IO Wizard

Tutorial using RSKR8C27

Introduction

For this tutorial, we will be using IO Wizard together with Renesas Starter Kit (RSKR8C27) for R8C/27.

Please refer to Renesas Technology Website for more details on the general functionality of IO Wizard.

Target Device

IO Wizard supports selected devices in the R8C/1x Series and R8C/3x Series, all devices in the R8C/2x Series, M16C/Tiny Series and H8/38347.

Contents

1. Introduction

This tutorial provides user a step-by-step guide to using IO Wizard to generate initialization code for R8C/27 MCU peripherals. We will be using IO Wizard together with a Renesas Starter Kit (RSKR8C27) for R8C/27, E8a on-chip debugging emulator and High-performance Embedded Workshop (HEW) software.

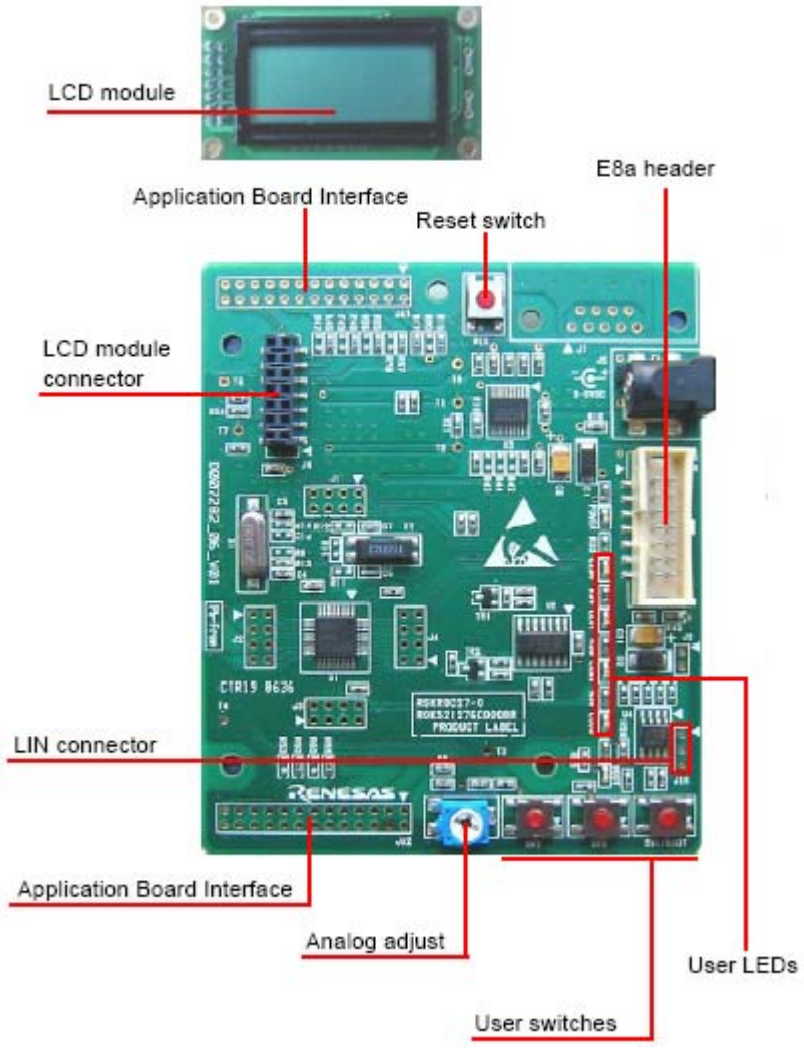

Figure 1 RSKR8C27 board

In this exercise, we will be using Timer RA interrupt to toggle (turn on/off) 4 user LEDs mounted on the RSKR8C27 board to achieve the blinking effect.

R8C/27 (RSKR8C27 board)	
User LED	**Pin**
LED0 (Green)	P0_0
LED1 (Orange)	P0_1
LED2 (Red)	P0_2
LED3 (Red)	P0_3

We will walk through the steps of using IO Wizard to generate initialization code for Timer RA and Port 0.

Note: Timer RA is used to generate the time base for the blink frequency. The 4 user LEDs are connected to Port 0.

By the end of this session, you will have gained the know-how on using IO Wizard to generate initialization routines, how to generate your source files into HEW, how to save the settings made and load the saved settings.

