Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010 Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

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.
- 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. 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; anticrime 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 majorityowned subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

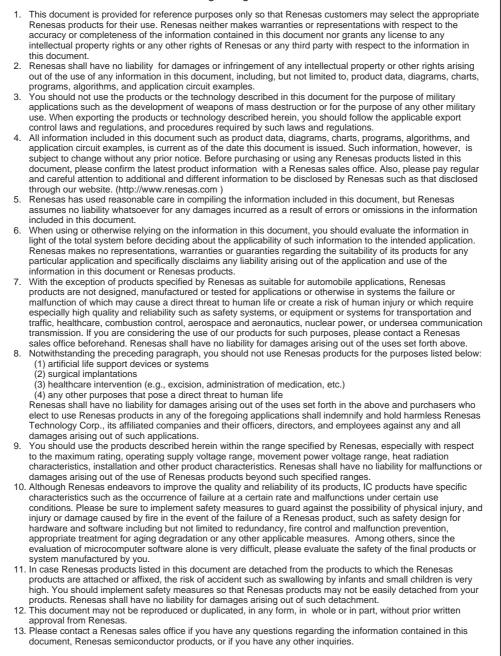


Peripheral Driver Generator V.1.02

Guide Book



Notes regarding these materials



For inquiries about the contents of this document or product, fill in the text file the installer generates in the following directory and email to your local

distributor.

\SUPPORT\Product-name\SUPPORT.TXT

Renesas Tools Homepage

http://www.renesas.com/en/tools

Preface

This manual provides detailed examples of operating the Peripheral Driver Generator (hereafter referred to as PDG). For information on how to operate the PDG or the High-performance Embedded Workshop (hereafter referred to as HEW), refer to the user's manual of the PDG and HEW.

Contents

1.	Over	<i>v</i> iew	1
2.	Usinę	the PDG	2
	2.1	Creating a Workspace with the HEW [HEW]	
	2.2	Creating a Project with the PDG [PDG]	
	2.3	Setting the Peripheral I/O Modules [PDG]	
	2.4	Registering the Output C Source files [PDG]	7
	2.5	Viewing the Created Functions [PDG]	
	2.6	Creating the Application [HEW]	
	2.7	Compile/Link [HEW]	
	2.8	Execution [HEW]	
3.	Settir	ng a Build	
	3.1	Specifying the Header File	
	3.2	Specifying Libraries	
	3.2.1	List of Libraries for Sample Projects	
	3.2.2	When Using HEW V.4.02 or Later	
	3.2.3	When Using Earlier Version than HEW V.4.02	
	3.2.4	Exculding Interrupt Vector Table	
4.	Exan	ple of Creating an Application	24
	4.1	Flow Chart of an Application to be created	
	4.2	Setting Peripherals with the PDG	
	4.2.1	Creating a Project	
	4.2.2	Setting Clocks	
	4.2.3	Setting Timer Mode	
	4.3	Creating a Program	
	4.3.1	Creating a Workspace	
	4.3.2	Creating a Program	
	4.3.3	Sample Programs	
	4.4	Build Work	
	4.4.1	Registering the Generated Filies	
	4.4.2	Setting Compile Options	
	4.4.3	Setting Link Options	
	4.4.4	Excluding Interrupt Vector Files	

1. Overview

This section gives an overview of developing applications with the PDG.

The PDG generates C source files including functions that reflect selected peripherals and settings.

Applications are developed by calling the functions generated by the PDG.

Basically, the development of an application is performed through the following steps:

(1) Setting peripheral I/O modules

Using the PDG, you will create a project, select a CPU group and peripherals, and generate necessary files.

(2) Creating an environment for build/debug

Using the HEW, you will create a workspace for developing the application. Selecting [Create a new project workspace] or other similar menu items allows you to create a workspace for developing the application.

The directory in which the PDG is installed contains sample workspaces separately for M16C/28, H8/3687, and R8C/13.

