User Manual

16

RL78/G14 Group

Renesas Starter Kit Tutorial Manual
For e²studio

RENESAS MCU
RL78 Family / G1X Series

Renesas Electronics
www.renesas.com

Rev.1.00 Jun 2012

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.

2. List of Abbreviations and Acronyms

Abbreviation	Full Form
ADC	Analogue to Digital Converter
API	Application Programming Interface
CD	Compact Disk
CPU	Central Processing Unit
E1	E1 Emulator
LCD	Liquid Crystal Display
LED	Light Emitting Diode
ROM	Read-Only Memory
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?

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 e^{2}studio, compiler toolchains or the E1 emulator. Please refer to the relevant user manuals for more in-depth information.

2.1 Note Regarding Source Code

Due to 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 effect the functionality of the sample code or the validity of this accompanying manual.

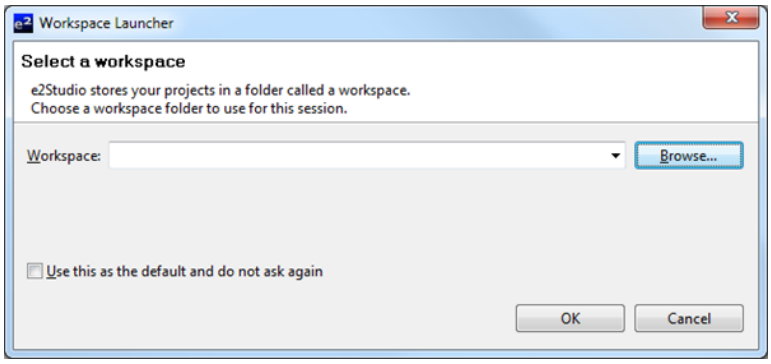
3. Project Workspace

3.1 Introduction

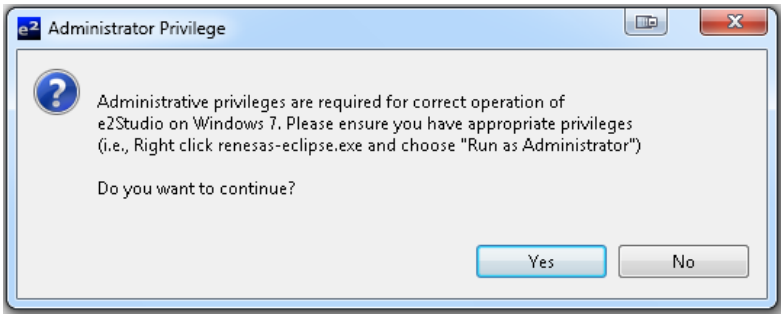
e^{2}studio is an open source integrated development tool that allows the user to write, compile, program and debug a software project on many of the Renesas Microcontrollers.

3.2 Starting e^{2}studio and Importing Sample Code

- Start e^{2}studio by selecting it from Start Menu. The first dialog to appear will be the Workspace Launcher.
- Click ‘Browse’ and select a suitable location to store your workspace, using the ‘Create New Folder’ option as necessary. Click ‘OK’.

