

RX Driver Package User's Manual

R01AN2466EJ0100 Rev.1.00 Nov 28, 2014

Introduction

This application note documents the RX113 Group RX Driver Package User's Manual, version 1.00.

The RX Driver Package is a software platform that collects middleware and drivers that support the Firmware Integration Technology (FIT). This package provides an environment for conveniently evaluating the rich set of peripheral functions provided by RX microcontrollers.

The RX Driver Package includes device drivers for the peripheral modules included in RX microcontrollers, middleware developed for RX microcontrollers, a variety of interface modules, and the Board Support Package (BSP). Users can freely combine the modules included in the RX Driver Package construct systems simply by creating applications that using those modules, and thus quickly start the evaluation process.

This document describes the RX Driver Package, the basic structures and features of RX Driver Package applications, and the FIT modules included in this application note.

See the related documents for detailed information on using the RX Driver Package and procedures for evaluating applications that use the RX Driver Package.

In the use to the product, to fit your environment, please careful evaluated.

Target Device

RX113 Group (Renesas Starter Kit RX113)

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 CubeSuite+ Projects (R01AN1826EJ)
- The User's Manual provided with the RX Driver Package Application.



Contents

1.	Overview	3
2.	RX Driver Package	4
3.	Structure of the RX113 Group RX Driver Package	7
4.	Usage Procedures	10
5.	RX Driver Package Application	26
6.	System Structure	28



1. Overview

1.1 Applicability

This document applies to the RX113 Group RX Driver Package, version 1.00.

1.2 RX113 Group RX Driver Package

The RX113 Group RX Driver Package (referred to as "this package" below) is a software platform(framework) that was packaged such as device drivers and middleware for the RX113. It is available for free of charge to everybody.

1.3 Operating Environment

This package runs under the operating environment described below.

Table 1.3.1 Operating Environment

Microcontroller	RX113 Group
Evaluation board	Renesas Starter Kit RX113
Integrated development	e ² 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. RX Driver Package

2.1 RX Driver Package Structure

The RX Driver Package is a software platform that collects the lower level software components required to develop application programs. A wide range of applications can be developed using the RX Driver Package.

The RX Driver Package consists of the BSP module, microcontroller peripheral module device drivers, middleware (TCP/IP, file systems, and other items), and various interfaces.



Figure 2.1.2 RX Driver Package Structure

For example, as shown in the figure below, a system can be constructed by collecting the required components from those in the RX Driver Package and providing an application program.



Figure 2.1.3 System Construction with the RX Driver Package



Furthermore, all of these modules meet the FIT specifications and since the folder structure and API are standardized (with a few exceptions), this structure makes it easy to swap modules in and out or to port an application to another microcontroller.



Figure 2.1.4 RX Driver Package Application Examples



2.2 RX Driver Package Features

The RX Driver Package has the following features.

- The RX Driver Package is a software platform that provides, in a single package, the device drivers and middleware required for application program development. Renesas will proved RX Driver Packages in a timely as microcontrollers are released.
- Application programs developed on one RX Driver Package can easily be ported to the RX Driver Package for a different microcontroller.
- All the device drivers and middleware included in the RX Driver Package are either compliant with or compatible with the FIT specifications. Thus applications that take advantage of the FIT features can be developed easily.
- The RX Driver Package provides a common interface to middleware and real-time OSes. This makes it easy to reuse software assets and to port applications between real-time OSes.
- Renesas supplies "RX Driver Package Application" sample applications that, when combined with the corresponding RX Driver Package, allow evaluation to be started immediately.
- The RX Driver Package is provided without charge. All the modules in the RX Driver Package can be used without charge, even in commercial applications.