(3) Creating an application

You will call the functions generated by the PDG.

These functions must be called in the right places of the application.

For example, call the following function to initialize a timer mode with timer A0.

_CreateTimer_TA0_p1();

The header file generated simultaneously by the PDG must be included.

Name of the generated header file: <project name>.h

(4) Registering the source files generated by the PDG

Calling functions generated by the PDG is not the only step necessary for performing a build (or link) without errors.

It is required that the C source files containing the functions' bodies be registered in the workspace opened in the HEW.

(5) Build

You will register the options necessary for a build in the HEW build option. The options to be set are as follows:

- Path settings for include files with the -I compile option
- Specifying API libraries with the -L link option or other options.
- These options are already set in the attached sample workspaces.

Other workspaces require these options.

(6) Debug and evaluation

The development of the application is completed by debugging and evaluating the built application.

2. Using the PDG

This section shows a procedure for creating objects by using a PDG sample project and a HEW sample workspace.

Note that the titles marked with [PDG] and [HEW] describe the operation of the PDG and HEW, respectively.

* Backup copies of the attached sample project file originals are stored in sample.bak under the PDG install directory

(When recovering the files, copy the directories under sample.bak into the sample directory).

2.1 Creating a Workspace with the HEW [HEW]

In this section, you use an empty workspace for the HEW, which is included in the PDG package.

Select [Start] -> [Renesas] -> [Peripheral Driver Generator] -> [Sample Projects]. Then, open any one of the followings:

- Sample workspace for H8/3687
- Sample workspace for M16C/28
- Sample workspace for R8C/13

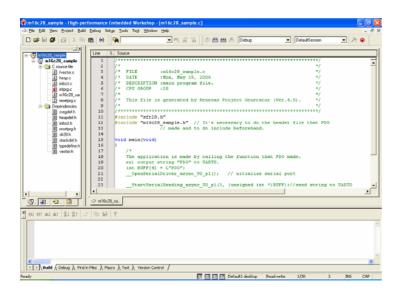
Here, the sample workspace for M16C/28 is used.



[1]: HEW sample workspace for M16C/28.

The HEW will be launched after [1] is selected. Proceed to the next step without closing the HEW.

Since this smaple workspace has been created with HEW V.4.00, a message dialog box asking whether to update or not will appear when the workspace is opened with a later version of the HEW. In this case, click [OK] to open it.



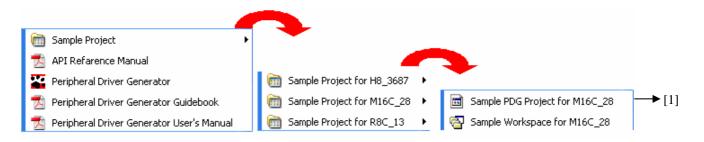
2.2 Creating a Project with the PDG [PDG]

In this section, you use an empty project for the PDG, which is included in the PDG package.

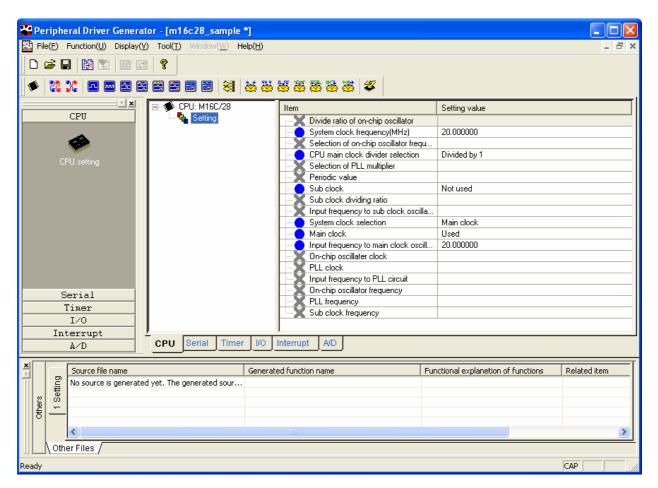
Select [Start] -> [Renesas] -> [Peripheral Driver Generator] -> [Sample Projects]. Then, open any one of the followings:

- Sample PDG project for H8/3687
- Sample PDG project for M16C/28
- Sample PDG project for R8C/13

Here, the sample PDG project for M16C/28 is used.