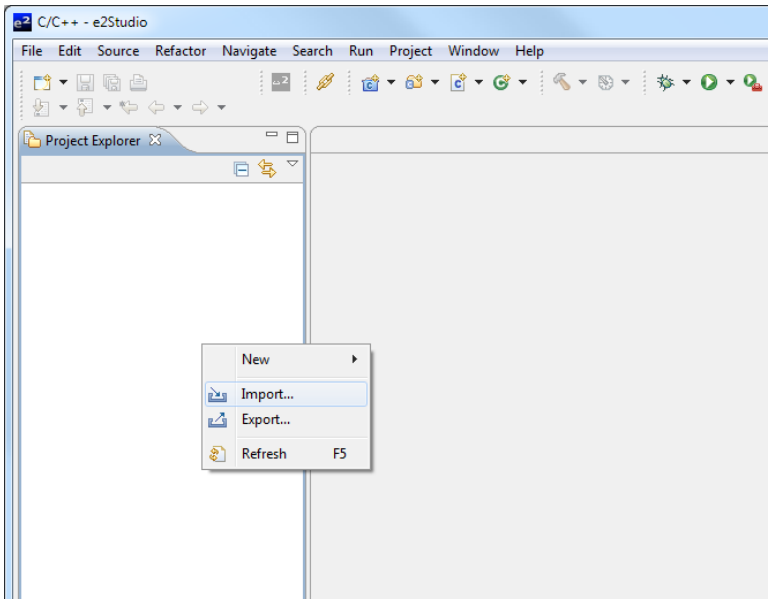


- Click ‘Yes’ when presented with the ‘Administrator Privilege’ dialog.

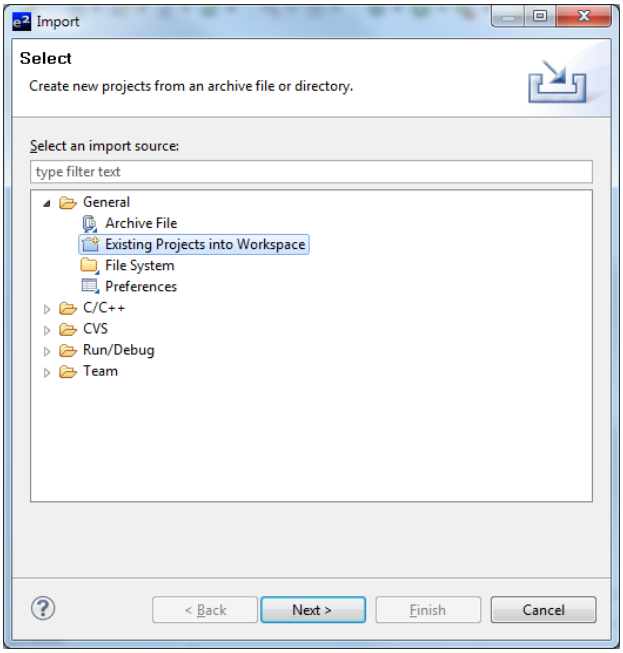


- The e2studio welcome splash screen will appear. Click the ‘Go to the workbench’ arrow button on the far right.

- Once the e^{2}studio environment has initialised, right click in the project explorer window and click ‘Import…’



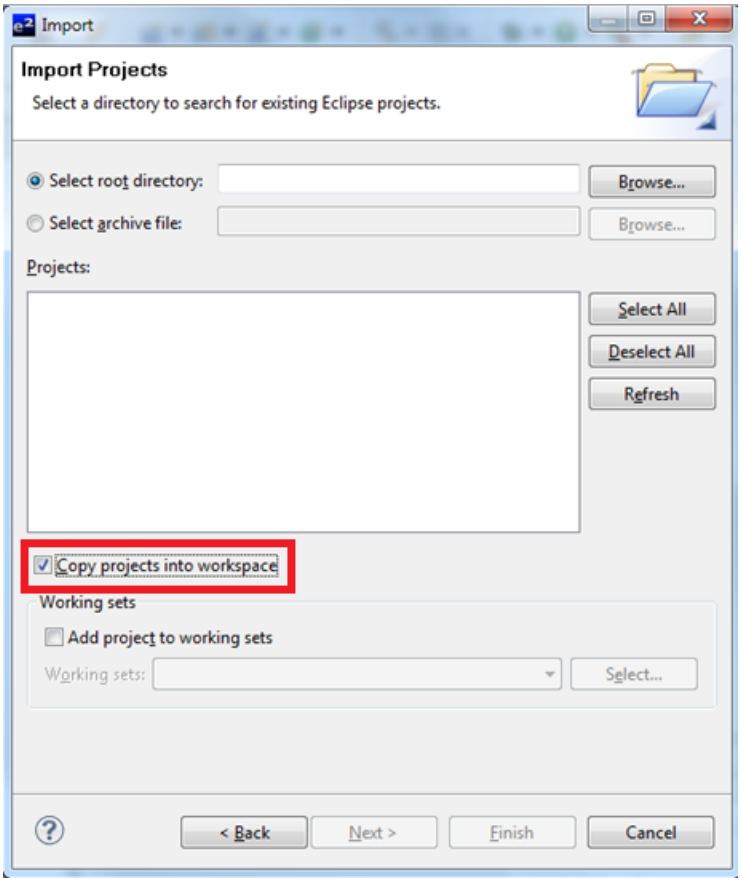
- The Import dialog will now show. Expand the ‘General’ folder icon, and select “Existing Projects into Workspace”, then click ‘Next’.



