RX Family

Creating Workspace with RI600/4

R20AN0091ES0100
Rev.1.00
Oct 01, 2010

Introduction

Target Device

Applicable MCU: RX Family

Contents

1. Guide in using this Document

This document aims to equip users with the know-how of creating workspace with RI600/4.

Table 1 Explanation of Document Topics

Topic	Objective	Pre-requisite
Preparing the Software	Describe the installation of RI600/4	None
Opening a Workspace	Guides users in working with RI600/4 workspace	Knowledge in High-performance Embedded Workshop
Running the Workspace	Guide users in the execution of RI600/4 workspace with E1 emulator	Knowledge in High-performance Embedded Workshop and E1 Emulator
Reference Documents	Listing of documents that equip users with knowledge in the pre-requisite requirements	None

2. Preparing the Software

RI600/4 is a real-time operation system (RTOS) product developed for the RX Family RX600 Series target devices. To be able to create a workspace with RI600/4, users are required to install itron package: RI600/4 V1.00. Prior to its installation, it is necessary to ensure Renesas High-Performance Embedded Workshop (HEW), C/C++ Compiler package for RX family and E1/E20 Emulator Debugger package have been installed. Figure 1 illustrates the installation sequences.

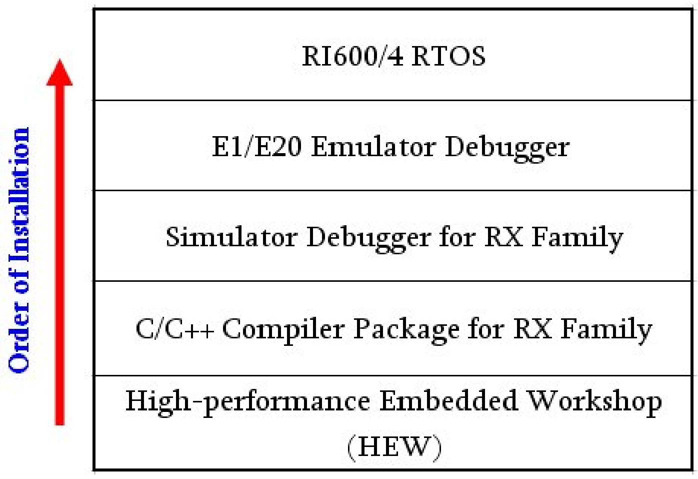
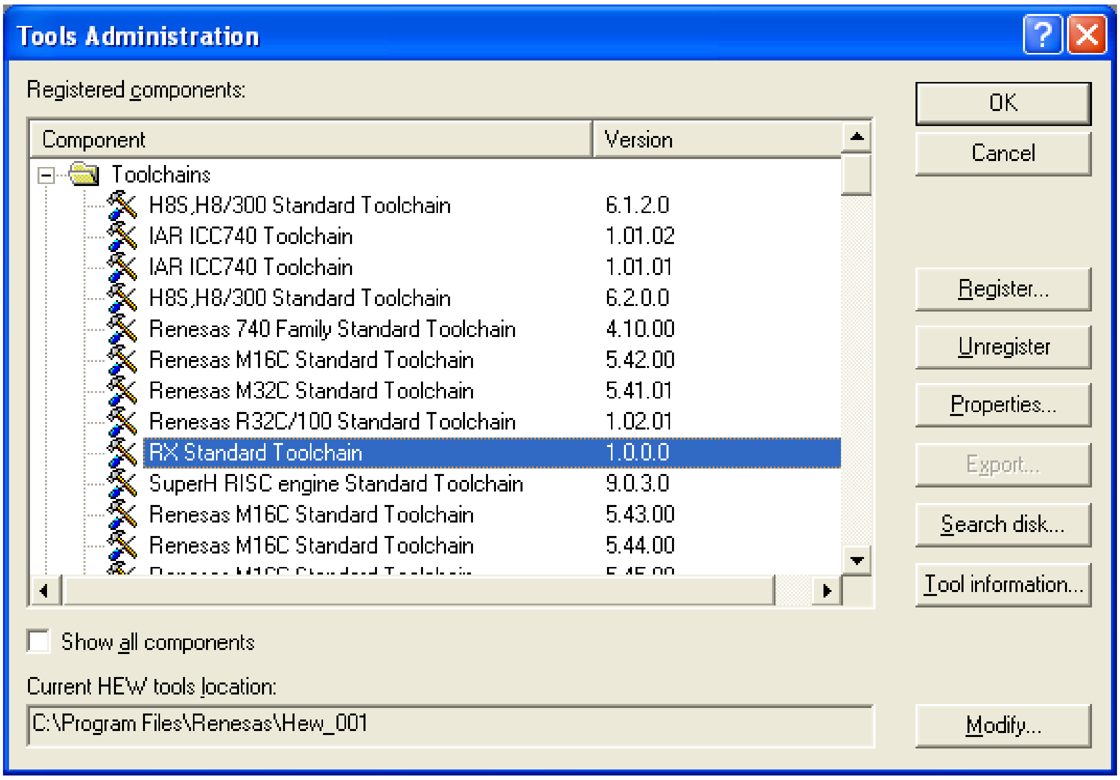

Order of Installation
RI600/4 RTOS
E1/E20 Emulator Debugger
Simulator Debugger for RX Family
C/C++ Compiler Package for RX Family
High-performance Embedded Workshop (HEW)

Figure 1 Installation Sequences

3. Installing RI600/4

To verify that “C/C++ Compiler Package for RX Family” and “E1/E20 Emulator Debugger Package” has been installed, refer to the “Tools Administration” option of HEW (Figure 2).



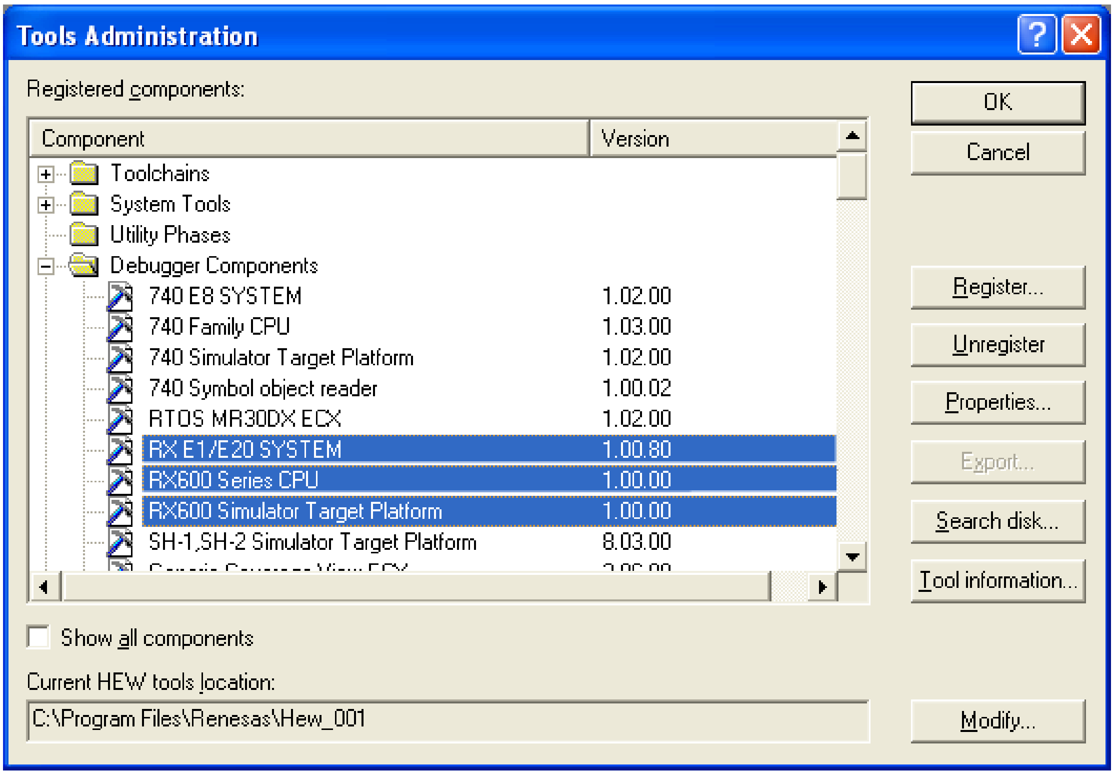
Registered components:

Component	Version
Toolchains	
H8S,H8/300 Standard Toolchain	6.1.2.0
IAR ICC740 Toolchain	1.01.02
IAR ICC740 Toolchain	1.01.01
H8S,H8/300 Standard Toolchain	6.2.0.0
Renesas 740 Family Standard Toolchain	4.10.00
Renesas M16C Standard Toolchain	5.42.00
Renesas M32C Standard Toolchain	5.41.01
Renesas R32C/100 Standard Toolchain	1.02.01
RX Standard Toolchain	1.0.0.0
SuperH RISC engine Standard Toolchain	9.0.3.0
Renesas M16C Standard Toolchain	5.43.00
Renesas M16C Standard Toolchain	5.44.00
Renesas M16C Standard Toolchain	5.45.00

□ Show all components

Current HEW tools location:

C:\Program Files\Renesas\Hew_001



Registered components:

Component	Version
Toolchains	
System Tools	
Utility Phases	
Debugger Components	
740 E8 SYSTEM	1.02.00
740 Family CPU	1.03.00
740 Simulator Target Platform	1.02.00
740 Symbol object reader	1.00.02
RTOS MR30DX ECX	1.02.00
RX E1/E20 SYSTEM	1.00.80
RX600 Series CPU	1.00.00
RX600 Simulator Target Platform	1.00.00
SH-1,SH-2 Simulator Target Platform	8.03.00

□ Show all components

Current HEW tools location:

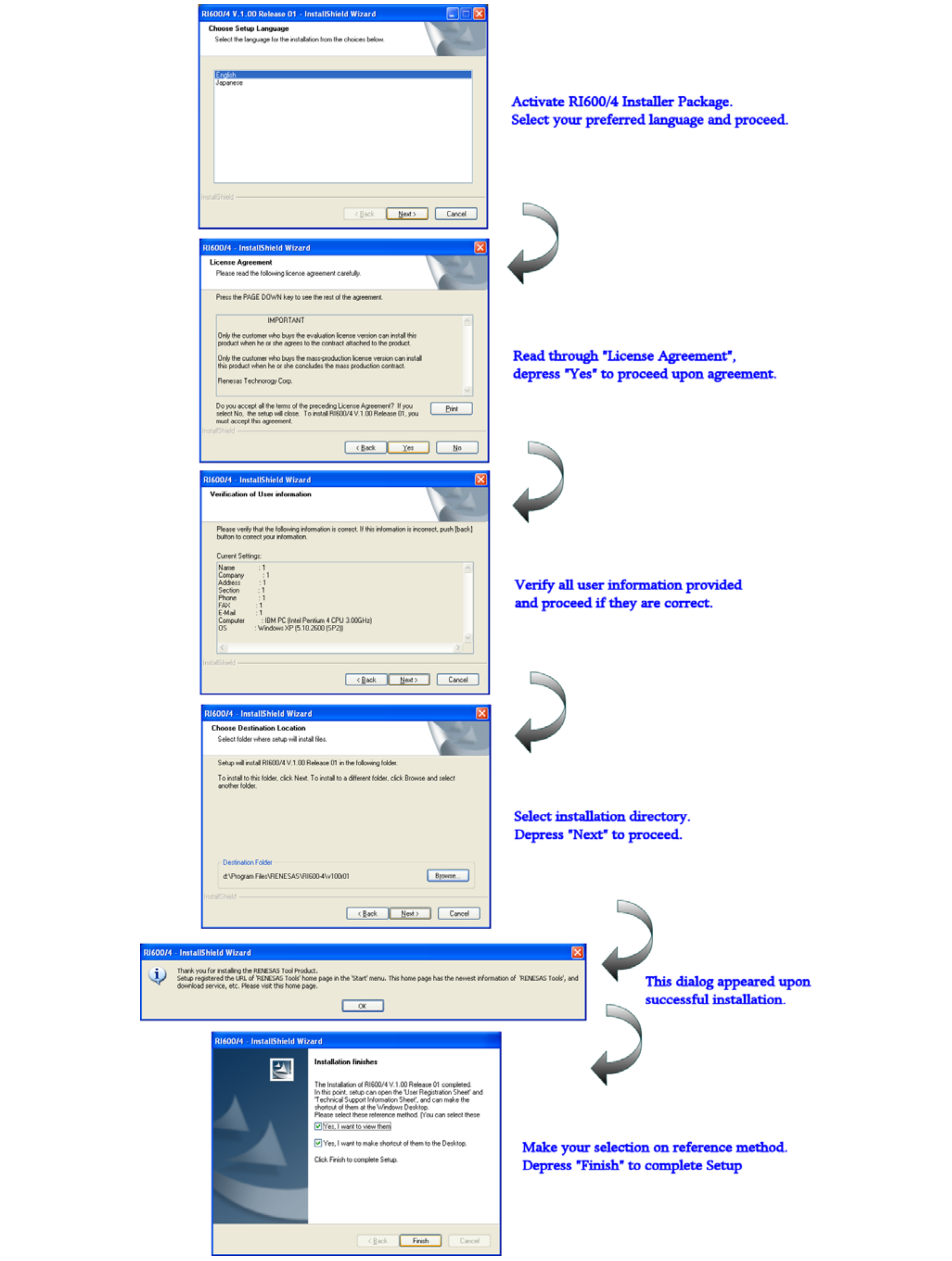
C:\Program Files\Renesas\Hew_001

Notes: 1. If “C/C++ Compiler Package for RX Family” has been installed correctly, Tools Administration in HEW should show “RX Standard Toolchain”, “RX600 Series CPU” and “RX600 Simulator Target Platform”.

2. If “E1/E20 Emulator Debugger Package” has been installed correctly, Tools Administration in HEW should show “ RX E1/E20 SYSTEM”.

Figure 2 Validating Installation of RX Compiler and Debugger Package

After performing the validation, being the installation of RI600/4 by following the steps described below.



RI600/4 V.1.00 Release 01 - InstallShield Wizard

Choose Setup Language

Select the language for the installation from the choices below.

English
Japanese

Activate RI600/4 Installer Package.
Select your preferred language and proceed.

RI600/4 - InstallShield Wizard

License Agreement

Please read the following license agreement carefully.

Press the PAGE DOWN key to see the rest of the agreement.

IMPORTANT

Only the customer who buys the evaluation license version can install this product when he or she agrees to the contract attached to the product.

Only the customer who buys the mass-production license version can install this product when he or she concludes the mass production contract.

Renesas Technorogy Corp.

Do you accept all the terms of the preceding License Agreement? If you select No, the setup will close. To install RI600/4 V.1.00 Release 01, you must accept this agreement.

Read through "License Agreement",
depress "Yes" to proceed upon agreement.

RI600/4 - InstallShield Wizard

Verification of User information

Please verify that the following information is correct. If this information is incorrect, push [back] button to correct your information.

Current Settings:

```
Name        : 1
Company     : 1
Address     : 1
Section     : 1
Phone       : 1
FAX         : 1
E-Mail      : 1
Computer    : IBM PC (Intel Pentium 4 CPU 3.00GHz)
OS          : Windows XP (5.10.2600 (SP2))
```

Verify all user information provided
and proceed if they are correct.

RI600/4 - InstallShield Wizard

Choose Destination Location

Select folder where setup will install files.

Setup will install RI600/4 V.1.00 Release 01 in the following folder.

To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

Destination Folder
d:\Program Files\RENESAS\RI600-4\v100r01

Select installation directory.
Depress "Next" to proceed.

