User's Manual

16

R8C/LA8A Group

Renesas Starter Kit Tutorial Manual

RENESAS MCU
R8C Family / R8C/Lx Series

Renesas Electronics
www.renesas.com

Rev.1.01 Jun 2011

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. 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 categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “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.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. 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.
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 is 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 the RSK hardware functionality, and electrical characteristics. It is intended for users designing sample code on the RSK platform, using the many different incorporated peripheral devices.

The manual comprises of an overview of the capabilities of the RSK product, but does not intend to be a guide to embedded programming or hardware design. Further details regarding setting up the RSK and development environment can found in the tutorial manual.

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 R8C 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.	RSK R8CLA8A User's Manual	R20UT0284EG0100
Tutorial	Provides a guide to setting up RSK environment, running sample code and debugging programs.	RSK R8CLA8A Tutorial Manual	R20UT0285EG0100
Quick Start Guide	Provides simple instructions to setup the RSK and run the first sample, on a single A4 sheet.	RSK R8CLA8A Quick Start Guide	R20UT0286EG0100
Schematics	Full detail circuit schematics of the RSK.	RSK R8CLA8A Schematics	R20UT0287EG0100
Hardware Manual	Provides technical details of the R8CLA8A microcontroller.	RSK R8CLA8A Hardware Manual	REJ09B0556

2. List of Abbreviations and Acronyms

Abbreviation	Full Form
CD	Compact Disk
CPU	Central Processing Unit
E8a	E8a Emulator
HEW	High-performance Embedded Workshop
LCD	Liquid Crystal Display
LED	Light Emitting Diode
RSK	Renesas Starter Kit
USB	Universal Serial Bus

Table of Contents

1. Overview

1.1 Purpose

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

1.2 Features

This RSK provides an evaluation of the following features:

- Renesas microcontroller programming
- User code debugging
- User circuitry such as switches, LEDs and a potentiometer
- Sample application
- Sample peripheral device 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 E8a emulator. Please refer the relevant user manuals for more in-depth information.

3. Tutorial Project Workspace

The workspace includes all of the files for two build configurations, ‘Build’ 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 (HEW) 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.