3. Structure of the RX113 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 RX113 Group RX Driver Package



3.2 Module Structure

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



Figure 3.2.1 RX113 Group RX Driver Package FIT Module Structure



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	Clock Generation Circuit (CGC)	r_cgc_rx100	1.30
Device Driver	Data Transfer Controller (DTC)	r_dtc_rx	2.01
Device Driver	Compare match timer (CMT)	r_cmt_rx	2.40
Device Driver	Serial Communications Interface (SCI: Simple I ² C Bus)	r_sci_iic_rx	1.40
Device Driver	I ² C Bus Interface (RIIC)	r_riic_rx	1.50
Device Driver	Serial peripheral interface (RSPI)	r_rspi_smstr_rx	1.07
Device Driver	USB Basic Mini Firmware	r_usb_basic_mini	1.00
Device Driver	USB Host Mass Storage Class	r_usb_hmsc_mini	1.00
Device Driver	USB Host Communication Device Class	r_usb_hcdc_mini	1.00
Device Driver	USB Host Human Interface Device Class	r_usb_hhid_mini	1.00
Device Driver	USB Peripheral Mass Storage Class	r_usb_pmsc_mini	1.00
Device Driver	USB Peripheral Communications Device Class	r_usb_pcdc_mini	1.00
Device Driver	USB Peripheral Human Interface Device Class	r_usb_phid_mini	1.00
Device Driver	IrDA Interface (IrDA)	r_irda_sci_rx	1.00
Device Driver	LCD Controller/Driver (LCDC)	r_lcdc_rx	1.00
Device Driver	Unique ID Read	r_uid_rx	1.00
Middleware	Simple I2C Module for EEPROM Access	r_eeprom_sci_iic_rx	1.30
Middleware	I2C Bus Interface (RIIC) Module for EEPROM Access	r_eeprom_riic_rx	1.40

Table 3.3.1 RX113 Group RX Driver Package FIT Modules



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+ (CubeSuite+), see the document "RX Family Adding Firmware Integration Technology Modules to CubeSuite+ Projects (R01AN1826EJ)" included in this package.

4.1 Environment Used

The RX113 is used as the target microcontroller and the Renesas Starter Kit RX113 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 e² studio

Install the FIT modules in the RX Driver Package into e² studio.

- 1. Decompress the downloaded file an_r01an2466ej0100_rx113_fit.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 e² 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^2 studio **FITModules** folder will be copied to the FIT modules.

ganize 🔻 🔡 Open 🛛 Burn	New	folder		** - 1
Renesas c2_studio DebugComp Drivers eclipse etc HTModules	-	[bsp.v2.70.xm] [r_tsp.v2.70.xm] [r_tgg_rA10_v1.30.xm] [r_tgg_rA10_v1.30.xm] [r_tgg_rA10_v1.30.xm] [r_tgg_rA10_v1.30.xm] [r_tgr_rvv2.40.xm] [r_tgr_rvv2.40.xm] [r_tgr_rvv2.01.xm] [r_tgr_rvv2.01.xm] [r_tgrprom_riic_rvv1.40.xm] [r_tgrprom_riic_rvv1.40.xm] [r_tgrprom_sci_iic_rvv1.40.xm] [r_tgrprom_sci_iic_rvv1.30.xm] [r_tgras_sci_rvv1.00.xm] [r_tida_sci_rvv1.00.xm] [r_tida_sci_rvv1.00.xm]	[] [r_usb_hmsc_mini_v1.00.xml r_usb_hmsc_mini_v1.00.zip r_usb_pcdc_mini_v1.00.xml r_usb_pcdc_mini_v1.00.zip r_usb_phid_mini_v1.00.zip r_usb_phid_mini_v1.00.zip r_usb_phid_mini_v1.00.zip r_usb_pmsc_mini_v1.00.xml r_usb_pmsc_mini_v1.00.xml
internal	-	r_lcdc_rx_v.1.00.xml	r_usb_hhid_mini_v1.00.zip	

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 ²	C/C++ - e2 studio					
Fi	ile Edit Source Refactor Navigate Search	Proje	t Run Window Help			
	New Alt+Shift+N	C	Renesas FIT Module	6	÷	
	Open File	B	C Project			Click here.
	Close Ctrl+W Close All Ctrl+Shift+W		C++ Project Makefile Project with Existing Code Project			
	Save As		Convert to a C/C++ Project (Adds C/C++ Natur Source Folder Folder Source File	e)		
8		6 6 6 6	Header File File from Template Class Task			
e	Print Ctrl+P		Other	Ctrl+N		
22		-				
	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**.