The PDG will be launched after [1] is selected.



2.3 Setting the Peripheral I/O Modules [PDG]

The first thing to do here is to set CPU clocks.

Select **CPU setting** from the peripheral I/O window.

Peripheral Driver Generator - [m16c28_sample *]			
File(E) Function(U) Display(V) Tool(T) Window(W) Help(H)		-	Ξ×
	10 10 10 10 10 10 10 10 10 10 10 10 10 1		
CPU setting	Item System clock frequency(MHz) Selection of on-chip oscillator System clock frequency(MHz) Selection of on-chip oscillator frequ. CPU main clock divider selection Selection of PLL multiplier Periodic value Sub clock Sub clock dividing ratio Input frequency to sub clock oscilla. System clock selection Main clock Input frequency to main clock oscilla. On-chip oscillator requency IL clock Input frequency to PLL circuit On-chip oscillator requency	Setting value 20.000000 Divided by 1 Not used Main clock Used 20.000000	
Serial Timer	PLL frequency Sub clock frequency		
I/O Interrupt À/D CPU Serial Timer VO	Interrupt A/D	·	
No source is generated yet. The generated sour	ed function name Fur	ctional explanetion of functions Related item	
Ready		CAP	//

CPU clock setting		X
System clock selection:	Main clock 🔹	[
System clock frequency:	20.000000	MHz
r Main clock	,	11112
☑ Use as peripheral function clock s	ource	
Input frequency to main clock circuit:	20.000000	MHz
On chip oscillator clock		
Use as peripheral function clock s	ource	
Frequency selection:		
Periodic value:	0	
Divider selection:		
On-chip oscillator frequency:	0.000000	MHz
- Pli clock		
Use as peripheral function clock s	ource	
Input frequency to PLL circuit:	10.000000	MHz
Selection of multiplication:		
PLL frequency:	0.000000	MHz
Sub clock		
Use as peripheral function clock s	ource	
Input frequency to sub clock circuit:	32.768000	MHz
Sub clock divider		
Sub clock	0.000000	MHz
CPU main clock divider	Divided by 1	
Internal period:	50.000000	ns
	Modify setting Ca	ncel

Peripheral Driver Generator

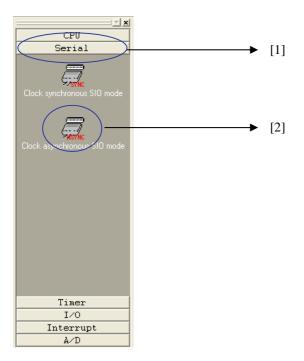
In the [CPU clock setting] dialog box, perform settings according to the program to be created and click [Modify setting] to complete the settings.

The next thing to do is to set the peripherals.

According to the program to be created, first select a peripheral and then a function of the peripheral.

For example, in order to create an asynchronous serial comunication program, follow the steps bellow:

- [1] First, select the [Serial] tab in the peripheral setup window.
- [2] Then, select [Clock asynchronous SIO mode].



Peripheral Driver Generator

In the [Clock asynchronous SIO mode setting] dialog box, specify the items (bit number, parity bit, baud rate, etc,) and select a serial port (UART0, UART1, etc,).

Check [Generate batch source (\underline{M})] and click [Setting].

C source files reflecting these settings will be generated.

* C source files cannot be generated without specifing a serial port.