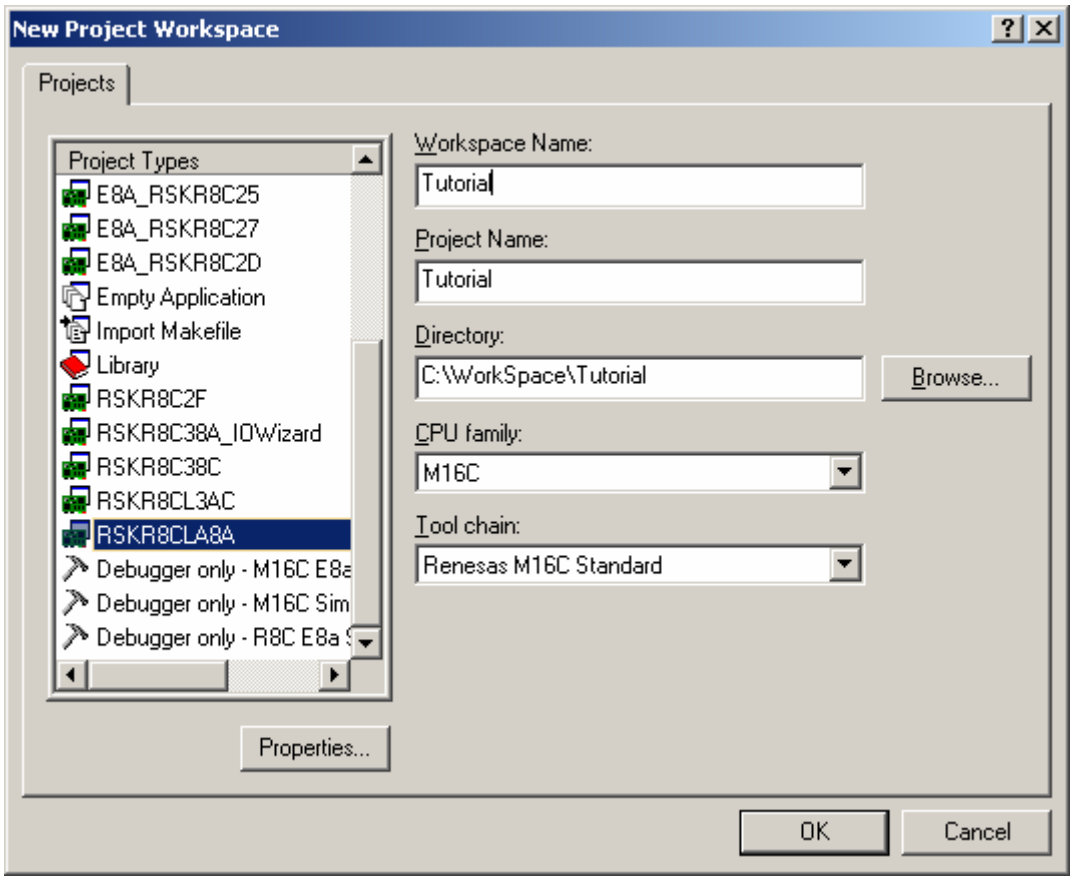
4. Project Workspace

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 E8a 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 R8C/LA8A selected.

- Select the M16C CPU family and 'Renesas M16C Standard' toolchain.
- Select the 'RSKR8CLA8A' 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 tool bar. 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 tool bar. The options vary between Renesas Starter Kit types however one will always start Debug and include the type of debug interface. The alternate selection will be 'SessionR8C_E8a_SYSTEM'. The purpose of the debug sessions is to allow the use of different debugger tools or different debugger settings on the same project.

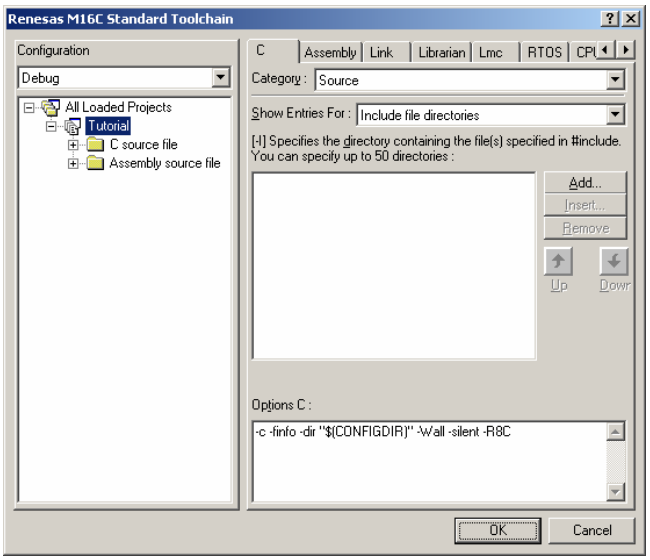
- Select the session:
  "SessionR8C_E8a_SYSTEM"

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 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’ tool bar button. This will build everything in the project that has not been excluded from the build. This includes the standard library.

- 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 more power than the E8a debugger is capable of supplying; in which case the external 5V power supply supplied should be used. Refer to the RSKR8CLA8A User Manual for further details.

The Quick Start Guide provided with the Renesas Starter Kit board gives detailed instructions on how to connect the E8a to the host computer. The following assumes that the steps in the Quick Start Guide have been followed and the E8a drivers have been installed.

- Fit the Renesas LCD Panel to the connector marked ‘JA4’ on the RSK. Ensure all the pins of the connector are correctly inserted in the socket (See LCDAPPV2 Quick Start Guide for more information).
- Connect the E8a Debugger to a free USB port on your computer.
- Connect the E8a Debugger to the target hardware ensuring that it is plugged into the connector marked ‘E8a’.
- If supplying external power to the board, it can be turned on now.

5.3 Connecting to the Target with the E8a Debugger

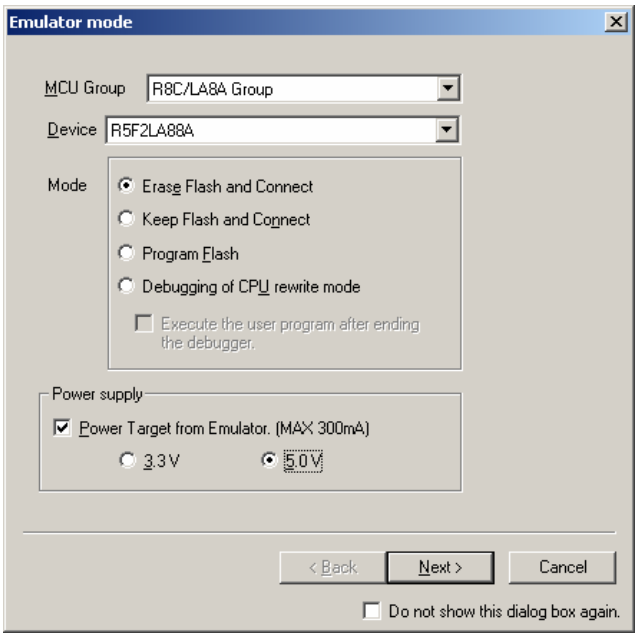
This section will take you through the process of connecting to the device, programming the Flash and executing the code.