C Project C Project C Project Create C project of selected type	
Project name: led_sample Image: led_sample Image: led_sample Location: C:\sample_workspace\led_sample Image: led_sample	Enter the project name
Project type: Toolchains:	
Executable (Renesas) KPIT GNUARM-RZ-EABI Toolchain Sample Project KPIT GNURL78-ELF Toolchain Static Library (Renesas) KPIT GNURL78-ELF Toolchain Sample Project KPIT GNURL78-ELF Toolchain Debug-Only Project Renesas RXC Toolchain Executable Renesas SHC Toolchain Shared Library Static Library V800 Standalone Static Library (Green Hills) V800 ThreadX Executable (Green Hills)	Click here.
Makefile project Image: Constraint of the state of the st	

6. Select the target. Click the "..." button under **Target Selection** and select **R5F51138AxFP**. After making these settings, click **Next**.

e ² C Project		
e2 studio - Project Generation Select Target Specific Settings		
Toolchain Version : v2.02.00 Debug Hardware: E1 Data endian : Little-endian data Select Target: RSF51138AxFP Select Configurations: Mardware Debug Mardware Debug : Debug using Simulator Debug using simulator Release (no debug) : Project without any debug information Build configurations will be created in the project only for the selected options, however by default the project will be built for the active config (rist configuration selected from group. Based on the device selection yi (RX100) the debug hardware (E1) and debug target (RSF51138AxFP), debu configuration will be automatically created for you.	guration i.e., ou made	Click here and select R5F51138AxFP .
(?) < <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Cancel	Click here.



7. Simply click Next here.



8. Simply click Next here.

e² C	Project		- • • ×	
	tudio - Project Generation ect Additional CPU Options			
	Select Additional CPU Options:			
	Round:	Nearest		
	Precision of Double:	Single precision	•	
	Sign of Char:	Unsigned	•	
	Sign of bit Field:	Unsigned		
	Allocate from Lower Bit	Lower bit	_	
	Width of Divergence of Function:	24 Bit		
	Specify Global Options: Denormalized number allo Replace from int with shor Enum size is made the sma Pack structures, unions and Use try, throw and catch of Use dynamic cast and type	t sllest d classes f C++		
(?) <u>< Back</u>	Next >	ish Concel	Click here.



9. Simply click **Next** here.

e ² 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**.

e C Project	
e2 studio - Project Generation Set various Stack Areas and to add additional Supporting Files	
Stack/Heap Configuration	Clear all the
User's Stack Size: (H') 100	check boxes.
Interrupt Stack Size: (H') 300	
Use Heap Memory Heap Size: (H') 400	
Generation of Supporting Files	
Vector Definition Files Vector Definition Files	
Generate Hardware Setup Function None v	
(?) < Back Next > Finish Cancel	Click here.
	Chok Here.

12. Click **OK**. The project will be generated.

Summary	-	×	<u> </u>
Project Summary:			
PROJECT DIRECTORY : CPU SERIES : CPU TYPE : TOOLCHAIN NAME :	ATOR led_sample C:\sample_workspace\led_samp RX100 RX113 Renesas RXC Toolchain v2.02.00	*	
Main Program C:\sample_workspace\lec Setting of B and R sec	tions I_sample\src\typedefine.h e Ⅲ ►	Ŧ	
	OK Cance]



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.





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 ² C	C/C++ - e2 studio						
File	Edit Source Refactor N	lavigate Search I	rojec	t Run Window Help			
	New	Alt+Shift+N ▶	C	Renesas FIT Module		2	Click here.
	Open File		C+	C Project			
	Close	Ctrl+W		C++ Project			
	Close All	Ctrl+Shift+W	⊡	Makefile Project with Existing Code Project			
	Save Save As	Ctrl+S	C+	Convert to a C/C++ Project (Adds C/ Source Folder	C++ Nature)		
R	Save All Revert	Ctrl+Shift+S		Folder			
_0	Move	53	C h	Source File Header File			
	Rename	F2	Ċ	File from Template			
8	Refresh	F5	¢	Class			
	Convert Line Delimiters To	+	Ċ	Task			
ð	Print	Ctrl+P	Ľ	Other	Ctrl+N		
	Switch Workspace	•					

 Select the created project with Project to Add FIT Module to. Next, select r_bsp and r_cmt_rx in the module list and click Finish.