RI600/4 - InstallShield Wizard

Thank you for installing the RENESAS Tool Product.
Setup registered the URL of 'RENESAS Tools' home page in the 'Start' menu. This home page has the newest information of 'RENESAS Tools', and download service, etc. Please visit this home page.

This dialog appeared upon
successful installation.

RI600/4 - InstallShield Wizard

Installation finishes

The Installation of RI600/4 V.1.00 Release 01 completed.
In this point, setup can open the 'User Registration Sheet' and 'Technical Support Information Sheet', and can make the shortcut of them at the Windows Desktop.
Please select these reference method. (You can select these

☑ Yes, I want to view them

☑ Yes, I want to make shortcut of them to the Desktop.

Click Finish to complete Setup.

Make your selection on reference method.
Depress "Finish" to complete Setup

Figure 3 Procedures in RI600/4 Installation

Upon correct installation of the package, the following files can be found.

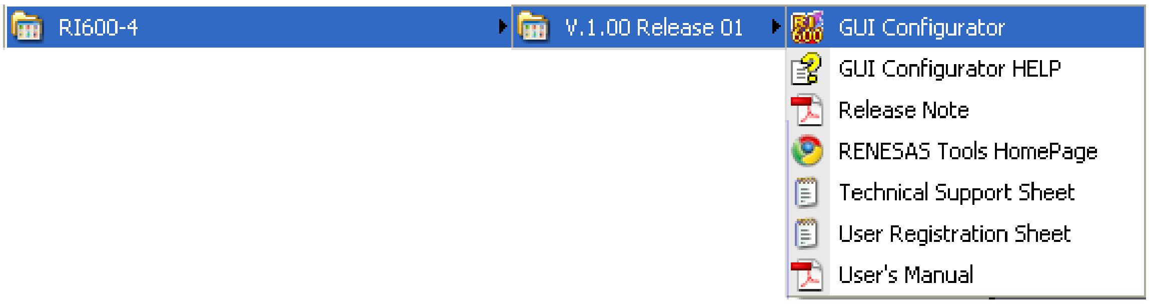
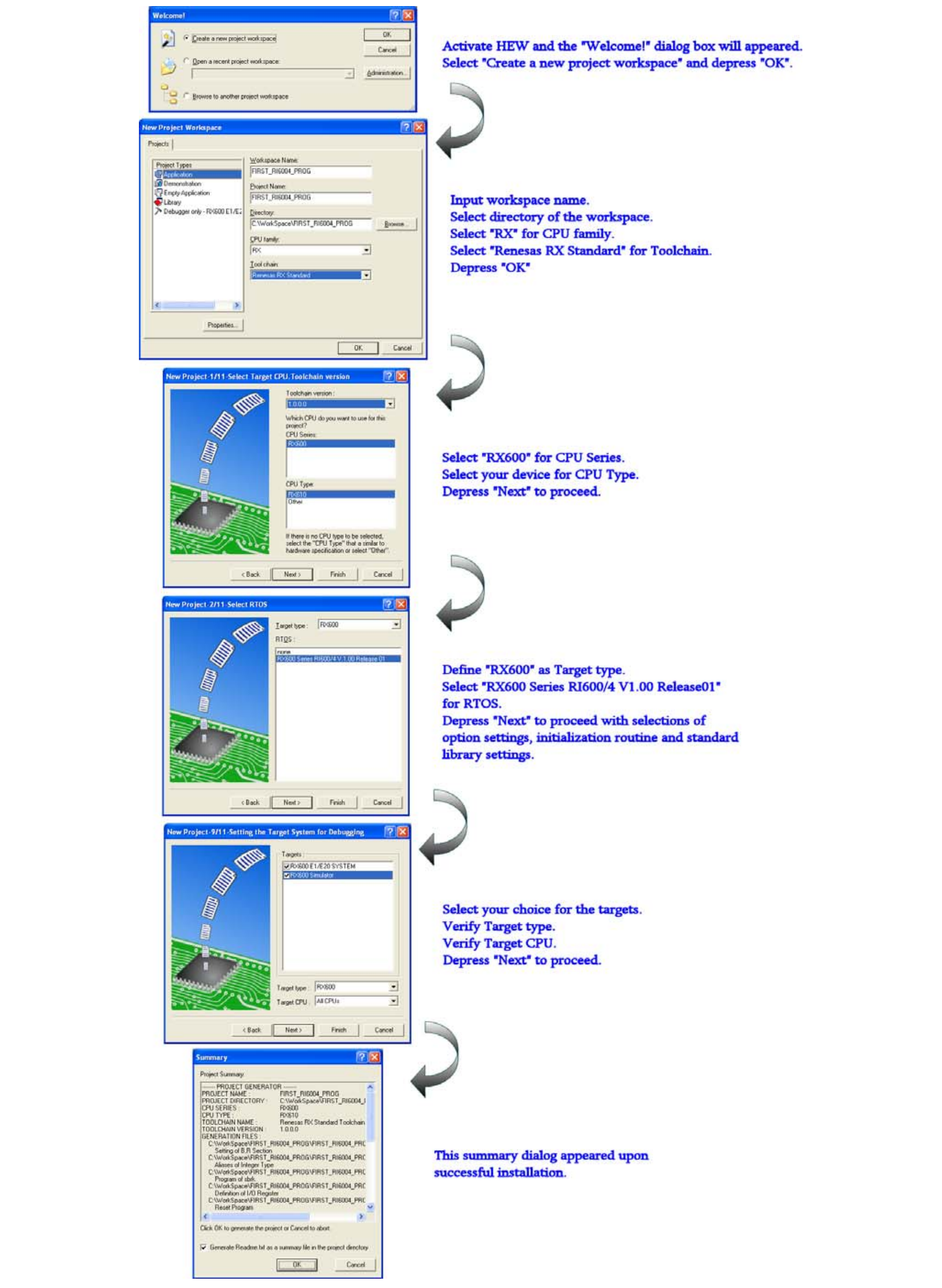


Figure 4 RI600/4 Directory Listing

4. Creating the First Workspace with RI600/4

Start High-Performance Embedded Workshop and follow the creation procedures described in Figure 5.



Welcome!

(●) Create a new project workspace

○ Open a recent project workspace:

○ Browse to another project workspace

Activate HEW and the "Welcome!" dialog box will appeared.
Select "Create a new project workspace" and depress "OK".

New Project Workspace

Projects

Project Types: Application, Demonstration, Empty Application, Library, Debugger only - RX600 E1/E2

Workspace Name: FIRST_RI6004_PROG

Project Name: FIRST_RI6004_PROG

Directory: C:\WorkSpace\FIRST_RI6004_PROG

CPU family: RX

Tool chain: Renesas RX Standard

Input workspace name.
Select directory of the workspace.
Select "RX" for CPU family.
Select "Renesas RX Standard" for Toolchain.
Depress "OK"

New Project-1/11-Select Target CPU,Toolchain version

Toolchain version : 1.0.0.0

Which CPU do you want to use for this project?

CPU Series: RX600

CPU Type: RX610, Other

If there is no CPU type to be selected, select the "CPU Type" that a similar to hardware specification or select "Other".

Select "RX600" for CPU Series.
Select your device for CPU Type.
Depress "Next" to proceed.

New Project-2/11-Select RTOS

Target type : RX600

RTOS : none, RX600 Series RI600/4 V.1.00 Release 01

Define "RX600" as Target type.
Select "RX600 Series RI600/4 V1.00 Release01" for RTOS.
Depress "Next" to proceed with selections of option settings, initialization routine and standard library settings.

New Project-9/11-Setting the Target System for Debugging

Targets:

☑ RX600 E1/E20 SYSTEM

☑ RX600 Simulator

Target type : RX600

Target CPU : All CPUs

Select your choice for the targets.
Verify Target type.
Verify Target CPU.
Depress "Next" to proceed.