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 will appear please choose the same options to connect.

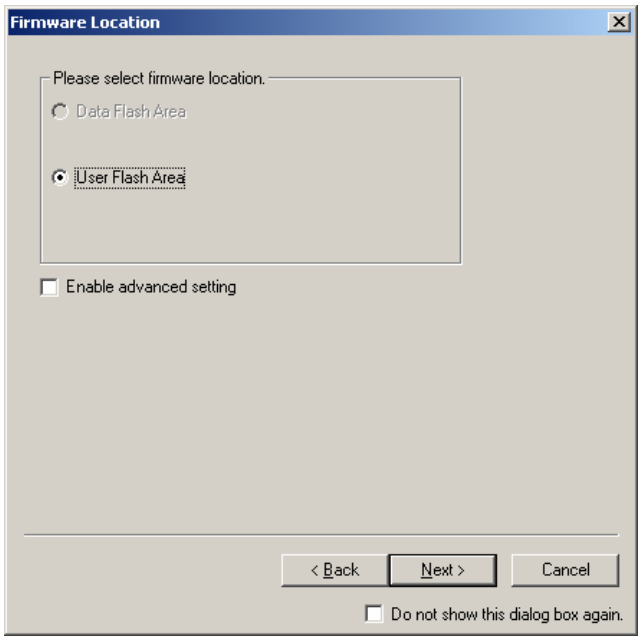
- Select the ‘SessionR8C_E8a_SYSTEM debug‘ session.
- Click the [Connect] button on the debug toolbar.

- The ‘Emulator Mode’ configuration dialog will appear. Ensure the following configurations are set:
  - MCU group: R8C/LA8A Group
  - Device: R5F2LA88A
  - Mode: Erase Flash and Connect
- If the E8a 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 RSKR8CLA8A User Manual for details).

Note: Powering from the E8a device will limit the maximum power consumption of the RSK, preventing some functionality. Refer to the RSKR8CLA8A User Manual for further details.

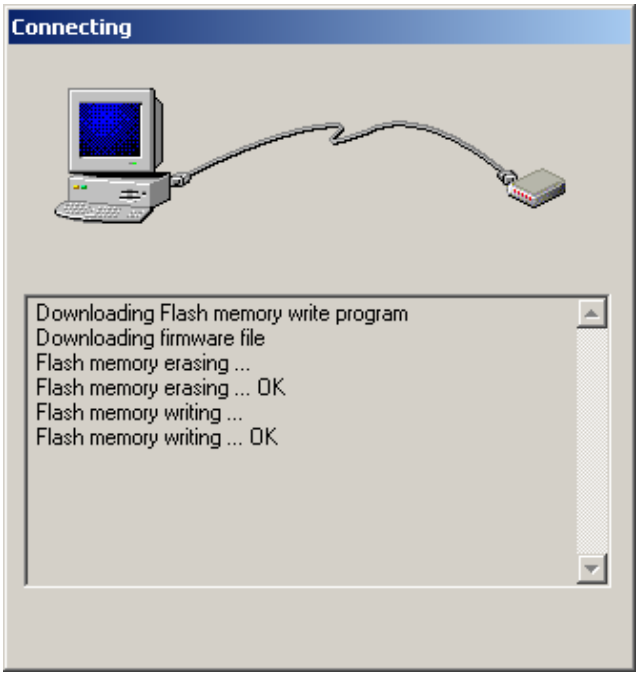


- Click [Next] button. ‘Firmware Location’ Window will appear.
- Choose “User Flash Area” in “Firmware Location” tab.



