

R01AN2460EJ0101 Rev.1.01 Jan 5, 2015

# Introduction

This document is the RX64M Group RX Driver Package User's Manual, version 1.01.

This User's Manual describes basic structures, features and usage of RX Driver Package applications, and about the sample application program using the FIT modules included in this package.

# **Target Device**

RX64M Group (Renesas Starter Kit+ RX64M)

When using this application note with your product, careful evaluation is recommended.

And when using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

### **Related Documents**

- RX Family Board Support Package Module Using Firmware Integration Technology (R01AN1685EU)
- Firmware Integration Technology User's Manual (R01AN1833EU)
- RX Family Adding Firmware Integration Technology Modules to Projects (R01AN1723EU)
- RX Family Adding Firmware Integration Technology Modules to CS+ Projects (R01AN1826EJ)
- The User's Manual provided with the RX Driver Package Application.



# Contents

1.	Overview	. 3
1.1	Applicability	. 3
1.2	Operating Environment	. 3
2.	About RX Driver Package	. 4
2.1	System Structure	. 4
2.2	RX Driver Package Features	5
3	Structure of the RX64M Group RX Driver Package	6
3.1	Folder Structure	6
3.2	Module Structure	.7
3.3	FIT Modules	. 7
4.	Usage Procedures	. 9
4.1	Environment Used	. 9
4.2	Install RX Driver Package in e <sup>2</sup> studio	. 9
4.3	Application Creation1	10
4.3	.1 Create a Workspace and a Project1	10
4.3	.2 Prepare to Embed the FIT Modules1	16
4.3	.3 Install the FIT Modules with the FIT Plugin1	19
4.3	.4 Create an LED Driving Program2	23
4.3	.5 Build and Try Running the Program2	24
5	RX Driver Package Application	77
5.1	RX Driver Package Application Structure	 27
5.2	RX Driver Package Application Features	 27
5.3	RX Driver Package Application Usage Example (when e <sup>2</sup> studio is used)	28
5.4	When Using in Combination with an RX Driver Package Application	28
6.	Supplement	28
6.1	M3S-T4-Tiny (TCP/IP protocol stack )	28
Weh	site and Support	29



# 1. Overview

# 1.1 Applicability

This User's Manual applies to the RX64M Group RX Driver Package, version 1.01.

# 1.2 Operating Environment

This package runs under the operating environment described below.

### Table 1.2.1 Operating Environment

Microcontroller	RX64M Group
Evaluation board	Renesas Starter Kit+ RX64M
Integrated development	e <sup>2</sup> studio, V3.1.2 or later
environment (IDE)	Or:
	CS+ V3.00.00 or later
Cross tools	RX Family C/C++ Compiler Package V2.02.00 or later
Emulator	E1, E20



# 2. About RX Driver Package

The RX Driver Package is a software platform (framework) that combines the following modules to be required for development in a single package. Since the package contains multiple modules, you can start developing immediately without having to obtain each module separately.

- Board Support Package (BSP) module
- FIT peripheral function modules (free version)
- FIT middleware modules (free version)
- FIT interface modules

You can develop the user application layer with ease by using the Sample Application Program (RX Driver Package Application) which utilizes the RX Driver Package.

# 2.1 System Structure

The figure below shows the system structure of the RX Driver Package.



Figure 2.1.1 System Structure



# 2.2 RX Driver Package Features

The RX Driver Package has the following features.

#### (a) Select necessary modules and start developing immediately the application program

You can easily build a system simply by selecting the modules you need from the package. After that, all you have to do is develop the application program.



Figure 2.2.1 An example of system build

#### (b) Free to use

All the modules included in the RX Driver Package can be used free of charge. Free versions of middleware modules such as TCP/IP and file system are included.

#### (c) Can upgrade to paid versions of modules

The free versions of modules in the RX Driver Package can be replaced with commercial (paid) versions. By using a commercial (paid) version, all the functionality of the module will be available, as well as support about a commercial version.

#### (d) Check operation including user application

The RX Driver Package Application is provided as a sample user application that uses the RX Driver Package. The RX Driver Package Application consists of programs for operating each module in the RX Driver Package, and the project files for building the programs. It enables you to start checking the operation of your user application immediately.



# 3. Structure of the RX64M Group RX Driver Package

### 3.1 Folder Structure

The folder structure used in this package is shown below.

When the ZIP file for this package is downloaded from the Renesas web site and decompressed, a folder of the same name will be present and it will contain a FITModules folder, a reference\_documents folder, and this document.

The FITModules folder contains the FIT modules for the modules shown in table 3.3.1 (as ZIP files and XML files).

The reference\_documents folder contains the documentation for using this package in various development environments.



Figure 3.1.1 Folder Structure of the RX64M Group RX Driver Package



# 3.2 Module Structure

The figure below shows the types and structure of the FIT modules included in this package.