2. Using IO Wizard to generate initialization routines

a) Create new HEW project

We have to setup a new project called LED_blink in the High-performance Embedded Workshop (HEW) integrated development environment to further edit the code and to prepare it for downloading to the target board.

1) Select “New Project Workspace” from the File drop down menu.

2) In the New Project Workspace dialog window, Select CPU family “M16C” and Tool chain “Renesas M16C Standard”. Select “E8A_RSKR8C27” under Project Types. Enter "LED_blink” into the Workspace Name field. The Project Name and Directory fields will be filled automatically. This will generate a C:\Workspace\LED_blink directory that will hold all of our project files. Click <OK> to continue.

3) In E8A_RSKR8C27 - Step 1 dialog window, select “Application”. This will create an empty application project with correct setup of the memory map, interrupt vector table and debug target for the RSK. Click <Next> to continue.

4) In E8A_RSKR8C27 - Step 2 dialog window, click <Finish>. Click <OK> in the Project Generator Information window.

b) Using IO Wizard

To invoke IO Wizard, go start → All Programs → IO Wizard → IO Wizard.exe
Choose MCU and configure the MCU and clock settings. Select the following:

Series:	R8C/Tiny Series
Group:	R8C/27 Group
Part No.:	R5F21276SNFP
CPU Clock:	Main Clock (Xin-Xout)
Clock Frequency:	20MHz

→ Click on <Next> to proceed to the next screen.

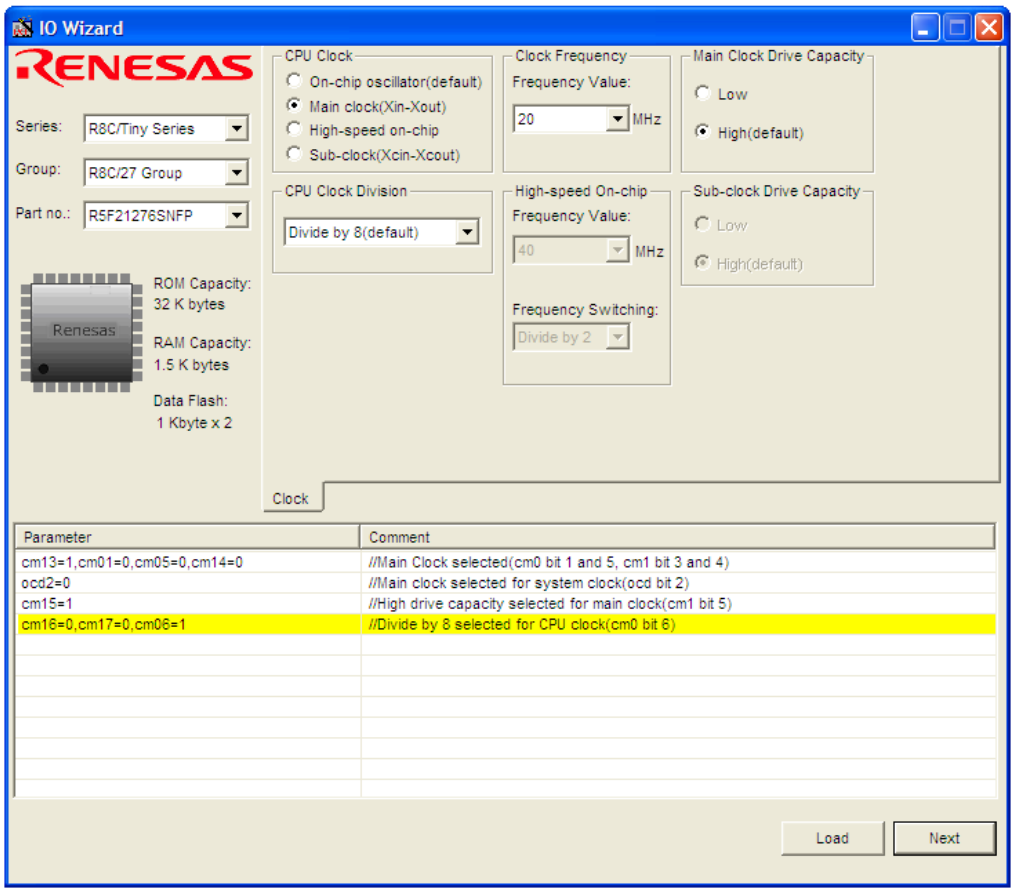

Figure 2 Configure MCU and clock settings

Port 2

Next, we have to configure Port 0 as the 4 user LEDs are connected to I/O port 0 pin 0-3.