Summary

Project Summary

```
---- PROJECT GENERATOR ----
PROJECT NAME :          FIRST_RI6004_PROG
PROJECT DIRECTORY :     C:\WorkSpace\FIRST_RI6004_I
CPU SERIES :            RX600
CPU TYPE :              RX610
TOOLCHAIN NAME :        Renesas RX Standard Toolchain
TOOLCHAIN VERSION :     1.0.0.0
GENERATION FILES :
   C:\WorkSpace\FIRST_RI6004_PROG\FIRST_RI6004_PRC
     Setting of B,R Section
   C:\WorkSpace\FIRST_RI6004_PROG\FIRST_RI6004_PRC
     Aliases of Integer Type
   C:\WorkSpace\FIRST_RI6004_PROG\FIRST_RI6004_PRC
     Program of sbrk
   C:\WorkSpace\FIRST_RI6004_PROG\FIRST_RI6004_PRC
     Definition of I/O Register
   C:\WorkSpace\FIRST_RI6004_PROG\FIRST_RI6004_PRC
     Reset Program
```

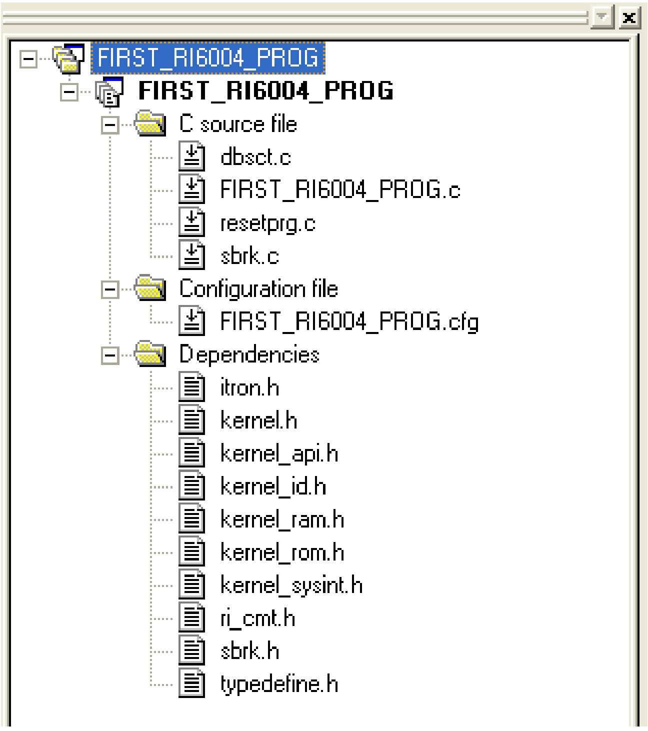
Click OK to generate the project or Cancel to abort

☑ Generate Readme.txt as a summary file in the project directory

This summary dialog appeared upon successful installation.

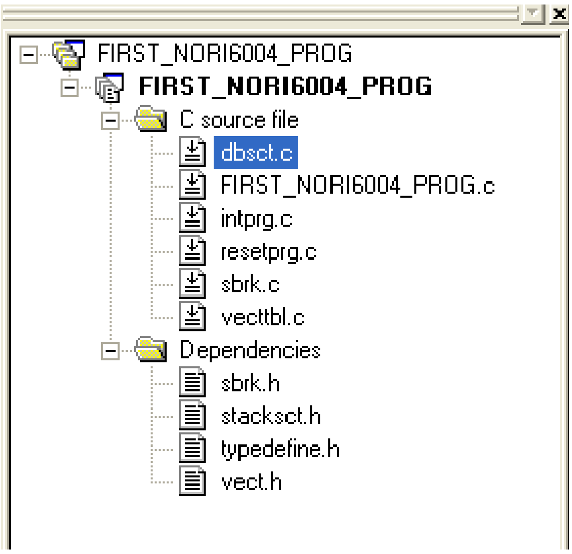
Figure 5 Procedures in Creating Workspace with RI600/4

Upon creation and compilation of the workspace “FIRST_RI6004_PROG”, user will get to see the following file structure. Figure 6 shows the file structure of a workspace created without RI600/4 for comparison.



```
FIRST_RI6004_PROG
  FIRST_RI6004_PROG
    C source file
      dbsct.c
      FIRST_RI6004_PROG.c
      resetprg.c
      sbrk.c
    Configuration file
      FIRST_RI6004_PROG.cfg
    Dependencies
      itron.h
      kernel.h
      kernel_api.h
      kernel_id.h
      kernel_ram.h
      kernel_rom.h
      kernel_sysint.h
      ri_cmt.h
      sbrk.h
      typedefine.h
```

Figure 6 “FIRST_RI6004_PROG” File Structure



```
FIRST_NORI6004_PROG
  FIRST_NORI6004_PROG
    C source file
      dbsct.c
      FIRST_NORI6004_PROG.c
      intprg.c
      resetprg.c
      sbrk.c
      vecttbl.c
    Dependencies
      sbrk.h
      stacksct.h
      typedefine.h
      vect.h
```

Figure 7 Workspace Created without RI600/4

Comparing both workspaces that are created with and without RI600/4 as shown in Figure 6 and Figure 7 respectively, it can be deduced few more files have been added for the former. Table 1 Description of RI600/4 Files provides a summary of the files added and their individual purpose.

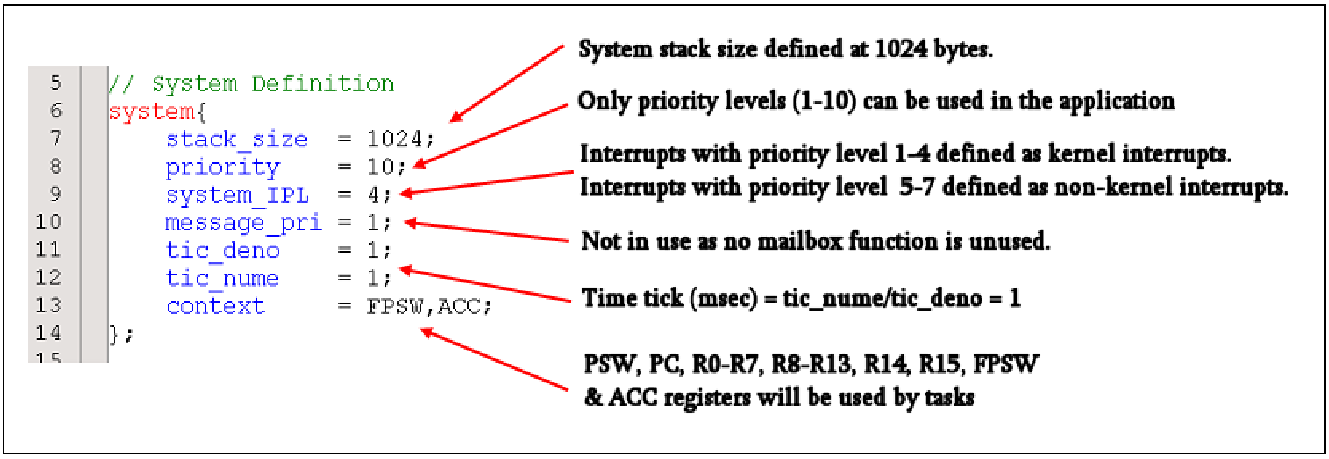
Table 1 Description of RI600/4 Files

File	Descriptions
FIRST_RI6004_PROG.cfg	The configuration file for the definition of RI600/4 RTOS resources
itron.h	Contain definitions of data types, constants and macros, and other definitions specified in ITRON General Definitions section
kernel.h	Contain all service call declarations, data types, constants, and macro definitions specified in the kernel specification
kernel._api.h	Define service call functions declarations
kernel_id.h	Define ID names, kernel configuration macros specified in the cfg file, proto-type declaration of tasks and handlers, etc.
kernel_ram.h	Define kernel RAM data structures
kernel_rom.h	Define kernel ROM data structures
kernel_sysint.h	Contains definitions necessary to invoke service calls by an INT instruction
ri_cmt.h	Contains the timer driver source code

5. A Walkthrough of “FIRST_RI6004_PROG” Workspace

5.1 Understanding the Configuration File “FIRST_RI6004_PROG.cfg” Settings

Upon the creation of the workspace, few objects have been defined in the configuration file (i.e. “FIRST_RI6004_PROG.cfg”) and its corresponding handlers declared in the source file (i.e. “FIRST_RI6004_PROG.c”). Figure 8 to Figure 14 interpret the settings of respective objects definitions.



```
// System Definition
system{
    stack_size   = 1024;
    priority     = 10;
    system_IPL   = 4;
    message_pri  = 1;
    tic_deno     = 1;
    tic_nume     = 1;
    context      = FPSW,ACC;
};
```

System stack size defined at 1024 bytes.