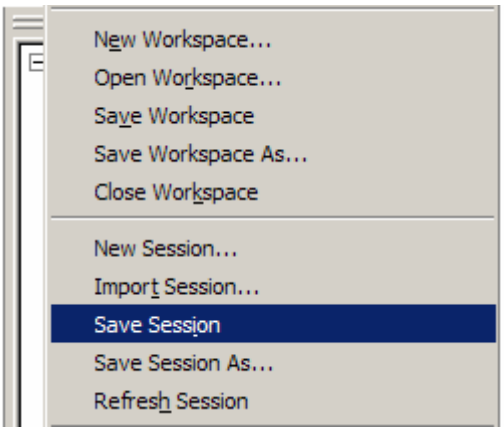
- Click [Next] button. ‘Communication Baud Rate’ window will open.
- Ensure the baud rate is set to 500000 bps. Once these settings have been confirmed, click the [Finish] button to continue.
- The Flash Memory write program will be downloaded to the target.

- A connecting dialog will appear, show the status of the connection process. Under default settings, this dialog box will disappear once the connection is complete.



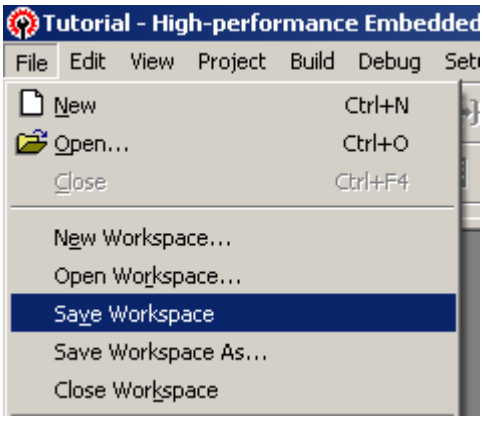
Now is a good time to save the High-performance Embedded Workshop session.

- Select ‘File’ | ‘Save Session’.



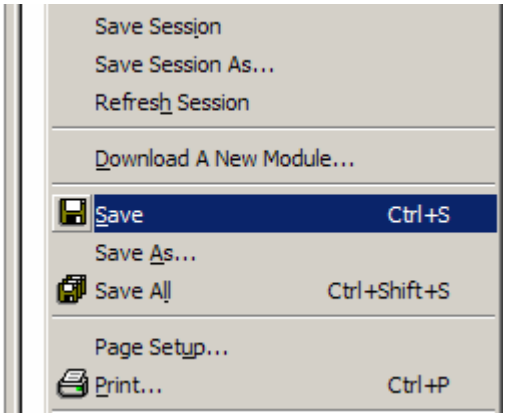
If you have changed any workspace settings now is a good time to save the workspace.

- Select ‘File’ | ‘Save Workspace’.



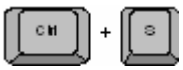
If you make any changes to files in HEW and want to preserve these change, you can save them by:

- 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:



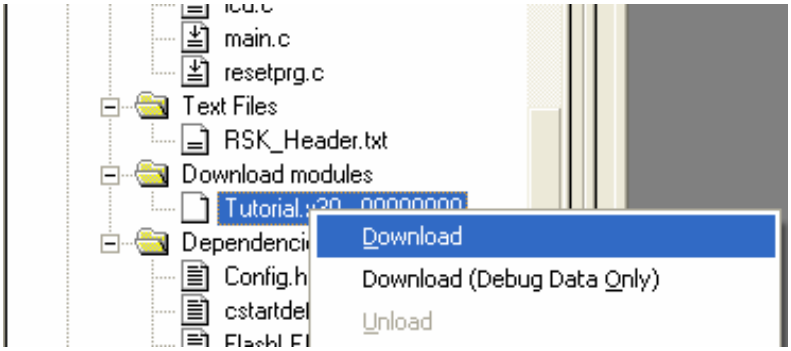
6. Downloading and Running the Tutorial

6.1 Downloading the Program Code

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

Now that you are 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 R8C/LA8A; press the ‘Reset CPU’ button on the Debug Tool Bar.