Clock asynchronous SIO mode setting			
Serial port: UART0	×	BRG register BRG register setting value: 129	
Bit number:	8 bit 💌		
Stop bit:	One stop bit	BRG count source: f1	
Parity bit:	Parity disable 💌	Baud rate: 9600 bps Set details	
Clock selection:	Internal clock	Interrupt enable	
Clock polarity selection:		Transmit interruption level: 0	
LSB first, MSB first selection:	LSB first		
Reverse data logic:		Receive interruption level: 0	
CTS/RTS function: Do not use CTS/RTS function		Permit SI/O interruption	
Noise canceller:		SI/O interruption level:	
Notification function name:	Notification function name: SerialNotificationFunc		
Transmit-Receive pins select:			
Clock pin select:			
	I⊄ Gen	erate batch source[M] Setting Cancel	

2.4 Registering the Output C Source files [PDG]

It is required that the C source files generated in section 2.3, Setting the Peripheral I/O Modules, be registered in the HEW.

This step is necessary for compile and link processes.

In order to register the files, select [Register file(s) in HEW project] from the PDG tool menu.

🌄 Peripheral Driver Generato	r - [m16c28_sample *]		
File(F) Function(U) Display(V)	Tool(T) Window(W) Help(H)		_ 8 ×
0 🛩 🖬 🔛 🎦 🖻 🕅	Setting(S)	-	
🔹 🗈 🖪 🖉	Option(0)	- 🦉 🗳	
X	Register file(s) in HEW project(R)	-	Setting value
CPU	Display output function list(D)		8 bit
Serial	Place output function list in the latest state(P)	ster setting value	129
	E- Clock c	plarity selection	
		e data logic	
STINC	CTS/R	TS function	Do not use CTS/RTS fun
Clock synchronous SIO mode	LSB firs	t, MSB first selection	LSB first
	Parity b	it	Parity disable
	Stop bi		One stop bit
to and the		election	Internal clock
Clock asynchronous SIO mode		punt source	f1
		tion function name	SerialNotificationFunc
		it interrupt	Transmit interrput inhibit
		it interrupt level	0
		e interrupt e interrupt level	Receive interrupt inhibit
	SI/O in		
		terrupt level	
Timer	Baud ra	•	9600
I/O		anceller	
Interrupt			
Å∕D	CPU Serial Timer I/O Interrupt A/D		
Source file name	Generated function name	Functional explanetion of	functions Related item 🔨
C:\Renesas\PDG\sample'	\m16c_28\m16c28_sam BooleanOpenSerialDriv	er_async_U0_p1(void) Open(Initialize) the appoi	nted serial
C:\Renesas\PDG\sample	\m16c_28\m16c28_sam BooleanCloseSerialDriv		II/F
	\m16c_28\m16c28_sam BooleanConfigSerialDri		
C:\Renesas\PDG\sample	\m16c_28\m16c28_sam BooleanSetSerialForma \m16c_28\m16c28_sam BooleanSetSerialInterr		Bit number, । 🚦
C, r/cei lesas (PD/d)sample	miloc_zomiloczo_sam Dooleanbecbenamilen	apc_async_oo_pr(void) bec up senail interrupt	
Clock asynchronous SIO n	node /		

Note:

- Once the source files generated by the PDG are registered in the HEW, you cannot cancel their registration via the PDG. Cancel the registration via the HEW, if necessay.
- Make sure that the CPU group selected for the HEW workspace is the same as for the PDG project before registering the source files.

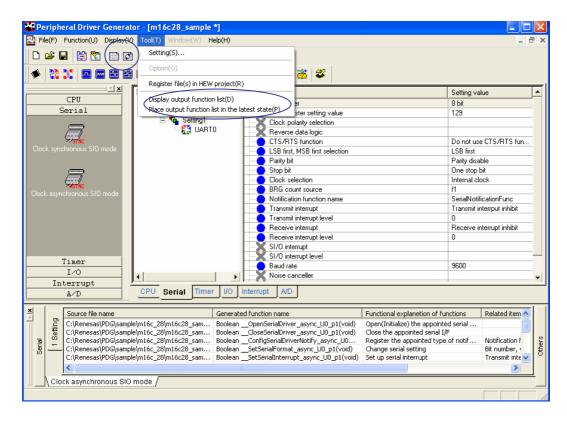
2.5 Viewing the Created Functions [PDG]

The application is created by calling the functions created in section 2.3, Setting the Peripheral I/O Modules. Available functions are listed in the source display window.