e ² Add FIT Module			
FIT Modules			
Select FIT Modules to	add to the	related project	
Select III Modules to	audito the	selected project	
Name of the project t	o add FIT n	nodules: led_sample	Select the project.
Family RX 👻	Target Boa	rd RSKRX113 - Function Any -	
	-		
Series RX100 -	Toolchain	Any V Application Any V	
Group RX113 👻		Reset	
Module	Version	Description	
r_bsp	2.70	Board Support Packages.	- Select this item.
r_cgc_rx100	1.30	Configure clock states, outputs, sources and dividers.	
r_cmt_rx	2.40	Simple CMT driver for creating timer tick.	
r_dtc_rx	2.01	DTC driver	
r_eeprom_riic_rx	1.40	EEPROM RIIC Middleware.	
r_eeprom_sci_iic_rx	1.30	EEPROM Simple IIC Middleware.	
r_irda_sci_rx	1.00	IrDA Driver.	
r lcdc rx	1.00	LCD Control Driver.	
Details			
Dependency: r_bsp		.70 k using a CMT channel based on a frequency input by the user.	
?		<u>Einish</u> Cancel	



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



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



5. Click **OK** with changing anything.

e ² Properties for led_sample			- • •	
	Paths and Symbols		(- • - • •	
 ▷ Resource Builders ▷ C/C++ Build △ C/C++ General ▷ Code Analysis Documentation File Types Formatter Indexer Language Mappings Paths and Symbols Preprocessor Include Pi Project References Refactoring History Run/Debug Settings ▷ Task Repository 	Configuration: Hard	wareDebug [Active] nbols 🙆 Source Location 관 References	Manage Configurations	
	Languages GNU C GNU C++ Assembly	Include directories S(TCINSTALL)/include (S(ProjName)/r_bsp (S(ProjName)/r_config (S(ProjName)/r_cmt_rx (S(ProjName)/r_cmt_rx/src	Add Edit Delete Export Move Up Move Down	
4 <u> </u>	 "Preprocessor In Show built-in value Import Settings) 🎼 Export Settings	nal entries	
?			OK Cancel	Click here



6. Open **r_bsp/board/rskrx113** from the e² studio Project Explorer, select **r_bsp_config_reference.h**, and click **Copy** on the **Edit** menu.



7. Select **r_config** from the e² studio **Project Explorer** and click **Paste** on the **Edit** menu.



8. Change the names of **r_bsp_config.h**. That is, delete the "_reference_" part of the file names.





9. Modify platform.h to correspond to the target board used.

Double click **r_bsp/platform.h** from the e^2 studio **Project Explorer** and, in the editor, uncomment from the include line for the r_bsp.h file for the RSKRX113.

Project Explorer 🛛 📄 🔄 岁 🏱 🗖	ⓑ *platform.h ⊠	
⊿ 🞏 led_sample [HardwareDebug]	147	
Binaries	148	→ /* HSBRX21AP */ ///********************************
Includes	149 150	//#include "./board/hsbrx21ap/r_bsp.h"
⊿ 😕 r_bsp	151	⊖ /* RSKRX220 */
> 🔁 board	152	//#include "./board/rskrx220/r_bsp.h"
⊳ 🔁 doc	154	⊖ /* RSKRX111 */
> 👝 mcu	155	<pre>//#include "./board/rskrx111/r_bsp.h"</pre>
⊳ h platform.h	156	Double click.
readme.txt	158	<pre>//#include "./board/rskrx110/r_bsp.h"</pre>
⊳ @ r_cmt_rx	159 160	⊖ /* RPBRX111 */
⊿ (² / ₂ src	161	//#include "./board/rpbrx111/r bsp.h"
Ic led_sample.c	162	
	163	/* RSKRX113 */
HardwareDebug	164 165	#include "./board/rskrx113/r_bsp.h" Uncomment this line.
🔺 🗁 r_config	166	⊖ /* User Board - Define your own board here. */
r_bsp_config.h	167	//#include "./board/user/r_bsp.h"



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 */
  LED0 = LED_ON;
  ledstatus = LED_ON;
 }
 else
 ł
   /* Turn OFF the LED0 If the status is LED_ON */
  LED0 = LED_OFF;
  ledstatus = LED_OFF;
 }
}
void main(void)
ł
uint32_t cmt_ch;
/* LED0 off */
LED0 =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 23 Properties Memory Usage Stack Analysis

 CDT Build Console [led_sample]

 C:\Renesas\E22FCB~\DEBUGC~1\RX\RX_CON~1.EXE led_sample.abs led_sample.x

 Loading input file led_sample.abs

 Parsing the ELF input file.....