- 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.

```
 90                    void start(void)
 91  05CFC             {
 92                        /* Set interrupt stack pointer */
 93  05CFC                 _isp_   = &_istack_top;
 94                        /* Change protect mode register */
 95  05D00                 prcr    = 0x02U;
 96                        /* Set processor mode register */
 97  05D04                 pm0     = 0x00U;
 98                        /* Change protect mode register */
 99  05D07                 prcr    = 0x00U;
100                        /* Set flag register */
101  05D0A                 _flg_   = __F_value__;
102                    #if __STACKSIZE__!=0
103                        /* Set user stack pointer */
104  05D0E                 _sp_    = &_stack_top;
105                    #endif
106                        /* Setting 400H (Do not change) */
107  05D12                 _sb_    = 0x400U;
108                        /* Set variable vector's address */
109  05D16                 _intbh_ = 0x00U;
110  05D1A                 _asm("  ldc #(topof vector)&0FFFFh,INTBL");
111
112                        /* Initialize each sections */
113  05D1E                 initsct();
114
115                    #if __HEAPSIZE__ != 0
116                        /* Initialize heap */
117                        heap_init();
118                    #endif
119                    #if __STANDARD_IO__ != 0
120                        /* Initialize standard I/O */
121                        _init();
122                    #endif
123                        /* Initialize FB register for debugger */
124  05D22                 _fb_ = 0U;
125
126                        /* Call main() routine */
127  05D26                 main();
```

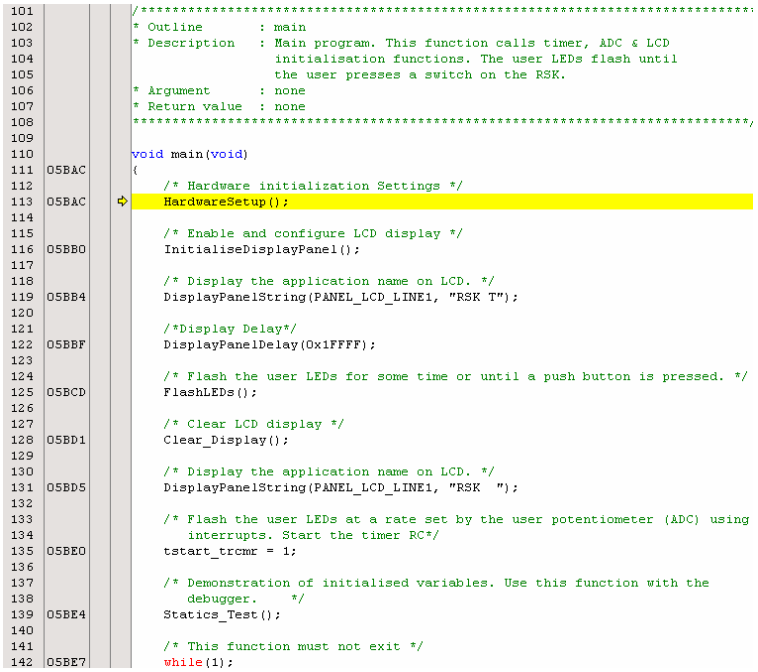
- Highlight the ‘main()’ function call by left clicking to the right of the text, and holding the left mouse button and dragging over to the left of it and releasing the left mouse button.
- Click the ‘Go to Cursor’ button to run the program up to this point.

- Click ‘Step In’ to open the main.c file and enter the main function.

7.2 Main Functions

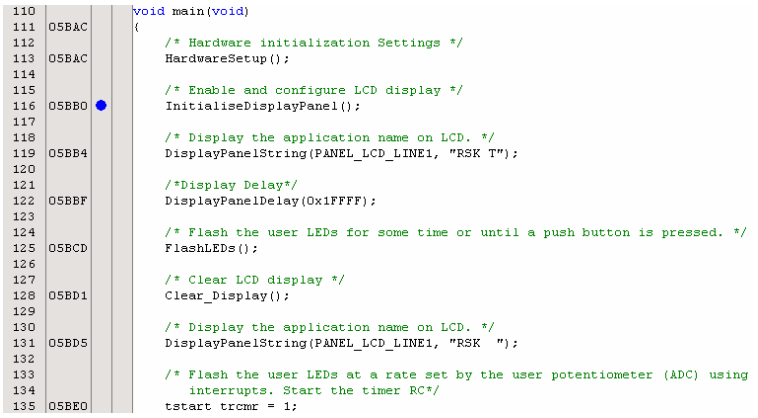
This section will look at the program code called from with the main() function, and how it works.