🌄 Peripheral Driver Genera	tor - [m16c28_sample *]			
File(F) Function(U) Display(V) Tool(T) Window(<u>W</u>) Help(H)			-
C ≥ R M M ■ C ◆ 22 X ■ ■ 25 E		u a a a a a a a a a a a a a a a a a a a		
	E-2 Serial: M16C/28	Item	Sett	ing value 🔺
CPU	Clock synchronous SIO (Bit number	8 bi	
Serial	🖻 💦 Clock asynchronous SID	BRG register setting value	129	
	🖻 🔽 Setting1	Clock polarity selection		
	Starto	🛛 🗙 Reverse data logic		
STNC		CTS/RTS function	Do	not use CTS/RTS fun
Clock synchronous SIO mode		LSB first, MSB first selection	LSB	first
		Parity bit	Pari	ty disable
		🔵 Stop bit		stop bit
6		Clock selection		mal clock
Clock asynchronous SIO mode		BRG count source	<u>f1</u>	
		Notification function name		alNotificationFunc
		Transmit interrupt		nsmit interrput inhibit
		Transmit interrupt level	0	1.1.1
		Receive interrupt		eive interrupt inhibit
		Receive interrupt level	0	
		SI/O interrupt SI/O interrupt level		
Timer		Baud rate	960	· · · · · · · · · · · · · · · · · · ·
I/0		Noise canceller	360	J
Interrupt		Noise canceller		▼
A/D	CPU Serial Timer 1/0 1	nterrupt A/D		
Source file name		d function name	unctional explanetion of function	s Related item 🔨
C:\Renesas\PDG\samp C:Renesas\PDG\samp C:\Renesas\PDG\samp C:\Renesas\PDG\samp C:\Renesas\PDG\samp C:\Renesas\PDG\samp	le\m16c_28\m16c28_sam Boolean le\m16c_28\m16c28_sam Boolean le\m16c_28\m16c28_sam Boolean le\m16c_28\m16c28_sam Boolean	_CloseSerialDriver_async_U0_p1(void) C _ConfigSerialDriverNotify_async_U0 R _SetSerialFormat_async_U0_p1(void) C	pen(Initialize) the appointed seri lose the appointed serial JF egister the appointed type of no hange serial setting et up serial interrupt	

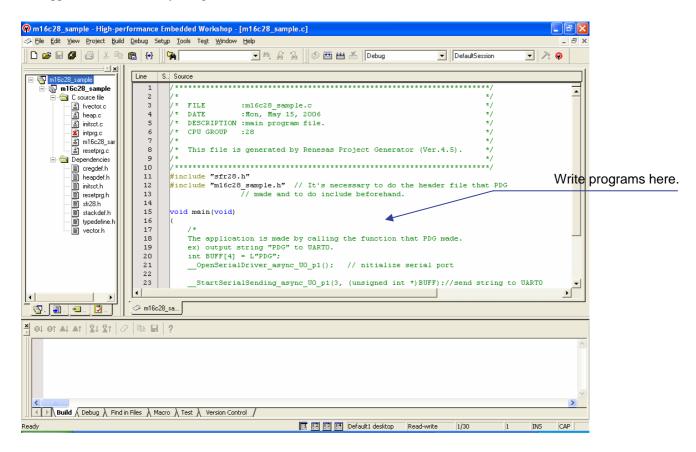
Peripheral Driver Generator

To list output functions in CSV file format, select [Display output function list] from the tool menu. The application associated with the *.csv extension will be launched.



2.6 Creating the Application [HEW]

The application is created by using the HEW.