# 3.3 FIT Modules

The table below lists the FIT modules included in this package.

Туре	Module	FIT Module Name	Version
Board Support Package	Board support package	r_bsp	2.70
Device Driver	Interrupt Controller (IRQ)	r_irq_rx	1.30
Device Driver	Data transfer controller (DTC)	r_dtc_rx	2.01
Device Driver	DMA controller (DMAC)	r_dmaca_rx	1.01
Device Driver	General-purpose I/O	r_gpio_rx	1.30
Device Driver	Multi-function pin controller (MPC)	r_mpc_rx	1.30
Device Driver	Compare match timer (CMT)	r_cmt_rx	2.40
Device Driver	Compare Match Timer W (CMTW)	r_cmtw_rx	1.00
Device Driver	Real-Time Clock (RTC)	r_rtc_rx	2.10
Device Driver	Serial Communications Interface with FIFO (SCI: Asynchronous/Clock Synchronous)	r_scif_rx	1.00
Device Driver	Serial Communications Interface (SCI: Simple I2C Bus)	r_sci_rx64m	1.00
Device Driver	I2C Bus Interface (RIIC)	r_riic_rx	1.05
Device Driver	Serial Peripheral Interface (RSPI: Device Driver for Serial Memory Control)	r_rspi_smstr_rx	1.07



Туре	Module	FIT Module Name	Version
Device Driver	Quad Serial Peripheral Interface (QSPI: Device Driver for Serial Memory Control)	r_qspi_smstr_rx	1.06
Device Driver	USB basic firmware	r_usb_basic	1.00
Device Driver	USB host communication device class	r_usb_hcdc	1.00
Device Driver	USB host human interface device class	r_usb_hhid	1.00
Device Driver	USB host mass storage class	r_usb_hmsc	1.00
Device Driver	USB peripheral communication device class	r_usb_pcdc	1.00
Device Driver	USB peripheral mass storage class	r_usb_pmsc	1.00
Device Driver	Ethernet controller PTP controller (EPTPC)	r_ptp_api_rx	1.01
Device Driver	Ethernet controller (ETHERC)	r_ether_rx	1.00
Device Driver	12-Bit A/D Converter (S12AD)	r_s12ad_rx64m	1.00
Device Driver	D/A Converter (DAC)	r_dac_rx	2.10
Device Driver	Flash Memory (Flash API)	r_flash_rx	1.11
Device Driver	Sampling rate converter (SRC)	r_src_api_rx	1.10
Device Driver	Serial Sound Interface (SSI)	r_ssi_api_rx	1.00
Device Driver	Parallel Data Capture Unit (PDC)	r_pdc_rx	1.00
Device Driver	Byte Queue Buffer (Data Management)	r_byteq	1.20
Device Driver	Long Queue Buffer (Data Management)	r_longq	1.20
Middleware	M3S-S2-Tiny (ADPCM encoding/decoding library)	r_s2_rx	3.01
Middleware	M3S-T4-Tiny (TCP/IP protocol stack library)	r_t4_rx	2.01
Middleware	M3S-TFAT-Tiny (FAT file system)	r_tfat_rx	3.00
Middleware	DHCP Client Module	r_t4_dhcp_client_r x	1.01
Middleware	DNS Client Module	r_t4_dns_client_rx	1.01
Middleware	FTP Server Module	r_t4_ftp_server_rx	1.02
Middleware	HTTP Server Module	r_t4_http_server_r x	1.03
Interface	POSIX Wrapper	r_posix	1.00
Interface	Socket API Module for M3S-T4-Tiny	r_socket	1.20
Interface	Interface Conversion for Ethernet Controller Driver Module for M3S-T4-Tiny	r_t4_driver_rx64m	1.01
Interface	File driver for FTP server and Web server Module	r_t4_file_driver_rx	1.00
Interface	M3S-TFAT-Tiny Memory Driver Interface Module	r_tfat_driver_rx	1.00

Note: This package includes the M3S-T4-Tiny (TCP/IP protocol stack library) of evaluation version. For the commercial version, please go to the below URL.

http://www.renesas.com/mw/t4



### 4. Usage Procedures

The RX Driver Package allows programs to be easily constructed by using the FIT plugin included in  $e^2$  studio. The remainder of this section presents a simple usage example using  $e^2$  studio. To use CS+, see the document "RX Family Adding Firmware Integration Technology Modules to CS+ Projects (R01AN1826EJ)" included in this package.

# 4.1 Environment Used

The RX64M is used as the target microcontroller and the Renesas Starter Kit+ RX64M is used as the target board. If a different environment is used, replace the specifics used in the example with the ones for that environment as you read.

# 4.2 Install RX Driver Package in e<sup>2</sup> studio

Install the FIT modules in the RX Driver Package into e<sup>2</sup> studio.