Only priority levels (1-10) can be used in the application

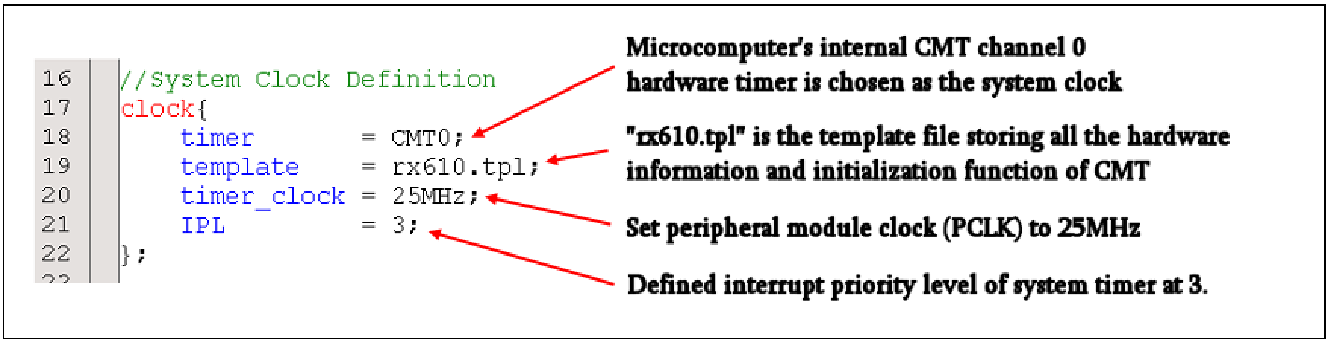
Interrupts with priority level 1-4 defined as kernel interrupts.
Interrupts with priority level 5-7 defined as non-kernel interrupts.

Not in use as no mailbox function is unused.

Time tick (msec) = tic_nume/tic_deno = 1

PSW, PC, R0-R7, R8-R13, R14, R15, FPSW
& ACC registers will be used by tasks

Figure 8 System Definition Settings



```
//System Clock Definition
clock{
    timer        = CMT0;
    template     = rx610.tpl;
    timer_clock  = 25MHz;
    IPL          = 3;
};
```

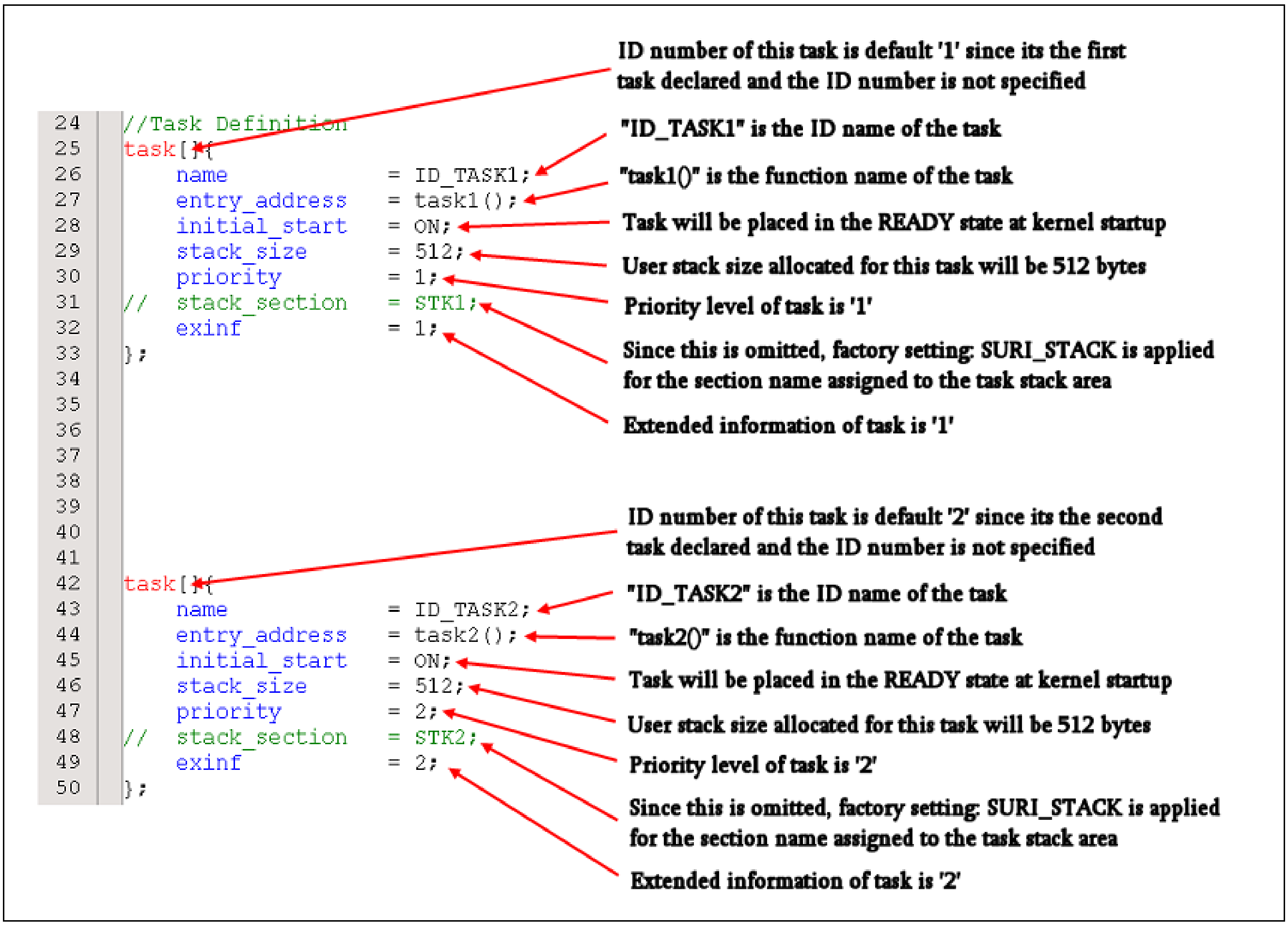
Microcomputer's internal CMT channel 0 hardware timer is chosen as the system clock

"rx610.tpl" is the template file storing all the hardware information and initialization function of CMT

Set peripheral module clock (PCLK) to 25MHz

Defined interrupt priority level of system timer at 3.

