware Manual.

For information about the RX210 assembly language, refer to the RX200 Series Software Manual.

Online technical support and information is available at: <http://www.renesas.com/rskrx210b>

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 Renesas Electronics Europe Limited. All rights reserved.

© 2013 Renesas Electronics Corporation. All rights reserved.

© 2013 Renesas Solutions Corp. All rights reserved.

REVISION HISTORY	RSK RX210B Tutorial Manual
------------------	----------------------------

Rev.	Date	Description	
		Page	Summary
1.00	May 02, 2013	—	First Edition issued

Renesas Starter Kit User's Manual: Tutorial Manual

Publication Date: Rev. 1.00 May 02, 2013

Published by: Renesas Electronics Corporation



Renesas Electronics Corporation

<http://www.renesas.com>

SALES OFFICES

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.

2880 Scott Boulevard Santa Clara, CA 95050-2554, 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: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.

7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.

Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

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: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei, 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.

11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RX210 Group (B Mask)