 25 segments required LMA fixes

 Converting the OWARF information....

 Constructing the output ELF image....

 Saving the ELF output file led_sample.x

 'Build complete.'

 17:31:13 Build Finished (took 19s.594ms)

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 16.0000.

When these changes have been made, click **Debug**.

e ² Debug Configurations		
Create, manage, and run configura	tions	
	Name: led_sample HardwareDebug	
type filter text	📄 Mair 🌾 Debugger 🕒 Startup 🔲 Common 💱 Source	
c [∞] Debug-only		
C GDB Hardware Debugging	Debug hardware: E1 Target Device: R5F51138	Click here.
💽 GDB Simulator Debugging (S		
GHS Local C/C++ Launch	GDB Settings Connection Settings Debug Tool Settings	
🌮 Renesas GDB Hardware Atta	▲ Clock	
Renesas GDB Hardware Deb	Main Clock Source EXTAL	Modify to be
Ied_sample HardwareDel	Extal Frequency[MHz] 16.0000	
Renesas Simulator Debuggir	Permit Clock Source Change On Writing Interna Yes	16.0000 .
	Connection with Target Board	
	Emulator (Auto)	
	Connection Type Fine Trac Clock Frequency[MHz] 16.5	
	JTag Clock Frequency[MHz] 16.5 Fine Baud Rate[Mbps] 2.00	
	Hot Plug No	
	A Power	
	Power Target From The Emulator (MAX 200mA) Yes	
	Supply Voltage 3.3V	
	▲ CPU Operating Mode	
	Register Setting Single Chip	
<		
Filter matched 9 of 13 items	Apply	
Filter matched 9 of 13 items		
		Click here.
?	Debug	



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



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

and the second se	mple/r_bsp/board/rskrx113/resetprg.c - e2 : e Refactor Navigate Search Project					
		· i# = ::: % · 1 # 2 × 3 • 0	· Q · @ 0 / · [3 4 - 8	Quick Access	1 風C なデバッグ
▲ 🔐 led_sat	le HardwareDebug [Renesas GDB Hardware mple.x [1] read [1] 1 (single core) (Suspended : Signal : PowerON_Reset_PC() at resetprg.c:136 Dxff	SIGTRAP:Trace/breakpoint trap)	00+ Variables 22 Po 8	eakpoints IIII Registers 🛋 Medule Type		
📓 gao	erver		<i>e</i>			
131 132 132 134 135 136 136 137 137 137 137 138 139 140 141 42 ff80e28 144 145 144 145 146 147 146	* Arguments : none * Return value : none boid PowerNN Reset PC(void) { /* Stack pointers are setup p /* Initialise the NCU process = #IF _ REMESAG VERKTON _ >* dwaldn set_intb((unsigned long)_sec Remit = #If (ESC_CFG_USE_CGC_MODULE == 0) /* Gwitch to high-speed opera- operating_frequency.set():	0000 vECT")); top("CSVECT"));		nould not return.	। State (Ha	indwareDobug)
	월 夕スク · 잘 Renesas Coverage · · @ Memory U wareDebug [Renesas GDB Hardware Debugo	age 💿 Performance Analysis 🤮 Profile 👯 Real-tir anol adh	ne Chart 🧕 Trace 👘 Visua	IExpression 团 貝麵 ○ 寅行可能つ	7テイル 🛚 Memory 📕 其 張 隆 🔤 .	
monitor set_io monitor set_io monitor set_io monitor set_io	access width, RW, 1,88600-8860f,8862- access width, RW, 1,88684,88866,88894, access width, RW, 1,880c0-880cd,88100 access width, RW, 1,800c0-880cd,88100 access_width, RW, 1,80029-8002f,80032,	Mg 1900 B8574 J8639 - 88637, 88634, 88636, 88638 - 88639, 88574, 88639, 88634, 8884, 88857, 88854, 8886 8804, 8120 - 8314, 8806 - 88160 8049-812085, 80499 - 8049, 80451 - 80452, 8040 80449 - 8049, 80444, 80479-80158, 80154, 60469, 8160 80449, 80444, 80479-80158, 80154, 60469, 8040 Smart Insert 126 : 1	8890c,8900c,89060-89061 8bl1a,8bl1f-8bl23,8bl25 8c065,8c069-8c06f,8c071 8c166,8c168-8c16e,8c188	,89070-89071,89073-89079,8907 ,8b127,8b129-8b12a,8b12d,8b30 -8c072,8c080-8c086,8c08a-8c08	c,89080,8a000-8a00d,8a0 0-8b30d,8b320-8b333,8c0 b,8c094-8c099,8c09c-8c0	320-8a02d,8a040-8a04d,8a0 900-8c005,8c009-8c00f,8c(99d,8c0a4,8c0c0-8c0c3,8c0