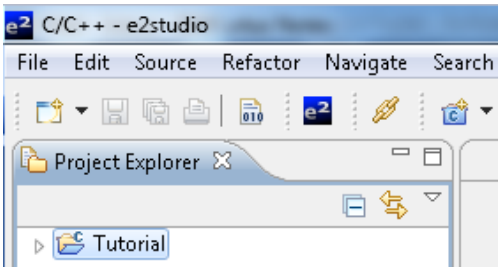
- The import dialog will now allow you to specify the project to import. Click the ‘Browse’ button, and locate the following directory:

  C:\Workspace\RSK\RSKRL78G14

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



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



3.3 Build Configurations and Debug Sessions

3.3.1 Build Configuration

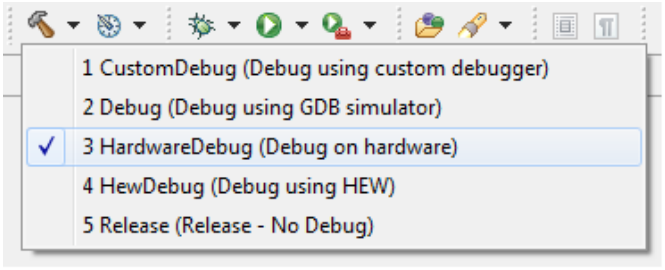
The e^{2}studio workspace will be created with several build configurations – the two we will address in this manual is 'HardwareDebug' and 'Release'.

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

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

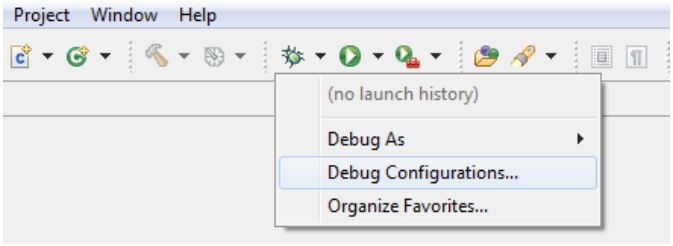
- Click the top level tutorial project folder again, and then the arrow next to the build button (hammer icon), and select the 'HardwareDebug' option.

- e^{2}studio will now build the code.