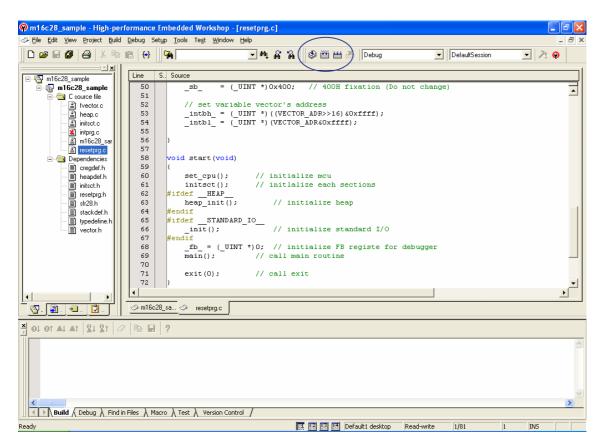
For example, to send data to the UART0 serial port, call the following function:

__StartSerialSending_async_U0_p1()

if(__StartSerialSending_async_U0_p1(15,(unsigned int *)"hello PDG World") == 0) printf("False\n"); //Failed to send

2.7 Compile/Link [HEW]

Perform a build after creating the application by clicking the build button on the HEW.



Note:

(1) When a build is performed while the HEW is installed in other than the C drive, errors will occur during the link operation. In this case, specify the directory containing the API location (lib\M16C_Tiny, lib\H8_Tiny, or lib\R8C_Tiny under the PDG installation directory) by using link options.

For details on how to specify the API library, refer to section 3.2, Specifying Libraries.

(2) When the M3T-NC30WA V.5.40 Release 00 compiler package for M16C series is used, errors may occur during updating dependency information. Use version 5.42 or another method of specifying libraries.

All the development work is completed.

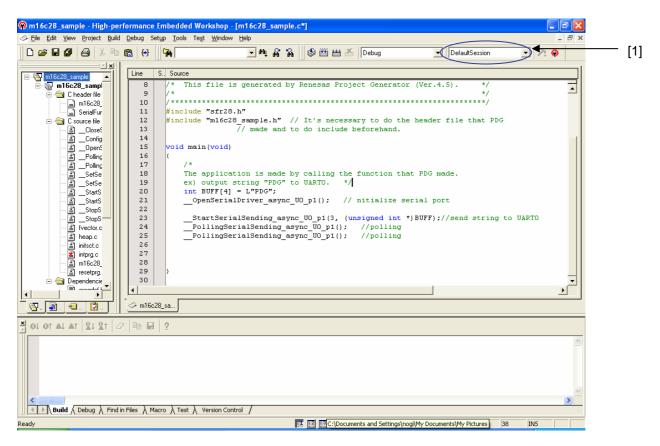
2.8 Execution [HEW]

This section explains how to use the simulator to evaluate the application after the development work. The following preparations for executing the application are required as explained so far:

(1) Uncomment the following lines written in the main function of the sample workspace to enable the program.

<pre>int BUFF[4] = L"PDG"; OpenSerialDriver_async_U0_p1();</pre>	//Initializes UART0
<pre>StartSerialSending_async_U0_p1(3, (unsigned int *)BUFF);</pre>	//Sends to UART0
PollingSerialSending_async_U0_p1();	//Polling processing
PollingSerialSending_async_U0_p1();	//Polling processing

- (2) Use the PDG to register the generated source files in the HEW.
- (3) Perform a build.
- Executing the simulator



- Select [SessionM16C_R8C_Simulator] for session.
 Select [Yes].

High-pe	erformance Embedded Workshop 🛛 🛛 🔀
⚠	Session "DefaultSession" in project "m16c28_sample" has been modified. Do you want to save it?
	<u>Y</u> es <u>N</u> o Cancel

[3] Select [OK].