Select Port 0 by clicking on Port 0 from the peripheral selection list. All pins of port 0 are set to input by default.

→ Configure pins 0,1,2 and 3 as outputs to drive the LEDs.

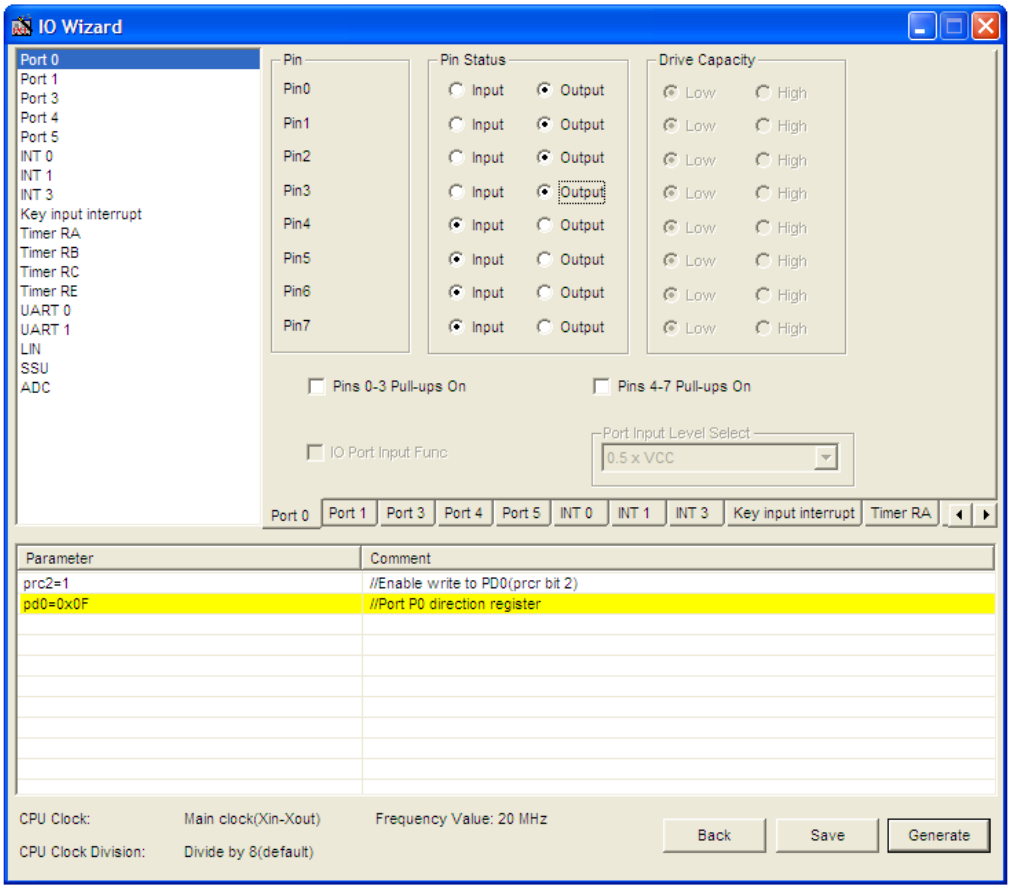

Figure 3 Configure Port 0 settings

Timer RA

Subsequently, we have to configure Timer RA as we are using Timer RA to generate the 100ms time base for the blink frequency (LED switch on and off every 100 ms)

→ Click activate radiobutton to enable usage of Timer RA.

→ Leave the Operation Mode setting as “Timer”

→ Internal Count Source as “No division”.

→ Select 2ms from Timer Value.

→ Set interrupt priority to level 2.

Timer RA will be initialized to work in timer mode, underflowing every 2ms and thereby generating a 2ms time base

IO Wizard has calculated a prescaler value of 249 and a count register value of 159.