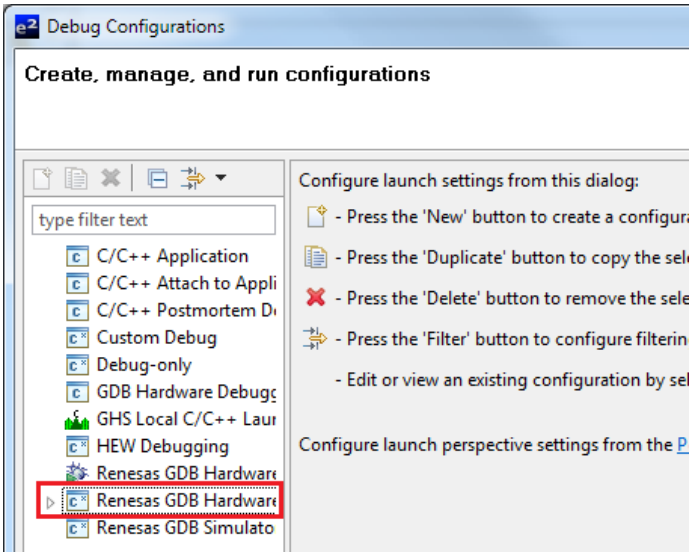


3.3.2 Debug Configuration

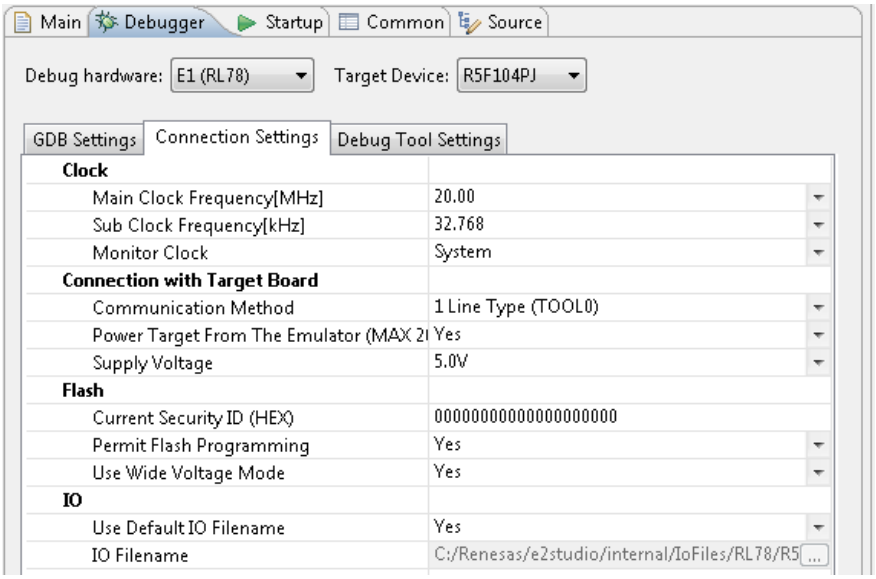
- Click the arrow next to the debug button (bug icon). Select ‘Debug Configurations’.



- The ‘Debug Configuration’ dialog will appear. Click the small arrow next to ‘Renesas GDB Hardware Debugging’ option.
- The build configurations for each project will appear. Select the entry for the tutorial project.

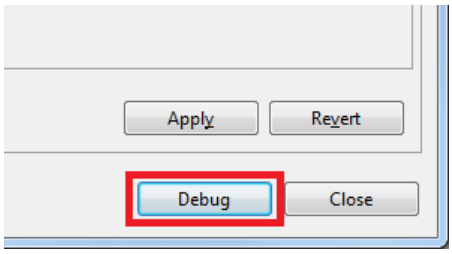


- The debug configurations control page will then show for the tutorial project. Change the main tab to ‘Debugger’, and then secondary tab to ‘Connection Settings’. Check through the debugger settings. If you intend to use an external power supply, set the ‘Power Target From The Emulator’ option to No (drop down menu).
- Refer to the RSK’s User Manual for details of power supply configurations.

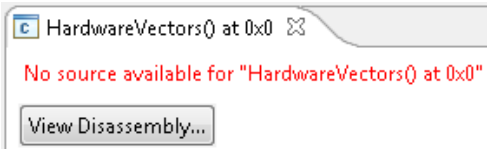


Note: e²studio will display a warning dialog if you attempted to connect with an incorrect power supply setting.

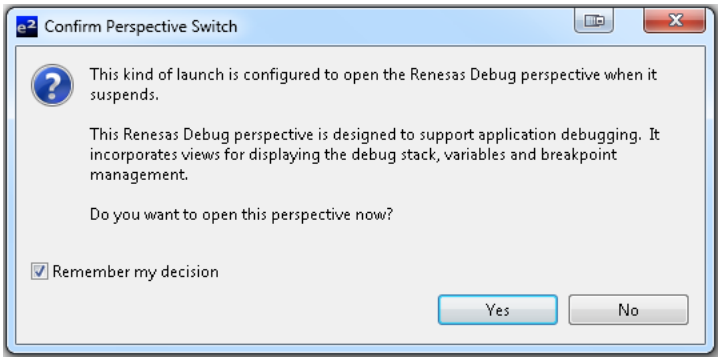
- Click the debug button to continue. e²studio will now connect to the debugger, and download the code to the target.