Figure 9 System Clock Definition Settings



```
//Task Definition
task[]{
    name            = ID_TASK1;
    entry_address   = task1();
    initial_start   = ON;
    stack_size      = 512;
    priority        = 1;
//  stack_section   = STK1;
    exinf           = 1;
};








task[]{
    name            = ID_TASK2;
    entry_address   = task2();
    initial_start   = ON;
    stack_size      = 512;
    priority        = 2;
//  stack_section   = STK2;
    exinf           = 2;
};
```

ID number of this task is default '1' since its the first task declared and the ID number is not specified

"ID_TASK1" is the ID name of the task

"task1()" is the function name of the task

Task will be placed in the READY state at kernel startup

User stack size allocated for this task will be 512 bytes

Priority level of task is '1'

Since this is omitted, factory setting: SURI_STACK is applied for the section name assigned to the task stack area

Extended information of task is '1'

ID number of this task is default '2' since its the second task declared and the ID number is not specified

"ID_TASK2" is the ID name of the task

"task2()" is the function name of the task

Task will be placed in the READY state at kernel startup

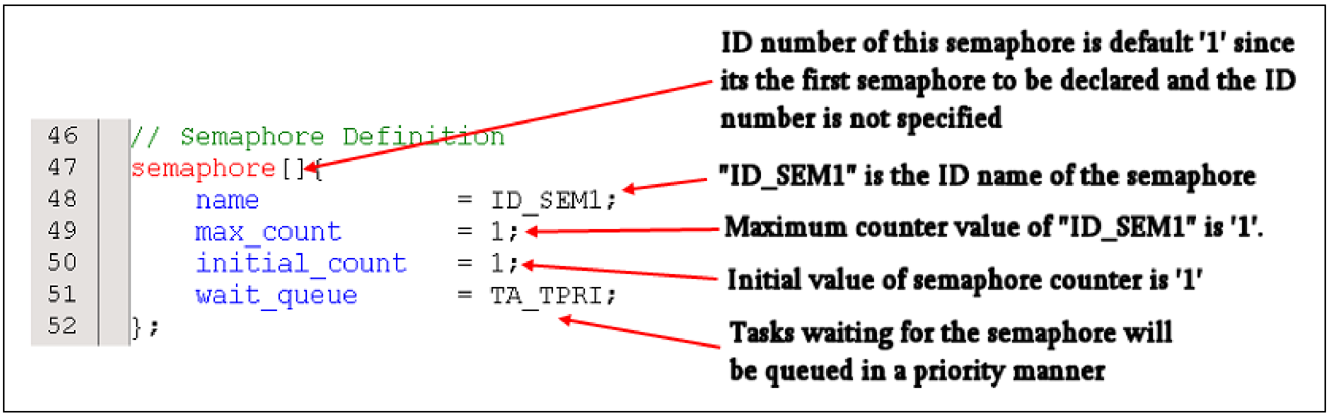
User stack size allocated for this task will be 512 bytes

Priority level of task is '2'

Since this is omitted, factory setting: SURI_STACK is applied for the section name assigned to the task stack area

Extended information of task is '2'

Figure 10 Task Definition Settings



```
// Semaphore Definition
semaphore[]{
    name            = ID_SEM1;
    max_count       = 1;
    initial_count   = 1;
    wait_queue      = TA_TPRI;
};
```

ID number of this semaphore is default '1' since its the first semaphore to be declared and the ID number is not specified

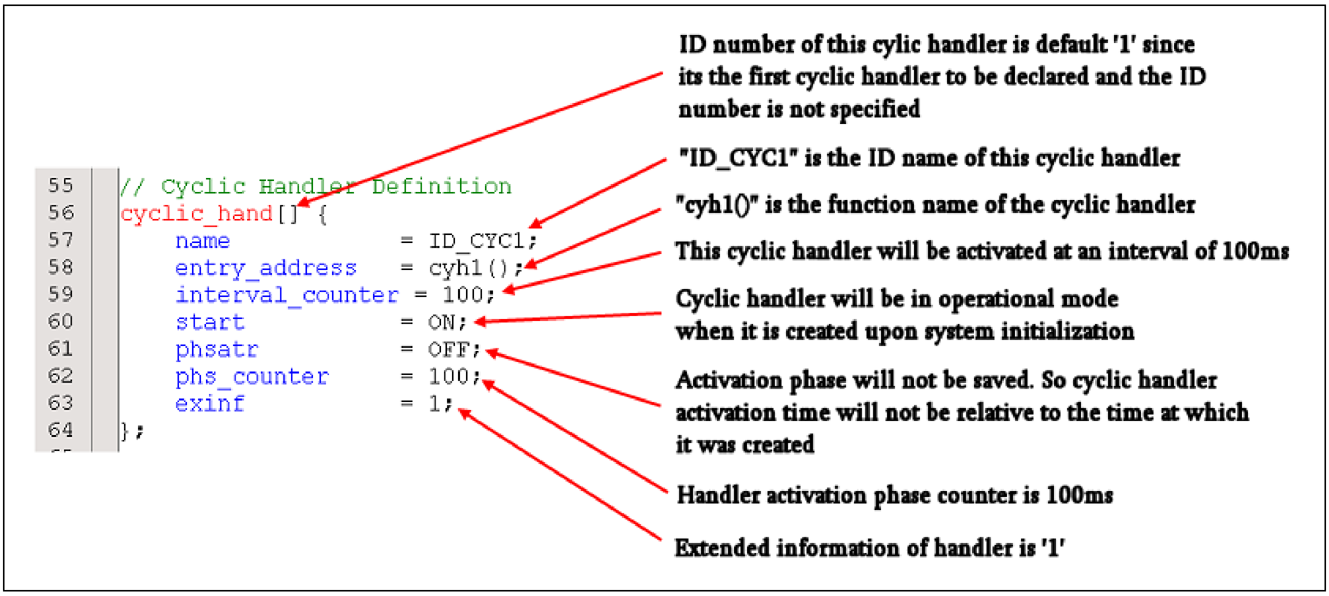
"ID_SEM1" is the ID name of the semaphore

Maximum counter value of "ID_SEM1" is '1'.

Initial value of semaphore counter is '1'

Tasks waiting for the semaphore will be queued in a priority manner

Figure 11 Semaphore Definition Settings



```
// Cyclic Handler Definition
cyclic_hand[] {
    name              = ID_CYC1;
    entry_address     = cyh1();
    interval_counter = 100;
    start             = ON;
    phsatr            = OFF;
    phs_counter       = 100;
    exinf             = 1;
};
```

ID number of this cylic handler is default '1' since its the first cyclic handler to be declared and the ID number is not specified

"ID_CYC1" is the ID name of this cyclic handler

"cyh1()" is the function name of the cyclic handler

This cyclic handler will be activated at an interval of 100ms

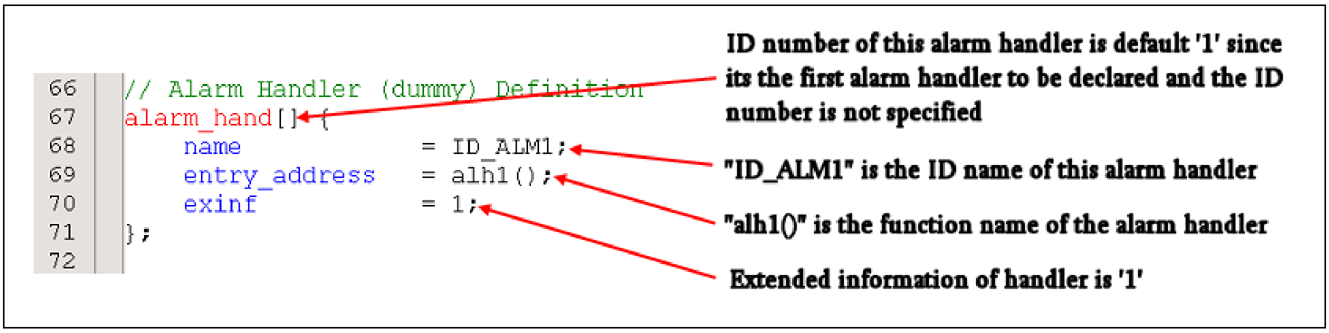
Cyclic handler will be in operational mode when it is created upon system initialization

Activation phase will not be saved. So cyclic handler activation time will not be relative to the time at which it was created

Handler activation phase counter is 100ms

Extended information of handler is '1'