(249+1)(159+1)÷20MHz=2ms

Generate a software counter later on to get the desired 100ms time base.

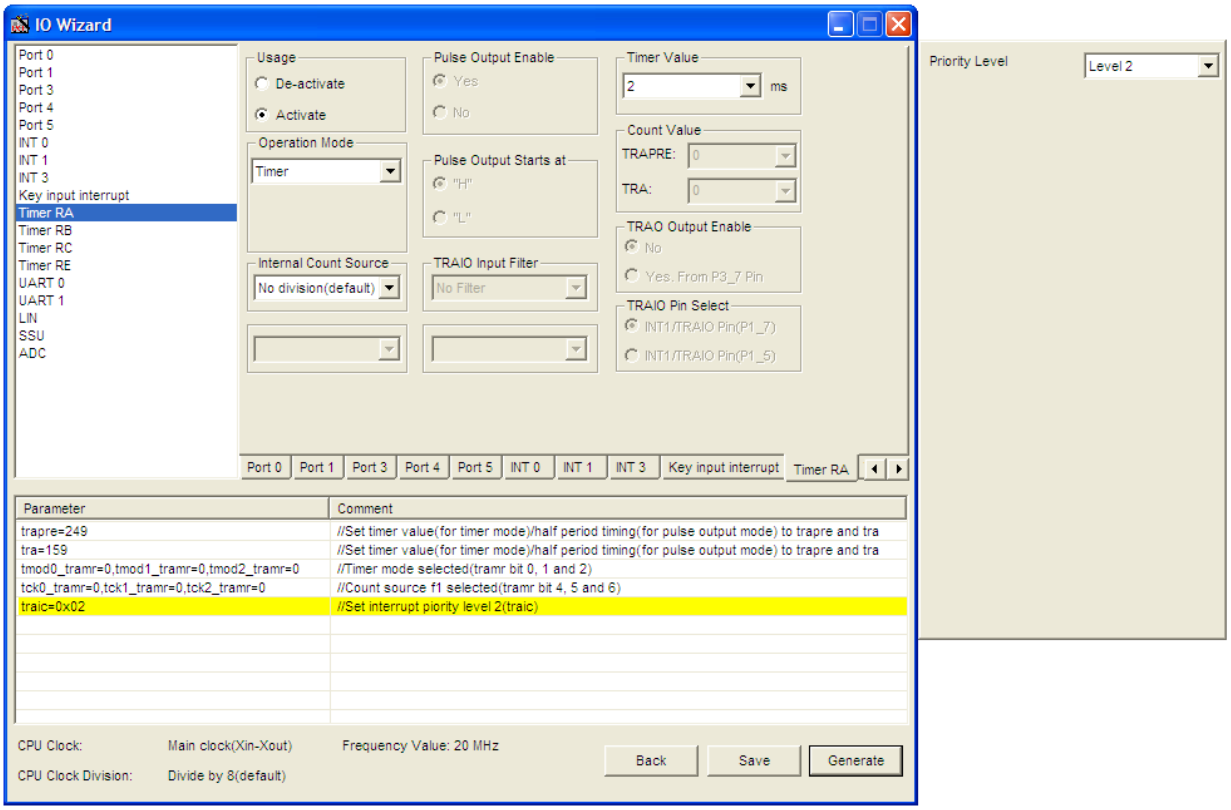

Figure 4 Configure Timer RA settings

→ Click <Save> button to save all current settings made. This includes the MCU and clock settings made in the Main Window.

A "Save As" dialog as shown on the next page, will pop out to prompt user to save their settings made into a file (filename.iow) for loading in future sessions.