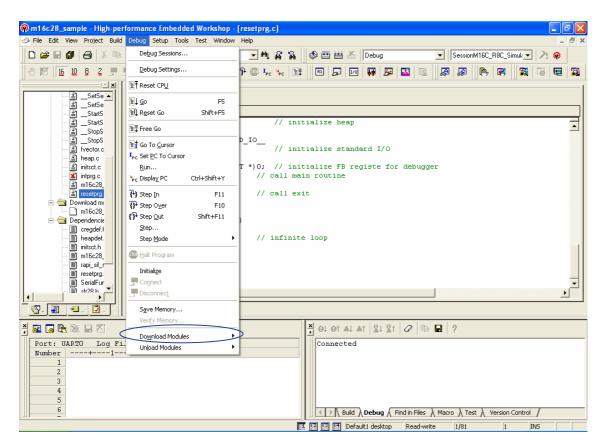
Init (M16C R8C Simulator)	×
MCU Debugging Information Script Trace 1/0 Script	_
MCU: M16c2x.mcu Refer	
Time Count Resource	
MCU Clock: 10 MHz / 1	
OK Cancel Help	
🔽 Do not show this dialog box ag	ain.

Peripheral Driver Generator

[4] Select [Download Modules] to download the following file.

C:\renesas\PDG\sample\m16c_28\m16c28_sample\debug\m16c28_sample.x30

(This file may automatically be downloaded after a build by following the steps [1] to [3].)



[5] Click the reset button.

🖗 m16c28_sample - High-performance Embedded Workshop - [m16c28_sample.]
A File Edit View Project Build Debug Setup Tools Test Window Help	_ 8 ×
🗋 😅 🛃 🛃 🔏 🕹 💼 💼 🖶 🙀 🏹 💽 🛝 🛃 🙀	🕸 🕮 📇 🚺 Debug 💽 SessionM16C_R8C_Simul 🔍 🗡 🏟
🗍 🗢 🕅 🔟 📴 🙎 💆 🗯 📫 🌘 計 🔃 🗄 Ər 🕀 🚥 🛼 🦕 🖹	
 Johns Johns	2006 */ file. */ file. */ */ Renesas Project Generator (Ver.4.5). */ */ */ Tt's necessary to do the header file that PDG do include beforehand. * calling the function that PDG made.
X Image: Constraint of the second s	Sol of Al At 21 21 0 10 10 20 Connected Build \ Debug \ Find in Files \ Macro \ Test \ Version Control / Build \ Debug \ Find in Files \ Macro \ Test \ Version Control / Example Default destop Read-write 19/30 38 INS

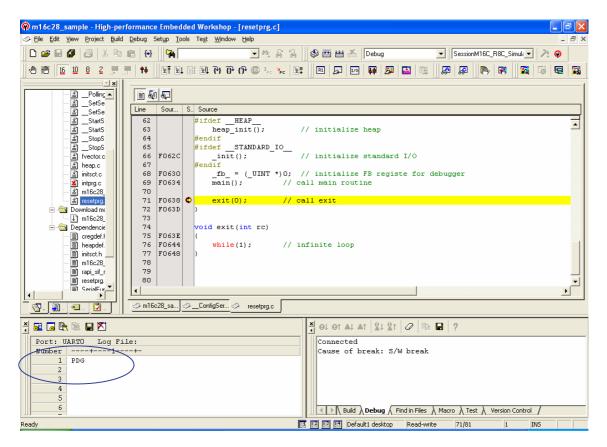
[6] Set a breakpoint at the place where exit(0) is called in resetprg.c.

fb = (_UINT *)0;	// initialize FB registe for debugger	
main();	// call main routine	
exit(0);	// call exit	

[7] Click the run button.

🏟 m16c28_sample - High-performance Embedded Workshop - [m16c28_sample.c]
😓 Elle Edit View Project Build Debug Setup Iools Tegt Window Help 💷 🗗 🛪
🗋 🖸 🖉 🖉 🕹 🐘 💼 😥 🙀 🤜 💌 📩 🖄 🖉 🖄 🔛 🖄 🚱 🕮 🔠 🗡 Debug 🔍 SessionM16C_R8C_Simult I 🥕 🍂 🐢
Image: Construction of the second
X G G K K J X Port: UARTO Log File: Number+ 1 2 3 4 5 6 ++ 1 2 3 4 5 6 ++ 1 2 3 4 5 6 +
Ready III Default1 desktop Read-write 19/30 38 INS

[8] Make sure that "PDG" is displayed.