- 1. Decompress the downloaded file an\_r01an2460ej0101\_rx64m.zip into an arbitrary directory.
- 2. Open the folder that was decompressed and open the FITModules folder in that folder.
- 3. Select all the files in the **FITModules** folder and click **Copy** in the **Edit** menu.





- 4. Open the e<sup>2</sup> studio install folder (Usually, this will be c:/Renesas/e2\_studio.) and open the **FITModules** folder in that folder.
- 5. Click **Paste** on the **Edit** menu.

The e<sup>2</sup> studio **FITModules** folder will be copied to the FIT modules.

✓ → Computer → Local Disk (C Edit <u>View Tools Help</u>	):) ▶ Renesas ▶ e2_studio ▶ F	TModules 🕨 👻 🐓	Search FITModules	٩
ganize 🔻 🛅 Open New folder			\$₿ ▼	
🕌 kapan lin. 🔸	r_bsp_v2.70.xml	r_eeprom_spi_v2.30.zip	r_posix_v1.00.xml	🔒 r_s12ad_
🔉 inspanilas	🔒 r_bsp_v2.70.zip	r_ether_rx_v1.00.xml	n_posix_v1.00.zip	🔮 r_sci_rx64
and the second	r_byteq_v1.20.xml	r_ether_rx_v1.00.zip	r_ptp_api_rx_v1.01.xml	🔒 r_sci_rx64
🖗 shift.heep	🔒 r_byteq_v1.20.zip	r_flash_rx_v1.11.xml	🚹 r_ptp_api_rx_v1.01.zip	🔮 r_scif_rx_
🔰 sipalita julit. 🖓	r_cmt_rx_v2.40.xml	r_flash_rx_v1.11.zip	r_qspi_smstr_rx_v1.06.xml	🔒 r_scif_rx_
Marked (Climb)	r_cmt_rx_v2.40.zip	r_gpio_rx_v1.30.xml	r_qspi_smstr_rx_v1.06.zip	🔮 r_socket
퉬 Renesas	r_cmtw_rx_v1.00.xml	🔒 r_gpio_rx_v1.30.zip	r_riic_rx_v1.50.xml	🔒 r_socket
🔉 m. direktirjan, A., 60401	r_cmtw_rx_v1.00.zip	r_irq_rx_v1.30.xml	n_riic_rx_v1.50.zip	🖭 r_src_api
🔉 ang ditardaling ng pigabada	r_dac_rx_v2.10.xml	r_irq_rx_v1.30.zip	r_rspi_smstr_rx_v1.07.xml	🔒 r_src_api
鷆 e2_studio	🔒 r_dac_rx_v2.10.zip	r_longq_v1.20.xml	r_rspi_smstr_rx_v1.07.zip	🕑 r_ssi_api
🍌 DebugComp	r_dmaca_rx_v1.01.xml	r_longq_v1.20.zip	r_rtc_rx_v2.10.xml	🔒 r_ssi_api
🎉 Drivers	r_dmaca_rx_v1.01.zip	r_mpc_rx_v1.30.xml	r_rtc_rx_v2.10.zip	🔮 r_t4_dhc
鷆 eclipse	r_dtc_rx_v2.01.xml	r_mpc_rx_v1.30.zip	r_s2_rx_v3.01.xml	🚮 r_t4_dhc
鷆 etc	r_dtc_rx_v2.01.zip	r_pdc_rx_v1.00.xml	<pre>r_s2_rx_v3.01.zip</pre>	🔮 r_t4_dns
FITModules	r_eeprom_spi_v2.30.xml	r_pdc_rx_v1.00.zip	r_s12ad_rx64m_v1.00.xml	r_t4_dns_
鷆 internal 🔹	• • •			•
86 items selected Show more details				

Open the **FITModules** folder and click **Paste** on the **Edit** menu. The folder will be copied.

# 4.3 Application Creation

In this section, create a simple application that drives an LED.

### 4.3.1 Create a Workspace and a Project

First, create a new workspace and a new project.

- 1. Start  $e^2$  studio.
- 2. Enter an arbitrary workspace folder in the displayed dialog box and click **OK**.





3. When the following window is displayed, click Workbench.



4. When the workbench has started, select New from the File menu and click C Project.

e <sup>2</sup>	C/C++ - e2 studio													
File	Edit Source Refactor	Navigate Search F	rojec	t Run Window Help										
	New	Alt+Shift+N ►	C	Renesas FIT Module		9								
	Open File		C <sup>4</sup>	C Project			-	C	Clic	Clic	Click	Click h	Click h	Click he
	Close	Ctrl+W	C.¥	C++ Project										
	Close All	Ctrl+Shift+W	C+	Makefile Project with Existing Code										
	Save	Ctrl+S	Ľ	Project										
	Save As	carro	C++	Convert to a C/C++ Project (Adds C/	(C++ Nature)									
	Save All	Ctrl+Shift+S	62	Source Folder										
	Revert		<u></u>	Folder										
	Move		C	Source File										
-A	Rename	F2	h	Header File										
2 2	Refresh	F5		File from Template										
	Convert Line Delimiters To	*	ଞ 	Tack										
	Print	Ctrl+P												
-		c		Other	Ctrl+N	N								
	Switch Workspace	•	L .											
	Kestart													
è	Import		L											
4	Export													
	Properties	Alt+Enter												
	Exit													
_														