Figure 12 Cyclic Handler Definition Settings



```
// Alarm Handler (dummy) Definition
alarm_hand[] {
    name              = ID_ALM1;
    entry_address     = alh1();
    exinf             = 1;
};
```

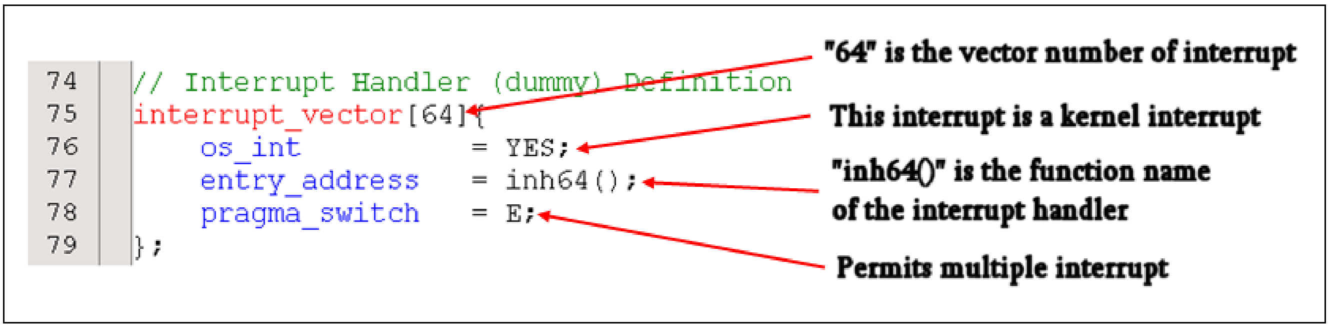
ID number of this alarm handler is default '1' since its the first alarm handler to be declared and the ID number is not specified

"ID_ALM1" is the ID name of this alarm handler

"alh1()" is the function name of the alarm handler

Extended information of handler is '1'

Figure 13 Alarm Handler Definition Settings



```
// Interrupt Handler (dummy) Definition
interrupt_vector[64]{
    os_int            = YES;
    entry_address     = inh64();
    pragma_switch     = E;
};
```

"64" is the vector number of interrupt

This interrupt is a kernel interrupt

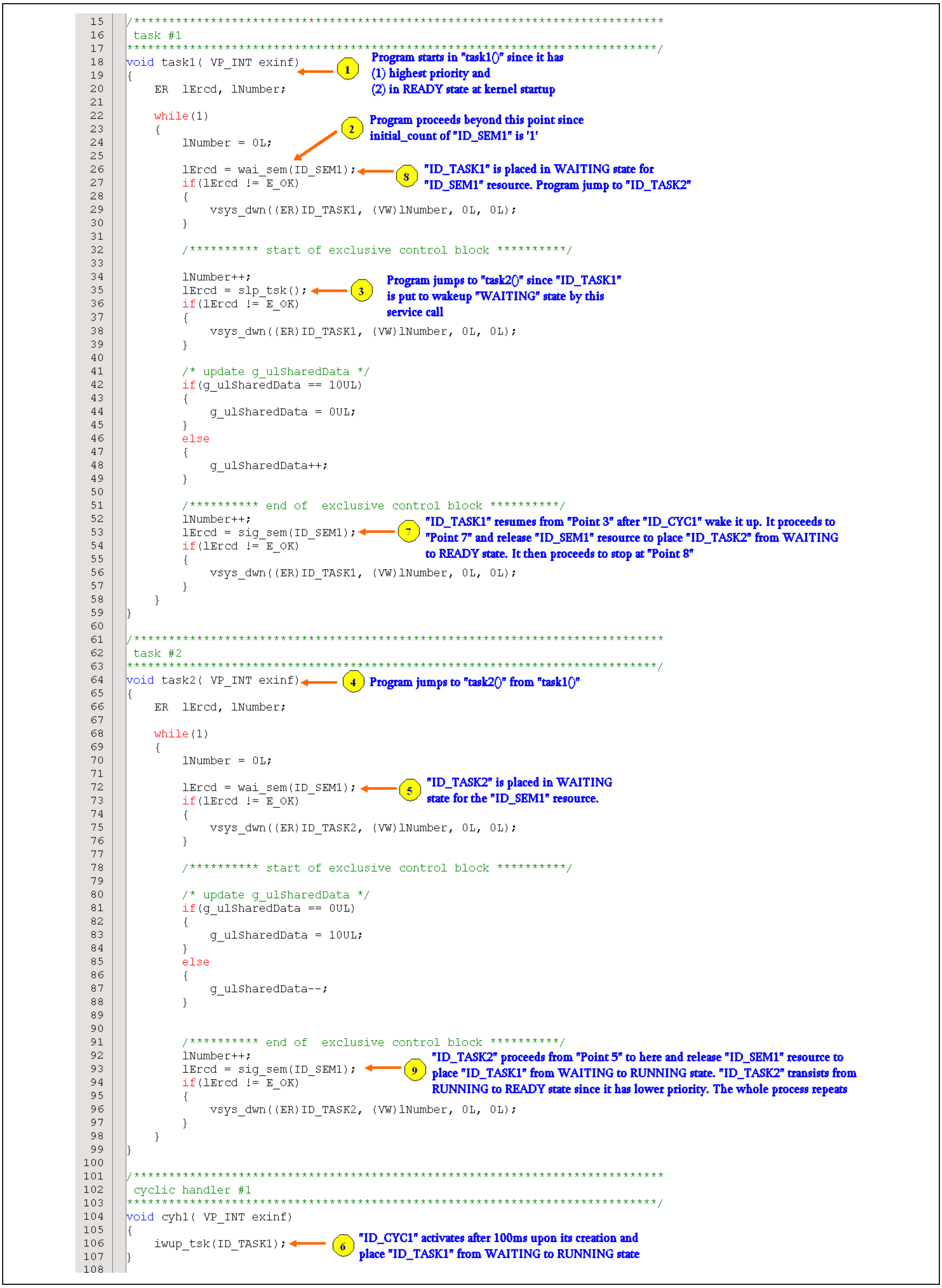
"inh64()" is the function name of the interrupt handler

Permits multiple interrupt

Figure 14 Interrupt Handler Definition Settings

5.2 Understanding the Program Flow in "FIRST_RI6004_PROG.c"

Figure 15 explains the program flows of the application.



```
/***********************************************************************
 task #1
***********************************************************************/
void task1( VP_INT exinf)
{
    ER  lErcd, lNumber;

    while(1)
    {
        lNumber = 0L;

        lErcd = wai_sem(ID_SEM1);
        if(lErcd != E_OK)
        {
            vsys_dwn((ER)ID_TASK1, (VW)lNumber, 0L, 0L);
        }

        /********** start of exclusive control block **********/

        lNumber++;
        lErcd = slp_tsk();
        if(lErcd != E_OK)
        {
            vsys_dwn((ER)ID_TASK1, (VW)lNumber, 0L, 0L);
        }

        /* update g_ulSharedData */
        if(g_ulSharedData == 10UL)
        {
            g_ulSharedData = 0UL;
        }
        else
        {
            g_ulSharedData++;
        }

        /********** end of  exclusive control block **********/
        lNumber++;
        lErcd = sig_sem(ID_SEM1);
        if(lErcd != E_OK)
        {
            vsys_dwn((ER)ID_TASK1, (VW)lNumber, 0L, 0L);
        }
    }
}

/***********************************************************************
 task #2
***********************************************************************/
void task2( VP_INT exinf)
{
    ER  lErcd, lNumber;

    while(1)
    {
        lNumber = 0L;

        lErcd = wai_sem(ID_SEM1);
        if(lErcd != E_OK)
        {
            vsys_dwn((ER)ID_TASK2, (VW)lNumber, 0L, 0L);
        }

        /********** start of exclusive control block **********/

        /* update g_ulSharedData */
        if(g_ulSharedData == 0UL)
        {
            g_ulSharedData = 10UL;
        }
        else
        {
            g_ulSharedData--;
        }


        /********** end of  exclusive control block **********/
        lNumber++;
        lErcd = sig_sem(ID_SEM1);
        if(lErcd != E_OK)
        {
            vsys_dwn((ER)ID_TASK2, (VW)lNumber, 0L, 0L);
        }
    }
}

/***********************************************************************
 cyclic handler #1
***********************************************************************/
void cyh1( VP_INT exinf)
{
    iwup_tsk(ID_TASK1);
}