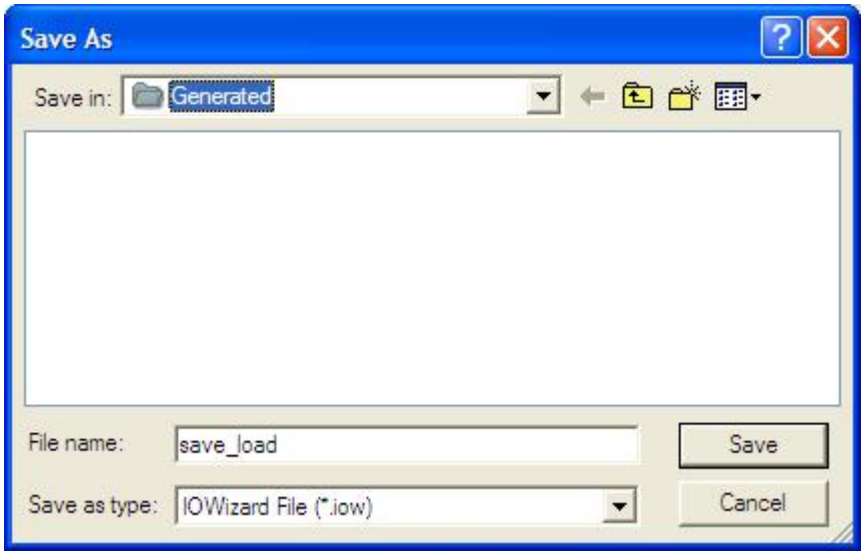

Figure 5 Save As dialog

After the settings are being saved, we will return to the main window. Remember all peripheral settings will be reset to default when we execute this step.

A "Load" button is added beside the "Next" button in the main window, it allows user to load any settings made in prior sessions.

→ Click the <Load> button, a "Open" dialog as shown below, will pop out to prompt user to open the saved file.

Once the selected "filename.iow" file is opened, all previous settings saved will be updated onto the screen.

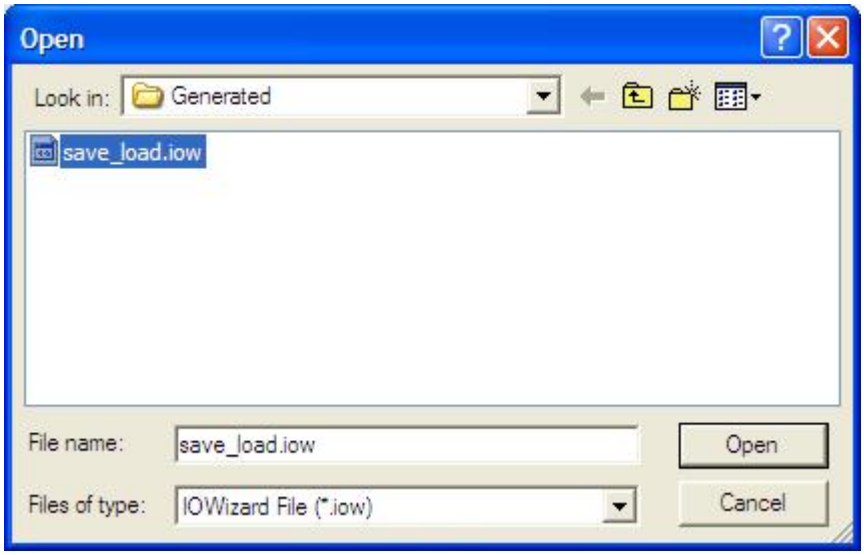

Figure 6 Open dialog

→ Return to the peripheral window, click on “Generate” button.

A “Generate” dialog as shown below will pop out to prompt user to save the generated files to their desired folder directory or insert the files into an existing open HEW workspace.

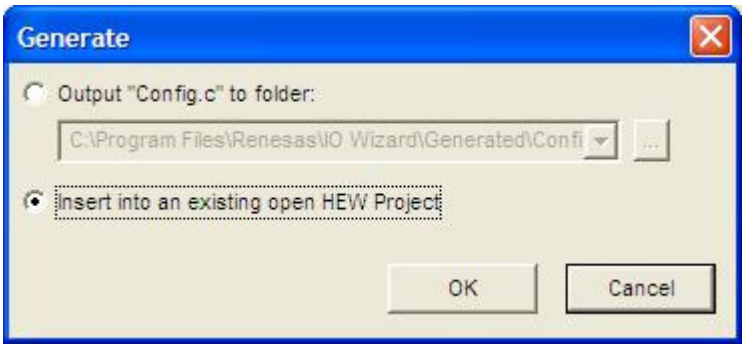

Figure 7 Generate dialog

The Link to HEW option allows user to export the generated source codes into HEW.

The prerequisite for this option to work:

1) HEW 4.02 (or above)
2) HEW Target Server (COM) has already been installed.
3) A HEW Workspace has to be opened.