5. Enter the project name. For the project type, click **Sample Project** under **Executable** (**Renesas**). For the tool chain, click **Renesas RXC Toolchain**. After making these settings, click **Next**.

e <sup>2</sup> C Project		
C Project		
Create C project of selected type		
Project name: led sample		Enter the
Elojeer land, leajampie		project name
Use <u>default</u> location		project name.
Location: C:\sample_workspace\led_samp	le B <u>r</u> owse	
Create Directory for Project		
Project type:	l oolchains:	
Executable (Renesas)	KPIT GNUARM-RZ-EABI Toolchain	Oliala harra
Sample Project	KPIT GNUKL/8-ELF Toolchain	Click nere.
Sample Project	KPIT GNURX-ELF Loolchain	
Debug-Only Project	KPIT GNUSH-ELF Toolchain	
Executable	Renesas RAC Toolchain	
Executable (IAR)	Renesas SHC Toolchain	
Shared Library		
Static Library		
V800 Standalone Executable (Green Hills)		
V800 Standalone Static Library (Green Hills	)	
V800 ThreadX Executable (Green Hills)		
Makefile project		
4 III +		
Show project types and toolchains only i	f they are supported on the platform	
	lext > Einish Cancel	
-		

6. Select the target. Click the "..." button under **Target Selection** and select **R5F564MLCxFC**. After making these settings, click **Next**.

e <sup>2</sup> C Project		
e2 studio - Project Generation Select Target Specific Settings		
Toolchain Version :       v2.01.00         Debug Hardware:       E1         Data endian :       Little-endian data         Select Target:       RSF56107VxFP         Select Target:       RSF56107VxFP         Select Configurations:       Image: Debug using hardware         Debug Using Simulator :       Debug using simulator         Release (no debug)       : Project without any debug int         Build configurations will be created in the project only for the options, however by default the project will be built for the a first configuration selected from group. Based on the device (RX600) the debug hardware (E1) and debug target (RSF5610 configuration will be automatically created for you.	formation he selected debug mode active configuration i.e., e selection you made 17VxFP), debug	Click here and select <b>R5F564MLCxFC</b> .
	<u>Finish</u> Cancel	Click here.



#### 7. Simply click **Next** here.



#### 8. Simply click Next here.

e <sup>2</sup> C Project			
e2 studio - Project Generation Select Additional CPU Options			
Select Additional CPU Options: Round: Precision of Double: Sign of Char: Sign of bit Field: Allocate from Lower Bit	Nearest Single precision Unsigned Unsigned Lower bit	•	
Width of Divergence of Function:	24 Bit	•	
Specify Global Options: Denormalized number allow Replace from int with short Enum size is made the sma Pack structures, unions and Use try, throw and catch of	ved as a result llest l classes C++ d of C++		
(?) <u>&lt; Back</u>	Next >	Cancel	Click here



### 9. Simply click **Next** here.

Shipiy chek itext her	<u>.</u>	
e <sup>2</sup> C Project		
e2 studio - Project Generation Global Options Settings		
Patch code generation	None •	
Fast interrupt vector register:	None	
ROM:	None	
RAM:	None	
Address (H'):	0000000	
Address Register:	None 🔻	
?	Back Next > Finish Cancel	Click here.

#### 10. Select C(C99) under Library Structure and click Next.





#### 11. Clear all four check boxes and click Finish.

Stack/Heap Configuration	Clear all the
User's Stack Size: (H') 100	check boxe
Interrupt Stack Size: (H') 300	
Use Heap Memory	
Heap Size: (H') 400	
Generate Hardware Setup Function None	
	Click boro
Sack Next > Finish Cancel	

12. Click OK. The project will be generated.





### 4.3.2 Prepare to Embed the FIT Modules

Since the FIT modules are independently initialized in BSP, it is necessary to modify certain aspects of the project generated by  $e^2$  studio.