```

1 Program starts in "task1()" since it has (1) highest priority and (2) in READY state at kernel startup

2 Program proceeds beyond this point since initial_count of "ID_SEM1" is '1'

8 "ID_TASK1" is placed in WAITING state for "ID_SEM1" resource. Program jump to "ID_TASK2"

3 Program jumps to "task2()" since "ID_TASK1" is put to wakeup "WAITING" state by this service call

7 "ID_TASK1" resumes from "Point 3" after "ID_CYC1" wake it up. It proceeds to "Point 7" and release "ID_SEM1" resource to place "ID_TASK2" from WAITING to READY state. It then proceeds to stop at "Point 8"

4 Program jumps to "task2()" from "task1()"

5 "ID_TASK2" is placed in WAITING state for the "ID_SEM1" resource.

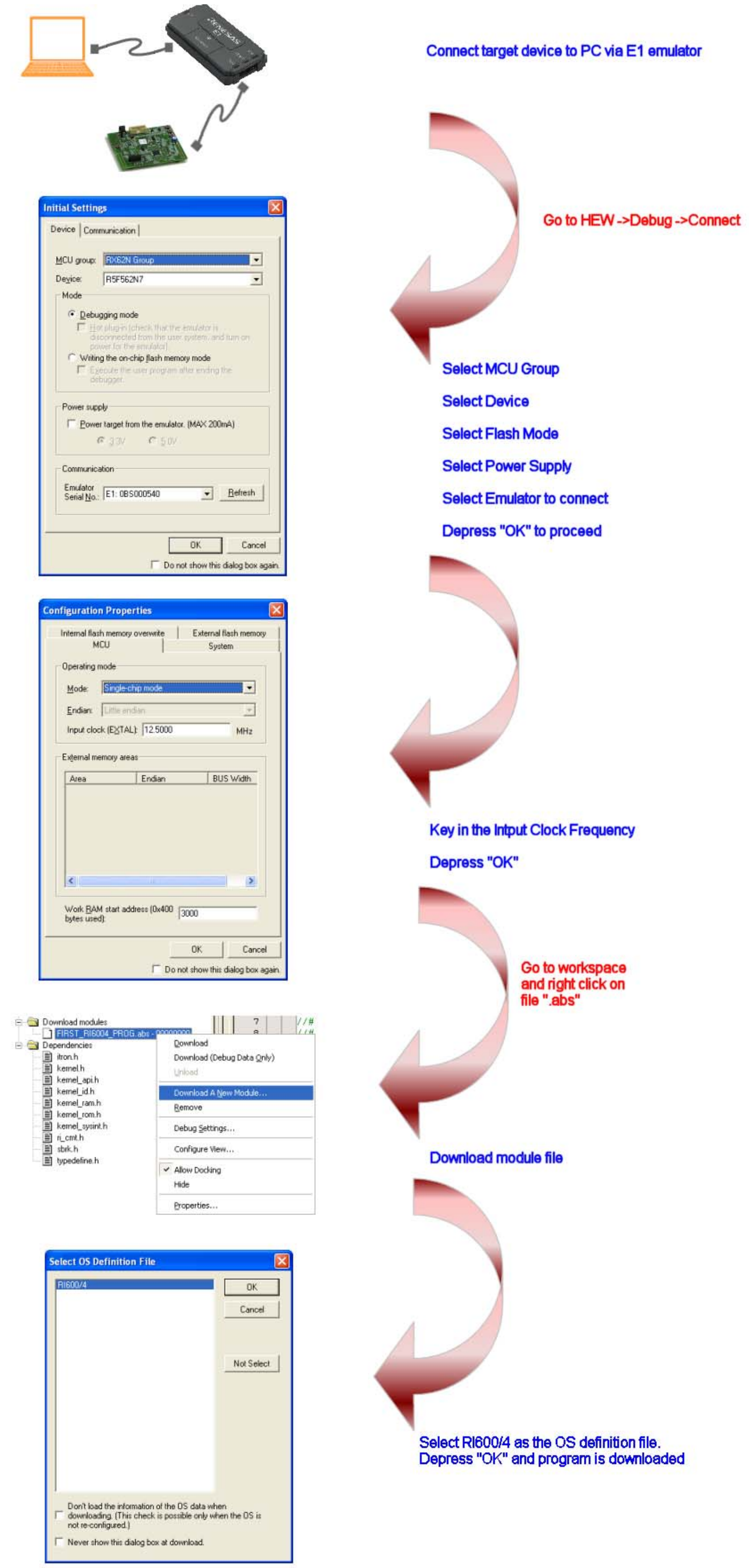
9 "ID_TASK2" proceeds from "Point 5" to here and release "ID_SEM1" resource to place "ID_TASK1" from WAITING to RUNNING state. "ID_TASK2" transists from RUNNING to READY state since it has lower priority. The whole process repeats

6 "ID_CYC1" activates after 100ms upon its creation and place "ID_TASK1" from WAITING to RUNNING state

Figure 15 Application Program Flows in “FIRST_RI6004_PROG.c”

6. Downloading Program with E1 Emulator

Upon the creation and compilation of the workspace, the next step is to download the program to the target device.



Connect target device to PC via E1 emulator

Go to HEW ->Debug ->Connect

Initial Settings

Device | Communication

MCU group: RX62N Group

Device: R5F562N7

Mode

● Debugging mode

□ Hot plug-in (check that the emulator is disconnected from the user system, and turn on power for the emulator)

○ Writing the on-chip flash memory mode

□ Execute the user program after ending the debugger.

Power supply

□ Power target from the emulator. (MAX 200mA)

● 3.3V ○ 5.0V

Communication

Emulator Serial No.: E1: 0BS000540

□ Do not show this dialog box again.

Select MCU Group

Select Device

Select Flash Mode

Select Power Supply

Select Emulator to connect

Depress "OK" to proceed

Configuration Properties

Internal flash memory overwrite | External flash memory | MCU | System

Operating mode

Mode: Single-chip mode

Endian: Little endian

Input clock (EXTAL): 12.5000 MHz

External memory areas

Area	Endian	BUS Width

Work RAM start address (0x400 bytes used): 3000

□ Do not show this dialog box again.

Key in the Intput Clock Frequency

Depress "OK"

Go to workspace and right click on file ".abs"

Download modules

FIRST_RI6004_PROG.abs - 00000000

Dependencies

itron.h
kernel.h
kernel_api.h
kernel_id.h
kernel_ram.h
kernel_rom.h
kernel_sysint.h
ri_cmt.h
sbrk.h
typedefine.h

Download module file

Select OS Definition File

RI600/4

□ Don't load the information of the OS data when downloading. (This check is possible only when the OS is not re-configured.)

□ Never show this dialog box at download.

Select RI600/4 as the OS definition file.
Depress "OK" and program is downloaded

Figure 16 Procedures in Downloading Program with E1 Emulator

7. Reference Documents

User's Manual

- RI600/4 V.1.00 User's Manual
- RX Family Hardware Manual
- RX Family E1/E20 Emulator User's Manual
- High-performance Embedded Workshop V4.08 User's Manual

The latest version can be downloaded from the Renesas Electronics website

Website and Support

Renesas Electronics Website

- http://www.renesas.com/

Inquiries

- http://www.renesas.com/inquiry

All trademarks and registered trademarks are the property of their respective owners.

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Oct.01.10	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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.

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

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

Renesas Electronics America Inc.
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: +49-211-6503-0, 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, R.O.C.
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