Below are the steps required to install HEW Target Server and get it running:

1. You will need HEW 4.02(or above) installed on your machine.
2. Preparing to use the HEW Target Server (COM)

Add HEWTargetServer.exe to your computers registry

→ Using Windows Explorer, surf to the directory where HEW was installed (in most cases, this is C:\Program Files\Renesas\HEW\).

→ Double click on **REGISTERSERVER.bat** file

- HewTargetServer.exe will be registered in your computers registry.
- To remove HewTargetServer.exe from the registry, double-click on the UNREGISTERSERVER.bat file located in the same folder.

Register the EcxHewTargetServer.dll within HEW

→ Launch High-performance Embedded Workshop

- From the **Start Menu**, click on All Programs -> Renesas -> High-Performance Embedded Workshop -> High-performance Embedded Workshop

→ When the "Welcome" dialog box appears:

- Click the <**Administration**> button.

→ When the Tool Administration dialog box appears:

- Click on the <**Register**…> button

→ When the "Select HEW Registration File" Window appears, surf to: : the folder in which the High-performance Embedded Workshop, or Renesas integrated development environment, is installed i.e. C:\Program\Files\Renesas\HEW\System\SEC\HewTargetServer

- Double click on "**EcxHewTargetServer.hrf**"

Registration of EcxHewTargetServer.dll has completed.

→ Click <OK> to have IO Wizard generate the initialization code for MCU selection, clock settings, Port 2 and Timer RA to an opened workspace.

HEW Projected detected dialog box appeared.

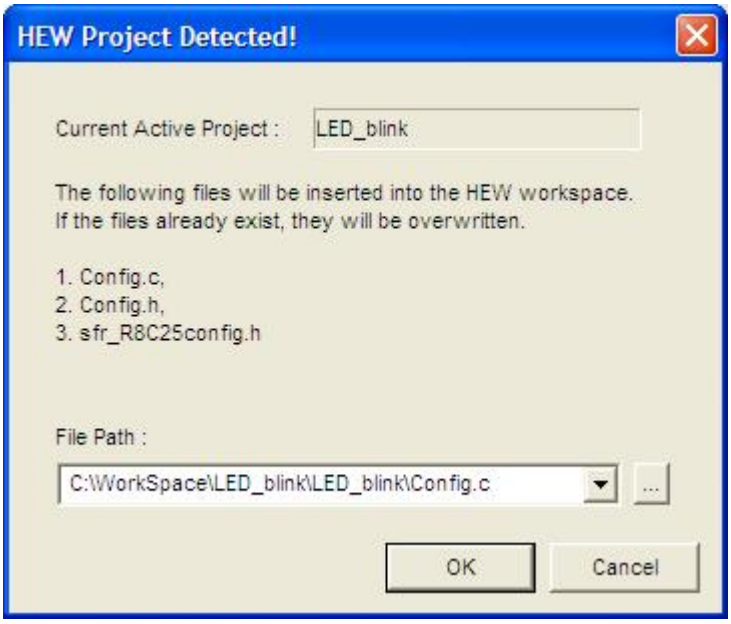

Figure 8 HEW Project Detected! dialog

→ Click <OK> to insert the generated files to the existing open "LED_blink" project workspace.

At this point, 3 files will be created and inserted to the workspace.

Filename	Description
Config.c	Contain the main routine and initialization settings
Config.h	Pragma directives and function declarations
Sfr_R8C27config.h	Declare the SFR addresses.

The 3 generated files are inserted into LED_blink HEW workspace as shown in below figure.