1. In Project Explorer, delete the dbsct.c and typedefine.h files in the src folder.



2. In Project Explorer, click Project and open the properties by clicking Renesas Tool Settings in the toolbar.





#### 3. Click **Settings** under **C/C++ Build** in the properties and click **Section** under **Linker**.

e <sup>2</sup> Properties for led_sample				- 6		
type filter text	Settings	> • •				
▷ Resource Builders ▲ C/C++ Build	Configuration: HardwareDebug [ A	ctive ]		<ul> <li>Manage Configuration</li> </ul>	15	
Build Variables Change Toolchain Vers Dependency Scan Device Environment	Tool Settings Build Steps	Build Artifact 🗟 Binar	ry Parsers 🧕 Error Parsers			
Logging	🖉 Source file	Address	Section Name			Olively being
Settings Tool Chain Editor ▷ C/C++ General Project References Run/Deug Settings ▷ Task Repository	By Object     By Object     By Object     By Optimize     By Optimize	0x0000004	SU SI B_1 R_1 B_2 R_2 B R PResetPRG C_1 C_2 C C C C C S* D* W* L PIntPRG P	Add Section Remove Section Move Up Move Down	E	— Click here.
	A List	0xFFFFF80	EXCEPTVECT			— Click here.
< >	Cotimize     Cotimize	Override Linker Script:	RESETVECT	Browse	cel	

### 4. In Section View, delete PResetPRG and PIntPRG. Click each section and click Delete Section.





5. Add an **OPT\_MEMORY** section after the **R** section. Click the **R** section and click **Add Section**. Click the added section, **NEW\_SECTION\_1**, and modify the name to be **OPT\_MEMORY**, and input the address to be **0x00120064**.

After completing the addition, click **OK**.

Address	Section Name		
0x00000004	SU		
	SI		
	B_1		Click here
	B 2		
	R 2	=	
	в	Add Section	
	R		
0x00120064	OPT_MEMORY	Remove Section	Add here.
0xFFC00000	C_1		
	C_2	Move Up	
	С		
	C\$*	Move Down	
	D*		
	W*		
	L		
	P		
0xFFFFFF80	EXCEPTVECT		
0xFFFFFFC	RESETVECT		
rride Linker Scrint			
inde einker senpe		Browse	
	Import	хроп ке-Арріу	



### 4.3.3 Install the FIT Modules with the FIT Plugin.

Install the required modules with the FIT plugin into the created project.

Here, install the BSP module (r\_bsp) and the compare match timer driver (r\_cmt\_rx).

1. In the File menu, select New and click Renesas FIT Module.

e <sup>2</sup>	C/C++ - e2 studio					
File	Edit Source Refactor N	lavigate Search I	Proje	ct Run Window Help		
	New	Alt+Shift+N ►	C	Renesas FIT Module	1	Click here
	Open File		C.	C Project		
	Close	Ctrl+W	C.	C++ Project		
	Close All	Ctrl+Shift+W	C2	Makefile Project with Existing Code		
			2	Project		
	Save	Ctrl+S	Cee	Convert to a C/C++ Project (Adds C/C	C++ Nature)	
딟	Save As		69	Source Folder	,	
	Save All	Ctrl+Shift+S		Folder		
	Revert			Source File		
	Move			Header File		
	Rename	F2	•	File from Template		
8	Refresh	F5	R	Class		
	Convert Line Delimiters To	+		Task		
Ð	Print	Ctrl+P	C)	Other	Ctrl+N	
	Switch Workspace		-			

 Select the created project with Project to Add FIT Module to. Next, select RX64M under Group and select RSKRX64M from Target Board. Next, click r\_bsp and r\_cmt\_rx in the module list and click Finish.

e <sup>2</sup> Add FIT Module		
FIT Modules		
Select FIT Modules to ad	d to the selected project	
Name of the project to a	idd FIT modules: Led_sample	Select the project.
Family RX 👻 T	arget Board RSKRX64M   Function Any	
Series RX600 🕶 T	oolchain Any - Application Any -	
Group RX64M -	Reset	Select this item
Module Ve	ersion Description	
r_bsp 2.	70 Board Support Packages.	Click here.
r_byteq 1.	20 Byte-based circular buffer library.	
r_cmtw_rx 1.	00 CMTW Driver	
r_cmt_rx 2.	40 Simple CMT driver for creating timer tick.	
r_dac_rx 2.	10 Digital-to-Analog Converter driver	
r_dmaca_rx 1.	01 DMACA driver	
r_dtc_rx 2.	01 DTC driver	
r eeprom spi 2.	30 Renesas R1EX25xxx series Serial EEPROM control software RX Driver Pac	
Details Dependency: r_bsp ver This module creates a	sion(s) 2.70 timer tick using a CMT channel based on a frequency input by the user.	
	· · · · ·	
?	Finish Cancel	



### 3. Click **OK** with changing anything.

Project Updated	×		
	_		
Paths for C Includes and Source Locations have been updated			
Press OK to continue			
OK Cancel		(	Click h