- The main.c file should already be open.



```
101                 /***********************************************************************
102                 * Outline       : main
103                 * Description   : Main program. This function calls timer, ADC & LCD
104                                   initialisation functions. The user LEDs flash until
105                                   the user presses a switch on the RSK.
106                 * Argument      : none
107                 * Return value  : none
108                 ***********************************************************************/
109
110                 void main(void)
111 05BAC           {
112                     /* Hardware initialization Settings */
113 05BAC             HardwareSetup();
114
115                     /* Enable and configure LCD display */
116 05BB0             InitialiseDisplayPanel();
117
118                     /* Display the application name on LCD. */
119 05BB4             DisplayPanelString(PANEL_LCD_LINE1, "RSK T");
120
121                     /*Display Delay*/
122 05BBF             DisplayPanelDelay(0x1FFFF);
123
124                     /* Flash the user LEDs for some time or until a push button is pressed. */
125 05BCD             FlashLEDs();
126
127                     /* Clear LCD display */
128 05BD1             Clear_Display();
129
130                     /* Display the application name on LCD. */
131 05BD5             DisplayPanelString(PANEL_LCD_LINE1, "RSK  ");
132
133                     /* Flash the user LEDs at a rate set by the user potentiometer (ADC) using
134                        interrupts. Start the timer RC*/
135 05BE0             tstart_trcmr = 1;
136
137                     /* Demonstration of initialised variables. Use this function with the
138                        debugger.    */
139 05BE4             Statics_Test();
140
141                     /* This function must not exit */
142 05BE7             while(1);
```

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



```
110                 void main(void)
111 05BAC           {
112                     /* Hardware initialization Settings */
113 05BAC             HardwareSetup();
114
115                     /* Enable and configure LCD display */
116 05BB0 ●           InitialiseDisplayPanel();
117
118                     /* Display the application name on LCD. */
119 05BB4             DisplayPanelString(PANEL_LCD_LINE1, "RSK T");
120
121                     /*Display Delay*/
122 05BBF             DisplayPanelDelay(0x1FFFF);
123
124                     /* Flash the user LEDs for some time or until a push button is pressed. */
125 05BCD             FlashLEDs();
126
127                     /* Clear LCD display */
128 05BD1             Clear_Display();
129
130                     /* Display the application name on LCD. */
131 05BD5             DisplayPanelString(PANEL_LCD_LINE1, "RSK  ");
132
133                     /* Flash the user LEDs at a rate set by the user potentiometer (ADC) using
134                        interrupts. Start the timer RC*/
135 05BE0             tstart_trcmr = 1;
```

The E8a 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 R8C Family E8a Emulator User's Manual.

- Press ‘Reset Go’ on the Debug Tool Bar.

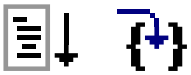
- The code will execute to the event point. At this point all the device initialisation will have been completed. The code window will open ‘main.c’ and show the new position of the program counter.

```
101                 /***********************************************************************
102                 * Outline       : main
103                 * Description   : Main program. This function calls timer, ADC & LCD
104                                   initialisation functions. The user LEDs flash until
105                                   the user presses a switch on the RSK.
106                 * Argument      : none
107                 * Return value  : none
108                 ***********************************************************************/
109
110                 void main(void)
111 05BAC           {
112                     /* Hardware initialization Settings */
113 05BAC             HardwareSetup();
114
115                     /* Enable and configure LCD display */
116 05BB0 ●  ➪         InitialiseDisplayPanel();
117
118                     /* Display the application name on LCD. */
119 05BB4             DisplayPanelString(PANEL_LCD_LINE1, "RSK T");
120
121                     /*Display Delay*/
122 05BBF             DisplayPanelDelay(0x1FFFF);
123
124                     /* Flash the user LEDs for some time or until a push button is pressed. */
125 05BCD             FlashLEDs();
126
127                     /* Clear LCD display */
128 05BD1             Clear_Display();
129
130                     /* Display the application name on LCD. */
131 05BD5             DisplayPanelString(PANEL_LCD_LINE1, "RSK  ");
132
133                     /* Flash the user LEDs at a rate set by the user potentiometer (ADC) using
134                        interrupts. Start the timer RC*/
135 05BE0             tstart_trcmr = 1;
136
137                     /* Demonstration of initialised variables. Use this function with the
138                        debugger.    */
139 05BE4             Statics_Test();
140
141                     /* This function must not exit */
142 05BE7             while(1);
```

- Insert event points on the FlashLEDs and Statics_Test function calls.

```
124                     /* Flash the user LEDs for some time or until a push button is pressed. */
125 05BCD ●           FlashLEDs();
126
127                     /* Clear LCD display */
128 05BD1             Clear_Display();
129
130                     /* Display the application name on LCD. */
131 05BD5             DisplayPanelString(PANEL_LCD_LINE1, "RSK  ");
132
133                     /* Flash the user LEDs at a rate set by the user potentiometer (ADC) using
134                        interrupts. Start the timer RC*/
135 05BE0 ●           tstart_trcmr = 1;
136
137                     /* Demonstration of initialised variables. Use this function with the
138                        debugger.    */
139 05BE4 ●           Statics_Test();
```

- Press ‘Go’ to run the program up to the event, then press ‘Step In’, to move the program counter to the beginning of the FlashLEDs() function definition.