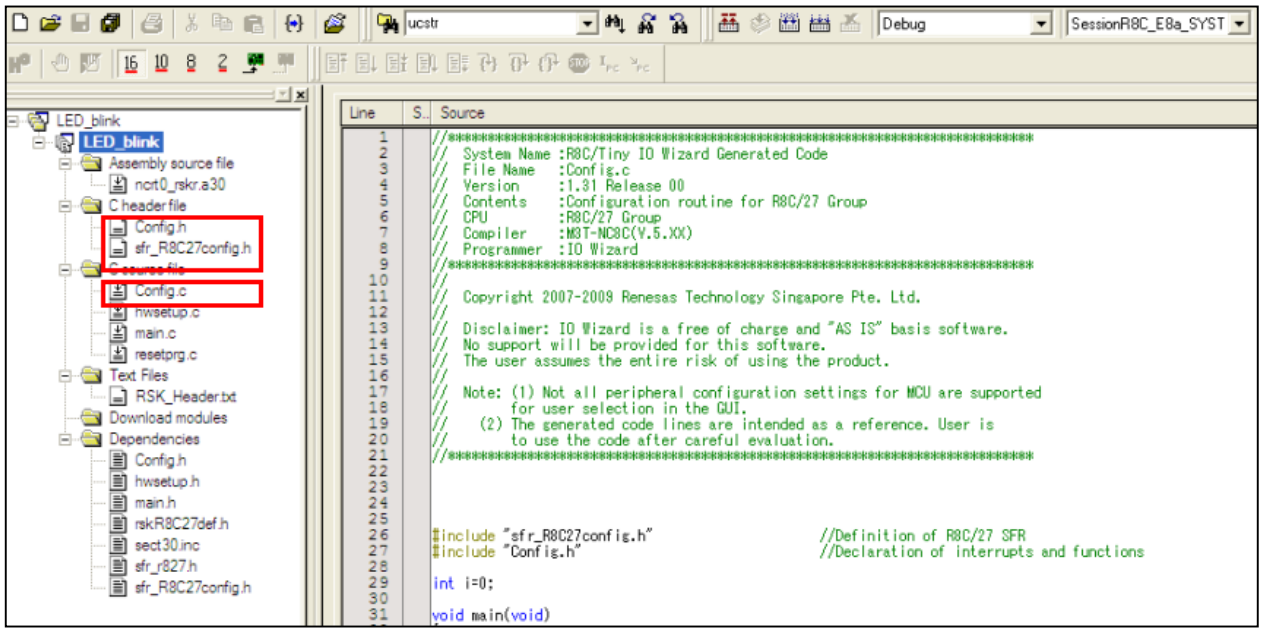

Figure 9 Generated files inserted to LED_blink HEW workspace

→ Remove main.c from the project workspace.

→ In config.c file, scroll down to the "Timer RA Setting" section and remove the double slashes (//) to enable tstart_tracr=1.

→ Add software counter by cutting and pasting code that have been that is being bold (as shown below) into the empty Timer RA interrupt service routine.

```
void _timer_ra(void)
{
      //Enter your timer RA routine
      //Static Variables (retains values between interrupts )
      static char counter = 0; // delay counter initial value
      if(++counter >= 50) //blink LEDs every 100ms
      {
            counter = 0; // return to initial state
            p0=p0^0x0F;  // invert LED value (pins 0..3)
      }
}
```

The Timer RA interrupt service routine will generate the 100ms period for the blinking of the LEDs.

Remember that in IO Wizard we selected a 2ms time base for counter RA. This means that with Timer RA interrupt enabled, we will get a timer interrupt every 2ms. The line if(++counter >= 50) in the Timer RA interrupt service routine generates a loop that is executed 50 times, giving a time period of 50×2ms=100ms before toggling the four user LEDs.

→ Finally, the last step is to define the interrupt vector for the Timer RA interrupt service routine. The vectors are defined in the file sect30.inc.

In the R8C/Tiny section, scroll then down to the “variable vector” section and replace the line as shown below.

```
.lword        dummy_int        ; vector 22

with

.glb          __timer_ra
.lword        __timer_ra       ; vector 22
```

Make sure there are no errors when you compile the project files.

→ In HEW, select “SessionR8C_E8a_SYSTEM” from the Session pull-down menu and click the connect button to establish a connection between your PC and the E8a on chip debugger.

E8a debugger module

→ In the Emulator Mode dialog window, select “R5F21276” from the Device pull-down menu.

→ Select Mode “Erase Flash and Connect”. Check “Power Target from E8a” and select “5.0V”. Click <Next>.

→ In the Firmware Location window, select “User Flash Area” and click <Next>.

HEW should now connect to the target. Right-click on Download Modules in HEW’s File List window and select “Download all modules” from the pop-up window

→ In HEW, click the Reset-Go button to execute the downloaded code. The board’s four LEDs will flash every 100ms. Click the Stop button to end code execution.

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	June.16.09	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.