### 4. Click **OK** with changing anything.

Properties for led_sample				
	Paths and Symbols		↓ ↓ ↓ ↓	
<ul> <li>Resource</li> <li>Builders</li> <li>C/C++ Build</li> </ul>	Configuration: Hardware	Debug [Active]	Manage Configurations	
▲ C/C++ General ▷ Code Analysis				
Documentation File Types	🕒 Includes 🗰 Symbol	Is 😕 Source Location 🖹 References		
Formatter	Languages	Include directories	Add	
Indexer	GNU C	S{TCINSTALL}/include	E-dit	
Language Mappings	GNU C++	/\${ProjName}/r_bsp	Edit	
Paths and Symbols	Assembly	/\${ProjName}/r_config	Delete	
Preprocessor Include Pa Droject Paferon.cor		/\${ProjName}/r_cmt_rx		
Run/Debug Settings		/\${ProjName}/r_cmt_rx/src	Export	
Task Repository				
			Move Up	
			Move Down	
	(i) "Preprocessor Includ	e Paths. Macros etc." property page may define additional entries		
	Show built-in valuer	,		
	Show built-in values			
	import Settings	* Export Settings		
4 III >			Restore Defaults Apply	
(?)			OK Cancel	— с
Ŭ				-



5. Open **r\_bsp/board/rskrx64m** from the e<sup>2</sup> studio Project Explorer, select the two files **r\_bsp\_config\_reference.h** and **r\_bsp\_interrupt\_config\_reference.h**, and click **Copy** on the **Edit** menu.



6. Select r\_config from the e<sup>2</sup> studio **Project Explorer** and click **Paste** on the **Edit** menu.



7. Change the names of the two copied files to **r\_bsp\_config.h** and **r\_bsp\_interrupt\_config.h**. That is, delete the "\_reference\_" part of the file names.





 Modify platform.h to correspond to the target board used. Double click r\_bsp/platform.h from the e<sup>2</sup> studio Project Explorer and, in the editor, remove the comment from the include line for the r\_bsp.h file for the RSKRX64M.

🎦 Project Explorer 🛛 📃 🗖	h *platform.h ⊠		
<ul> <li>☐ 4</li></ul>	7 129 130 131 132 133 134 135 136	<pre></pre>	
▷ ▷ mcu ▷ latform.h □ readme.bt ▷ CP r cmt rx	137 137 138 139 140	<pre> //#include './board/rdkrx631/r_bsp.h"</pre>	Double click.
<ul> <li>✓ Control of the section of the secti</li></ul>	141 142 143 144	/* RSKRX64M */ <b>#include "./board/rskrx64m/r_bsp.h"</b> ⊖ /* RSKRX210 */	Remove the commer
lh r_bsp_config.h h r_bsp_interrupt_config.h	145	<pre>//#include "./board/rskrx210/r_bsp.h"</pre>	



### 4.3.4 Create an LED Driving Program

Create a program that toggles the LED0 on/off state every 0.5 seconds using the compare match timer.

Open the file src/led\_sample.c and modify it as shown below.

#### src/led\_sample.c

```
/*
                                                           */
/* FILE
                                                           */
            :Main.c or Main.cpp
/* DATE :Tue, Oct 31, 2006
                                                           */
/* DESCRIPTION :Main Program
                                                           */
                                                           */
/*
  CPU TYPE
           :
/*
                                                           */
/* NOTE: THIS IS A TYPICAL EXAMPLE.
                                                           */
/*
                                                           */
#include "platform.h"
#include "r_cmt_rx_if.h"
/* LED Currently status */
uint32 t ledstatus = LED OFF;
void call back(void *pdata)
{
 if (ledstatus == LED OFF)
 {
   /* Turn ON the LEDO If the status is LED OFF */
 LEDO = LED ON;
 ledstatus = LED ON;
 }
else
 {
   /* Turn OFF the LEDO If the status is LED ON */
 LEDO = LED OFF;
 ledstatus = LED OFF;
 }
}
void main (void)
{
uint32_t cmt_ch;
/* LED0 off */
LEDO =LED OFF;
/* Create of 0.5 second(2Hz) cyclic timer. */
R CMT CreatePeriodic(2, &call back, &cmt ch);
 while(1);
}
```



### 4.3.5 Build and Try Running the Program

Build the program just created and verify that it runs.

#### 1. Click **Build Project** on the **Project** menu.



2. When the build completes, the following will be displayed in **Console** view.

Problems Tasks Console Cons

3. Click **Debug Build** on the **Run** menu.





4. Click **led\_sample HardwareDebug** under the **Renesas GDB Hardware Debugging**. Click the **Debugger** tab and click **Connection Setting**.