- After downloading the code. The File window will open the 'HardwareVectors() at 0x0'.



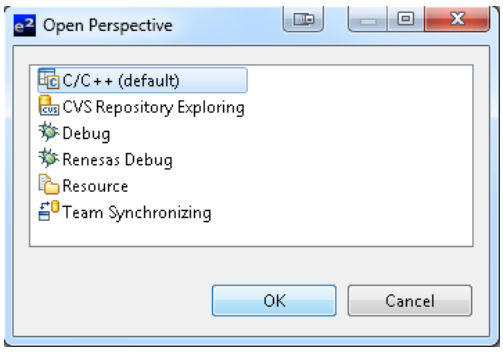
- e²studio may display a dialog, asking if you would like to switch to the 'Renesas Debug perspective'. Click 'Remember my decision' to prevent this dialog from appearing in future. Click 'Yes'.
- The new e²studio perspective layout is optimised for debugging.



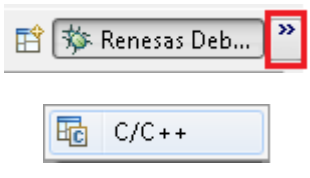
To change back to the default 'C/C++' perspective, from the menu bar select:

Window > Open Perspective > Other

The 'Open Perspective dialog will appear. Click on the desired perspective to select it then click 'OK'.



Alternatively, click on the button with the double arrow in the top right corner of the screen, as shown opposite, and select the 'C/C++' option that appears.



3.4 Running the Tutorial

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

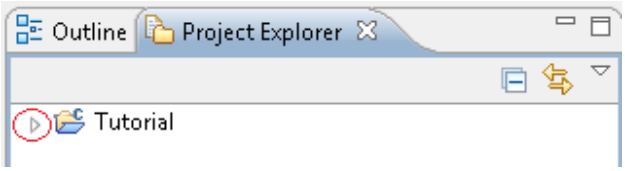
4. Basic Debugging of the Tutorial Program

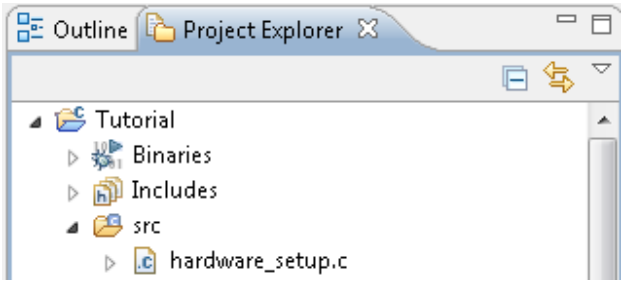
This section will look at basic debugging functionality in e^{2}studio.

4.1 Program Initialisation

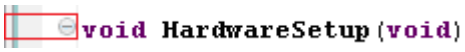
Before the main program can run, the microcontroller must be configured. The following parts of the tutorial program are used exclusively for initialising the RSK device so that the main function can execute correctly. The initialisation code is run every time the device is reset via the reset switch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

- After downloading the code, navigate to the Project Explorer window on the right hand side.
- Expand the 'Tutorial' folder by clicking on the arrow next to the folder icon, as highlighted by the red circle.
- Click the arrow next to the Tutorial project to expand the folder contents, and then click the arrow next to the 'src' folder to show the source files.
- Double click on 'hardware_setup.c' to open the file.