- The ‘if’ statement checks to see if a switch has been pressed. The value of the ulLEDCount variable, which counts up with every LED flash is checked by the first ‘for loop’ statement. Once the ulLEDCount variable reaches 200 or a switch is pressed the function destroys.

```
73
74 0F438            void FlashLEDs(void)
75                  {
76                      unsigned long ulLed_Delay;
77                      unsigned long ulLEDcount;
78
79                      /* Flash the LEDs for 200 times or until a user switch is pressed */
80 0F43B ➪            for (ulLEDcount = 0; ulLEDcount < 200; ++ulLEDcount)
81 0F44F             {
82                          /* Check if key press is detected   */
83 0F44F                 if (gucKeyPressed > 0x00)
84                      {
85                          break;
86                      }
87
88 0F458                 for (ulLed_Delay = 0; ulLed_Delay < 40000; ++ulLed_Delay)
89                      {
90                          /* LED Flashing Delay */
91                      }
92
93                      /*  Toggles the LEDs after a specific delay. */
94 0F474                 ToggleLEDs();
95 0F47F              }
```

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

- Press F11 to step into the function.

- The Statics_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 ‘RSK T’ appears on the second LCD line, to be gradually replaced with the string ‘TEST’. The program then reverts the LCD back to the original message of ‘RSK T’.

```
154                 * Return value  : none
155                 ***********************************************************************************/
156
157                 void Statics_Test(void)
158 05BEC           {
159                     uint8_t ucCount;
160
161 05BEF             DisplayPanelString(PANEL_LCD_LINE1,ucStr);
162
163                     /*  At this point please right click on the 'ucStr' variable and select
164                     'Instant Watch'. A dialog will be displayed showing the current value
165                      of the variable. Select 'Add' in the dialog and a new 'Watch Window'
166                      will open. Step through the following code to see that the initialised
167                      data is being overwritten with the different data. */
168 05BFA               for (ucCount=0; ucCount<6; ucCount++)
169 05C01               {
170                         /* Put a breakpoint here and press F5 to step through the string
171                          overwrite sequence */
172 05C01                   ucStr[ucCount] = ucReplace[ucCount];
173
174                         /* Display the updated string on the LCD */
175 05C0B                   DisplayPanelString(PANEL_LCD_LINE1,ucStr);
176
177                         /* Delay */
178 05C16                   DisplayPanelDelay(0x2FFFF);
179                     }
180                     /* Fill a NULL character */
181 05C28               ucStr[ucCount] = '\0';
182
183                     /* RSK name is displayed on the LCD.    */
184 05C2E               DisplayPanelString(PANEL_LCD_LINE1, "RSK T");
185 05C39           }
186                 /***********************************************************************************
187                 End of function Statics_Test
188                 ***********************************************************************************/
```

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 R8C/LA8A series microcontrollers refer to the R8C/LA8A Group hardware manual.

For information about the R8C/LA8A assembly language, refer to the R8C/LA8A Series Software Manual.

Online technical support and information is available at: http://www.renesas.com/rskr8cla8a

Technical Contact Details
America: techsupport.america@renesas.com
Europe: tools.support.eu@renesas.com
Japan: csc@renesas.com

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

REVISION HISTORY	RSKR8CLA8A Tutorial Manual

Rev.	Date	Description	
		Page	Summary
1.00	Sep 24, 2010	—	First Edition issued
1.01	Jun 16, 2011	—	Company name was corrected.

Renesas Starter Kit Tutorial Manual

Publication Date: Rev.1.01 Jun 16, 2011

Published by: Renesas Electronics Corporation

Renesas Electronics Corporation

SALES OFFICES http://www.renesas.com

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: +44-1628-585-100, Fax: +44-1628-585-900
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.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670
Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001
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

Colophon 1.0

R8C/LA8A Group

Renesas Electronics Corporation

R20UT0285EG0101