Modify **EXTAL Frequency** to be **24.0000** and change **Provide Power from Emulator** to **No**. When these changes have been made, click **Debug**.

e <sup>2</sup> Debug Configurations			×	
Create, manage, and run configurations	s		~	
		) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) (		
🗋 🖹 🗮 🖃 溝 👻	Name: led sample HardwareDebug			
type filter text				
Debug ophy	Main 3 Debugger Startup 4 Source			
C Debug-only	Debug bardware	PSES64MI	Â	Click horo
GDB Simulator Debugging (SH, BL7)	raige be			Click here.
GHS Local C/C++ Launch				
🎒 Renesas GDB Hardware Attach	GDB Settings Connection Settings Debug Tool	Settings		
Renesas GDB Hardware Debugging	▲ Clock	EVE AL		
Ied_sample HardwareDebug	Main Clock Source	24.0000	-	Modify to be
Renesas Simulator Debugging (RX o	Extal Frequency[WHZ]	24.0000		24 0000
	Connection with Target Roard			24.0000.
	Emulator	(Auto)		
	Connection Type	JTag		
	JTag Clock Frequency[MHz]	16.5 +		
	Fine Baud Rate[Mbps]	2.00 -		
	Hot Plug	No 👻		
	⊿ Power			
	Power Target From The Emulator (MAX 200	n No		— Modify to be No.
	Supply Voltage	3.3V 👻		
	▲ CPU Operating Mode			
	Register Setting	Single Chip 👻		
	Mode pin	Single-chip mode 👻		
	⊿ Communication Mode			
	Mode	Debug Mode 👻		
	Execute The User Program After Ending The	el No 👻	-	
Filter matched 9 of 13 items		Apply Revert		
	]			
0				Click here
T		<u>D</u> ebug Close		



5. When the following message is displayed, click **Yes**.



6. When the load module download completes, a **Debug** perspective opens.

e <sup>2</sup> Debug - led_sample/r_bsp/board/rskn64m/resetprg.c - e2 studio			
<u>File Edit Source Refactor Navigate Search Project Run Window H</u> elp			
III ▼    16    10 ▼ ¶ ▼    10     10    10    10    10    10    10    10    10    10    10	R 🙋 🖨 🛷 🔹 🏄 🛓	$ \bullet _{\Sigma} = \langle b   \bullet \bullet$	V Quick Access
<ul> <li>Debug ⊠</li> <li>Let Jample HardwareDebug (Renesas GDB Hardware Debugging)</li> <li>Let Jample Hardware Debugging)</li> <li>Let Jample Hardware Debugging (Let Jample Hardware Debugging)</li> <li>Let Jample Hardware Debugging (Let Jample Hardware Debugging)</li> <li>Let Jample Hardware Debug (Let Jample Hardware Debugging)</li> <li>Let Jample Hardware Debug (Let Jample Hardware Debugging)</li> <li>Let Jample Hardware Debug (Let Jample Hardware Debug (Let Jample Hardware Hardware Debug (Let Jample Hardware Hardwar</li></ul>	00- Variables 33	kpoints 🚻 Registers 🛋 M	lodules ੴ Expressions ● Exentpoints 📑 10 Registers 🖓 🖬 இ 🔩 🕞 🔷 இ 📽 🦄 📑 📽 🔻 Value
<pre> &gt; platform.h</pre>	ion is called. It shou above */	Jd not return.	BEP Outline 122 Project Explorer      Project Explorer      Project Explorer      Proventient      Provent      Proventient      Proventient      Proventient      Proventi
Console ﷺ	nant 👒 Trace 🕥 Visual Exp d6-902dd,902e6-902ed,90 b6-903bd,903c6-903cd,90 76-9127d,91286-9128d,91 56-9135d,91366-9136d,91 56-9135d,91366-9136d,91	pression 🖹 Problems 🕡 E 22f6-902fd,90306-9030d 33d6-903dd,903e6-903ed 1296-9129d,912a6-912ad, 3376-9137d,91386-9138d	ixecutables  Memory  M

7. Click **Restart** on the toolbar. The program will be executed and a break will occur at the start of the main function.

e <sup>2</sup> Debug - led_sample/r_bsp/board/rsknx64m/resetprg.c - e2 studio	
<u>File Edit Source Refactor Navigate Search Project Run Window H</u> elp	
■* ▼	Click here.
🏇 Debug 🔀 🦓 🍇 🦗 🕈 🍇 👔 🛃	
Ied_sample HardwareDebug [Renesas GDB Hardware Debugging]	
a 🙀 led_sample.x	
a 🝿 Thread [1] 1 (No thread info available) (Suspended : Signal : SIGTRAP:Trace/breakpoint trap)	
PowerON_Reset_PC() at resetprg.c:122 0xffc017e8	
gdb	
GDB server	
▶ platform.h	

 After the break at the start of the main function, click **Restart** on the tool bar again. The project will be run and the program will iterate toggling LED0 with a period of 0.5 seconds.



# 5. RX Driver Package Application

# 5.1 RX Driver Package Application Structure

The RX Driver Package Application is a sample application program provided so that users can use the RX Driver Package easily. The RX Driver Package Application consists of an application program that operates using device drivers and middleware included in the RX Driver Package and a project file for building that application. This allows users to start evaluation quickly.