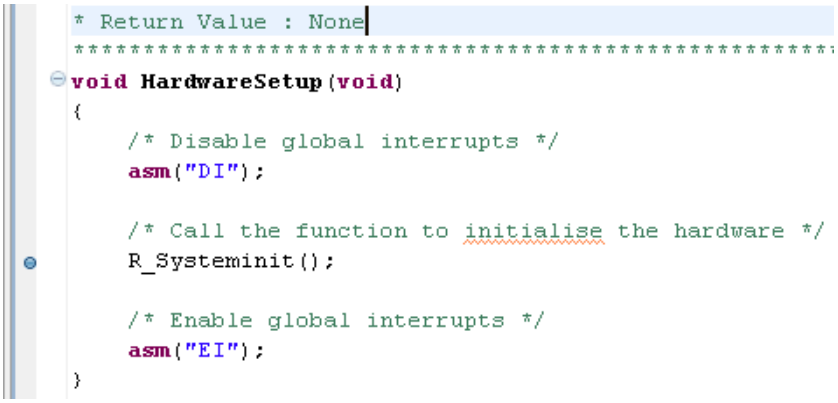




- Breakpoints can be set by double clicking within the column width space indicated by the red box. The column width is hereafter referred to as the breakpoint column.



- On the line with instruction R_Systeminit(), double click next to the vertical line to set a breakpoint.



Note:

The alternative to the above method requires reverting back to the default 'C/C++' perspective.

Whilst in the C/C++ perspective, set the mouse cursor on the instruction, then from the menu bar select Run > Toggle Breakpoint.

- Click the ‘Resume’ button (or press F8) to run the code up to this breakpoint.

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

```
* Return Value : None
*******************************************************
void HardwareSetup(void)
{
    /* Disable global interrupts */
    asm("DI");

    /* Call the function to initialise the hardware */
    R_Systeminit();

    /* Enable global interrupts */
    asm("EI");
}
```

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

- The ‘R_Systeminit’ function calls several initialisation functions which configure the MCU for normal operation. This includes input/output ports, and system clocks.
- The user can step through all the initialisation code by clicking the ‘Step Into’ icon and reading the code. For this guide, we will skip past it.
- Click the ‘Resume’ button, to run the code up to the main function.

HardwareVectors() at 0x0 | hardware_setup.c | *r_systeminit.c

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

/*****************************************************************
* Function Name: R_Systeminit
* Description  : This function initialises every macro.
* Arguments    : None
* Return Value : None
*****************************************************************
void R_Systeminit(void)
{
    PIOR0.pior0 = 0x00U;
    PIOR1.pior1 = 0x00U;
    R_CGC_Get_ResetSource();
    R_PORT_Create();
    R_CGC_Create();
    R_ADC_Create();
    R_TAU0_Create();
    R_INTC_Create();
    CRC0CTL.crc0ctl = 0x00U;
    IAWCTL.iawctl = 0x00U;
    PMS.pms = 0x00U;
}
```

For further details regarding hardware configuration, please refer to the RSKRL78G14 User’s Manual and the RL78/G14 Hardware Manual.

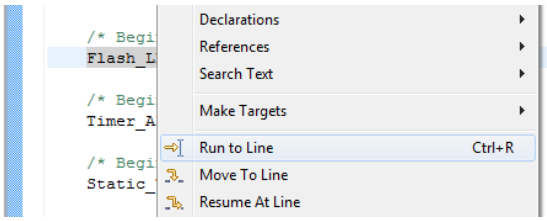
4.2 Main Functions

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

- The main function first initialises the debug LCD, and then displays ‘Renesas’ and the RSK name on the screen.
- Set a break point on the ‘Timer_ADC()’ function call by double clicking on the breakpoint column.
- 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 therefore not affected if the LCD display is attached or not.

```
* Return Value : None
*******************************************************************************
void main(void)
{
    /* Start user code. Do not edit comment generated here */

    /* Initialise the LCD module. */
    Init_LCD();

    /* Display information on the debug LCD. */
    Display_LCD(LCD_LINE1, "Renesas");
    Display_LCD(LCD_LINE2, NICKNAME);

    /* Flash the user LEDs for some time or until a push button is pressed. */
    Flash_LEDs();

    /* Flash the user LEDs at a rate set by the user potentiometer (ADC) using
    interrupts. */
    Timer_ADC();

    /* Demonstration of initialised variables. Use this function with the
    debugger.*/
    Statics_Test();

    /* Halt program in an infinite while loop */
    while(1U);

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

- Left click the function call ‘Flash_LEDs()’ to select it, then from the menu bar select the option ‘Run to Line’.
- This will cause the target to execute all code before this line. This can be observed by the text displayed on the debug LCD.