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



 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 e² 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 e² studio.
- Renesas provides the RX Driver Package Application without charge.



5.3 RX Driver Package Application Usage Example (when e² 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 e² 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. System Structure

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



Figure 6.1 System Structure



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

		Description	
Rev.	Date	Page	Summary
1.00	Nov 28, 2014	_	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 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. Renease 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.
- Renesse Electronics does not assume any liability for infringement of patients, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesse Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patients, copyrights or other intellectual property rights of Renesse Electronics or others.
- 4. You should not alter, modify, copy, or otherwise misappropriate any Renease Electronics product, whether in whole or in part. Renease Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renease Electronics product.
- Reneses Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Reneses 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-drime 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 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 which the product is not intended by Renesas Electronics.

- 6. You should use the Renease Electronics products described in this document within the range specified by Renease Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renease Electronics shall have no liability for malfunctions or damages arising out of the use of Renease Electronics beyond such specified ranges.
- 7. Although Renesses Electronics endeevors 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, Renesses 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 Renesses Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and maifunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult please evaluate the seriety of the finel products or system manufactured by you.
- 8. Please contact a Renease Electronics seles office for details as to environmental matters such as the environmental compatibility of each Renease Electronics product. Please use Renease Electronics products in compliance with all applicable laws and regulations that 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.
- 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 duplicated in any form, in whole or in part, without prior written consent of Reneses Electronics.

12. Please contact a Reneeas 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.



SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

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

Renease Electronics America Inc. 2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A. Tel: +1-406-588-6000, Fax: +1-406-588-6130 Renease Electronics Canada Limited 1011 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada Tel: +1-905-498-6441, Fax: +1-905-698-3220 Renease Electronics Europe Limited Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K Tel: +44-1028-585-100, Fax: +44-1028-585-900 Renease Electronics Europe GmbH Arcadiastrasse 10, 40472 Disseldorf, Germany Tel: +49-21-6503-0, Fax: +44-1028-585-900 Renease Electronics (China) Co., Ltd. Room 1709, Quantum Plaza, No. 27 ZhiChunLu Haidian District, Beijing 100191, P.R.China Tel: +49-21-2325-1155, Fax: +86-10-2325-7679 Renease Electronics (Shanghal) Co., Ltd. Niti 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghal, P. R. China 200333 Tel: +48-21-2226-0888, Fax: +862-12226-0999 Renease Electronics (Shanghal) Co., Ltd. Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghal, P. R. China 200333 Tel: +48-21-2226-1058, Fax: +852-2088-6688, Fax: +852-2080-60229/044 Renease Electronics Ing Kong Limited Unit 1001-1613, 16FF, Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong Tel: +852-1235-1155, Fax: +852-12226-0999 Renease Electronics Ingapore Pla. Ltd. 40 Bendemeer Read, Unit 800-7087, Hitkin Innovation Centre, Singapore 339949 Tel: +485-2175-9800, Fax: +862-213-0300 Reneases Electronics Malaysia Sdn.Bhd Unit 908, Block B, Menara Ameorp, Amoorp Trade Centre, No. 18, Jin Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia Tel: +492-393-390, Fax: +802-243-9300, Fax:

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