Figure 5.1.1 RX Driver Package Application Structure

Renesas plans to release a variety of types of this RX Driver Package Application in the future, such as system programs that operate using a combination of multiple drivers and middleware and evaluation programs for independent modules from the RX Driver Package.



Figure 5.1.2 Types of RX Driver Package Application

# 5.2 RX Driver Package Application Features

The RX Driver Package Application has the following features.

- The RX Driver Package Application is evaluated in combination with the RX Driver Package.
- Project files are included in the RX Driver Package Application. Since both build and debug configurations for the application project are already set up in the provided project file, the user can quickly begin building and evaluating simply by importing the project into a workspace.
- If e<sup>2</sup> studio is used as the integrated development environment (IDE), the device drivers and middleware used in the application project can be automatically added to the project by using the FIT plugin provided with e<sup>2</sup> studio.
- Renesas provides the RX Driver Package Application without charge.



# 5.3 RX Driver Package Application Usage Example (when e<sup>2</sup> studio is used)

The device drivers and middleware required by the RX Driver Package Application are automatically added to the project by the FIT plugin, which comes with the  $e^2$  studio.

After the project provided with the RX Driver Package Application has been installed in an e<sup>2</sup> studio workspace, the required device drivers and middleware from the RX Driver Package are also installed simply by installing in the project by selecting the RX Driver Package Application with the FIT plugin. Therefore all that remains is to build the project and start evaluation.



Figure 5.3.1 FIT Plugin Automatic Installation

# 5.4 When Using in Combination with an RX Driver Package Application

See the document provided with each RX Driver Package Application for detailed usage methods for that RX Driver Package Application.

# 6. Supplement

### 6.1 M3S-T4-Tiny (TCP/IP protocol stack)

This package include the M3S-T4-Tiny (TCP/IP protocol stack library) of evaluation version. For the commercial version, please go to the below URL.

http://www.renesas.com/mw/t4



# Website and Support

Renesas Electronics Website <u>http://www.renesas.com/</u>

#### Inquiries

http://www.renesas.com/contact/

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



# **Revision History**

s. je.
dule ckaç

# 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 shootthrough 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 type 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

- Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
- 2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
- Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
- Renesses Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesses Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

\*High Quality\*: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shell not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.

- 6. You should use the Reneses Electronics products described in this document within the range specified by Reneses Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Reneses Electronics shall have no liability for malfunctions or damages arising out of the use of Reneses Electronics products beyond such specified ranges.
- 7. Although Renessa 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 maifunctions under certain use conditions. Further, Renessas 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 Renessa Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and maifunction products or system manufactured by vou.
- 8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 9. Renesse Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesse Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesse Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
- 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.

11. This document may not be reproduced or dublicated in any form, in whole or in part, without prior written consent of Reneses Electronics.

12. Please contact a Renease Electronics sales office if you have any questions regarding the information contained in this document or Renease Electronics products, or if you have any other inquiries. (Note 1) "Renease Electronics" as used in this document means Renease 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.



#### SALES OFFICES

#### **Renesas Electronics Corporation**

http://www.renesas.com

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

Renesse Electronics America Inc. 2801 Scott Boulevard Santa Clara, CA 95050-2549, 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 Tei: +1-905-898-6441, Fas: +1-905-898-3220
Renesas Electronics Europe Limited Dukes Meadow, Miliboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K Tei: +44-1628-585-100, Fax: +44-1628-585-900
Renesas Electronics Europe GmbH Arcadiastrasse 10, 40472 Disseldorf, Germany Tei: +49-211-45630-J, Fex: +49-211-4503-1327
Renesas Electronics (China) Co., Ltd. Room 1709, Quantum Plaza, No.27 ZniChunLu Haidian District, Beijing 100191, P.R.China Tei +88-10-8235-1155, Fax +88-10-8235-7679
Renesas Electronics (Shanghai) Co., Ltd. Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333 Tel: +86-21-2226-0888, Fax: +86-21-2226-0999
Reneeas Electronics Hong Kong Limited Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong Tel: +852-2265-6688, Fax: +852 2865-8022/9044
Reness Electronics Talwan Co., Ltd. 13F, No. 363, Fu Shing North Road, Taipel 10543, Talwan Tel: +886-2-8175-9600, Fax: +886 2-8175-9670
Renesas Electronics Singapore Pts. Ltd. 80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949 Tel: +65-6213-0200, Fax: +65-6213-0300
Renease Electronics Malaysia Sdn.Bhd. Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jin Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia Tel: +60-3-7955-9390, Fax: +60-3-7955-9510
Renesas Electronics Korea Co., Ltd. 12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea Tei: +822-5563-737, Fex: +822-5565-5141

© 2014 Renesas Electronics Corporation. All rights reserved. Colophon 4.0