- Click the ‘Step Into’ button to step into the Flash_LEDs function.

- The program counter will now move to the Flash_LEDs function.
- Set a breakpoint on the ‘gSwitch_Flag = 0’ instruction line.
- The while loop iterates through and toggles the user LEDs. This loop will continue until it has flashed the LEDs 200 times or a user switch is pressed.
- Click ‘Resume’.

```
* Return value  : none
*******************************************************************************
void Flash_LEDs(void)
{
    /* Declare loop iteration count variable */
    uint32_t led_delay=0;
    uint8_t flash_count=0;

    /* Enable SW1 interrupts */
    R_INTC8_Start();

    /* Enable SW2 interrupts */
    R_INTC9_Start();

    /* Enable SW3 interrupts */
    R_INTC10_Start();

    /* Flash the LEDs for 200 times or until a user switch is pressed. */
    while((0 == gSwitch_Flag) && (flash_count++ < 200))
    {
        /* LED flashing Delay */
        for(led_delay=0; led_delay<400000; led_delay++)
        {
            /* Delay */
        }

        /* Toggles the LEDs after a specific delay. */
        Toggle_LEDs();
    }

    /* Reset the flag variable */
    gSwitch_Flag = 0;

    /* Disable SW1 interrupts */
    R_INTC8_Stop();

    /* Disable SW2 interrupts */
    R_INTC9_Stop();

    /* Disable SW3 interrupts */
    R_INTC10_Stop();
}
```

- Click the ‘Step Return’ button to resume program execution and exit the ‘Flash_LEDs’ function without stepping any remaining instructions.

```
    /* Flash the LEDs for 200 times or until a user switch is pressed. */
    while((0 == gSwitch_Flag) && (flash_count++ < 200))
    {
        /* LED flashing Delay */
        for(led_delay=0; led_delay<400000; led_delay++)
        {
            /* Delay */
        }

        /* Toggles the LEDs after a specific delay. */
        Toggle_LEDs();
    }

    /* Reset the flag variable */
    gSwitch_Flag = 0;

    /* Disable SW1 interrupts */
    R_INTC8_Stop();

    /* Disable SW2 interrupts */
    R_INTC9_Stop();

    /* Disable SW3 interrupts */
    R_INTC10_Stop();
}
```

- The program counter will halt at the ‘Timer_ADC’ function.
- This function initialises the ADC and timer units, so that a periodic interrupt toggles the user LEDs. The interrupt‘s period is varied using the ADC value set by the potentiometer.

```
* Return Value : None
*******************************************************************************
void main(void)
{
    /* Start user code. Do not edit comment generated here */

    /* Initialise the LCD module. */
    Init_LCD();

    /* Display information on the debug LCD. */
    Display_LCD(LCD_LINE1, "Renesas");
    Display_LCD(LCD_LINE2, NICKNAME);

    /* Flash the user LEDs for some time or until a push button is pressed. */
    Flash_LEDs();

    /* Flash the user LEDs at a rate set by the user potentiometer (ADC) using
    interrupts. */
    Timer_ADC();

    /* Demonstration of initialised variables. Use this function with the
    debugger.*/
    Statics_Test();

    /* Halt program in an infinite while loop */
    while(1U);

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

- Click the ‘Step Over’ button (or press F6) to step the program counter onto the next function, ‘Static_Test’.

- Click ‘Step Into’ to enter the ‘Static_Test’ function.

- In the *for* loop, the contents of the string ‘ucStr’ are replaced with the contents of ‘ucReplace’, one element at a time.
- While this happens, the code is regularly interrupted by the Timer ADC code, allowing both to appear to run simultaneously.

```
* Return value : none
*******************************************************************************
void Statics_Test(void)
{
    /* Declare loop count variable */
    uint8_t uiCount;
    /* Declare string variable to hold the string to be copied */
    uint8_t ucStr[] = "STATIC \0";
    /* Declare variable buffer to store the copied string */
    uint8_t ucReplace[] = "TESTTEST\0";

    /* Declare a delay count variable */
    uint32_t ulDelay;

    /* Write ucStr variable, "STATIC" to LCD */
    Display_LCD(LCD_LINE2, (int8_t*)ucStr);

    /* Begin for loop which writes one letter of ucReplace to the LCD at a time
       The nested while loops generate the delay between each letter change */
    for (uiCount=0; uiCount<8; uiCount++)
    {
        /* Replace letter number uiCount of ucStr from ucReplace */
        ucStr[uiCount] = ucReplace[uiCount];

        /* Display the character on the debug LCD */
        Display_LCD(LCD_LINE2, (int8_t*)ucStr);

        /* LED Flashing Delay */
        for(ulDelay=0; ulDelay<700000; ulDelay++)
        {
            /* Delay */
        }
    }

    /* Clear LCD Display */
    ucStr[uiCount] = '\0';

    /* Write MCU nickname to LCD again */
    Display_LCD(LCD_LINE2, NICKNAME);
}
```

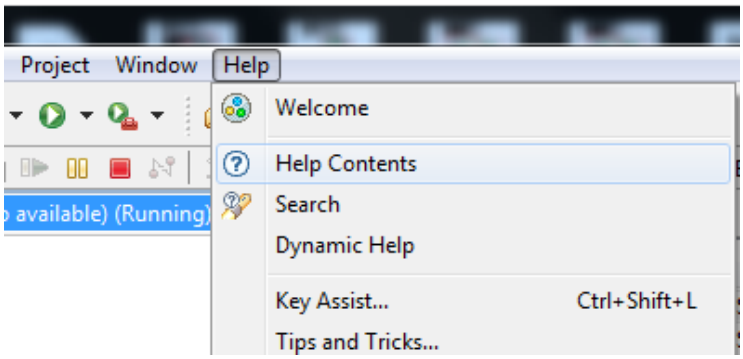
- Click the resume button to run the code through.

- You can observe the user LED flash rate vary as you adjust the potentiometer, at the same time as the second line of the debug LCD string changes character by character from ‘STATIC’ to ‘TESTTEST’.
- The second line of the debug LCD will then revert back to display the RSK name.
- This is the end of the tutorial code.

5. Additional Information

Technical Support

For details on how to use e^{2}studio, refer to the help file by opening e^{2}studio and clicking ‘Help’ and selecting ‘Help Contents’



Parts of the sample code provided with the RSKRL78G14 can be reproduced using the ‘Applilet3 for RL78_G14’ code generator tool. Applilet can be downloaded from the Renesas website.
Source files and functions generated by Applilet are prefixed with ‘r_’ and ‘R_’, respectively.

For information about the RL78/G14 microcontrollers refer to the RL78/G14 Group hardware manual.

For information about the RL78/G14 assembly language, refer to the RL78 Family’s Software Manual.

Online technical support and information is available at: http://www.renesas.com/rskrl78g14

Technical Contact Details
America: techsupport.america@renesas.com
Europe: software_support-eu@lm.renesas.com
Japan: csc@renesas.com
GNURL78 Compiler Support: www.kpitgnutools.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	RSKRL78G14 Tutorial Manual

Rev.	Date	Description	
		Page	Summary
1.00	Jun 21, 2012	¾	First Edition issued

Renesas Starter Kit Tutorial Manual

Publication Date: Rev. 1.00 Jun 21, 2012

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: +44-1628-585-100, 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

RL78/G14 Group

Renesas Electronics Corporation

R20UT0790EG0100