The operation using the samples is completed.

3. Setting a Build

In order to perform a build/compile with the HEW, the following settings are required as well as the steps for registering the generated source files as explained in section 2.

- Specifying the directory of the reference header file (using -I option)
- Setting for linking API libraries (using -L option) *
 - * When the High-performance Embedded Workshop V.4.02 or earlier is used.

This section explains how to perform these settings.

Note that these are already set in the sample workspaces attached to the Peripheral Drive Generator package.

3.1 Specifying the Header File

To call the functions generated by the PDG, the header file containing the functions' prototype declarations must be included.

A header file name is <project name>.h.

When the directory created by the HEW is different from the project directory created by the PDG, the directory containing the header file, that is, the directory where the project is created, must be specified by using –I option.

(1) From the build menu, select "Renesas M16C Standard Toolchain..." for M16C/Tiny and R8C/Tiny, or select "H8S,H8/300 Standard Toolchain..." for H8/300H Tiny.

H8S,H8/300 Standard Toolchai	n 🥐 🔀	
Configuration : Debug All Loaded Projects Burger C Source file C++ source file Linkage symbol file	C/C++ Assembly Link/Library Standard Library CPU Category: Source Show entries for: Include file directories Add Add Insert Bemove Move.gp Move.govn	
	Options C/C++ : -cpu=300HN -object=''\$(CONFIGDIR)\\$(FILELEAF).obj'' -debug -nolist -chgincpath -nologo	

(2) Select the [C/C++] tab ([C] tab for M16C/Tiny and R8C/Tiny).
When H8/300H Tiny is selected, select [Include file directories] for [Show entries for:].
When M16C/Tiny or R8C/Tiny is selected, select [Inlcude file directories] for [Show Entries For:].

[1] Click [Add...].

(3) Specify the directories containing include files.

Add include file directory		? 🔀
<u>R</u> elative to :		OK
Custom directory	•	Cancel
Directory :		
		<u>B</u> rowse

In the [Add include file directory] dialog box, select the relative path category and enter the directory name. For H8/300H Tiny, select and enter the followings:

- Relative to: HEW installation directory
- Sub-Directory: Directory name

For M16C/Tiny or R8C/Tiny, select and enter the followings:

- Relative to: Custom directory
- Directory: Directory name
- (4) Close all the dialog boxes to complete the settings.

3.2 Specifying Libraries

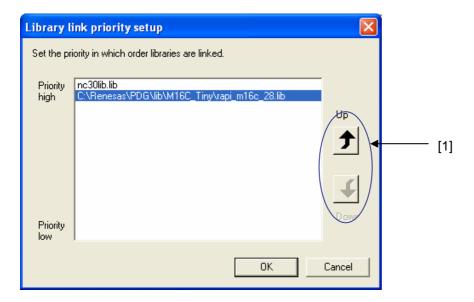
3.2.1 List of Libraries for Sample Projects

When a setting source of each peripheral I/O module is built, the libraries shown below must be linked. For the names of all library files of supported microcomputers, refer to the PDG's users manual.

CPU	Directory	Library File Name
<u>H8/3687</u>	lib\h8 3687	rapi h8 3687.lib
<u>R8C/13</u>	lib\r8c 13	rapi r8c 13.lib
M16C/28	lib\m16c 28	rapi m16c 28.lib

3.2.2 When Using HEW V.4.02 or Later

When the source is registered in the PDG while HEW V.4.02 or later is used, the following dialog box is displayed.



In this dialog box, determine the priorities of the libraries.

When the same symbol name exists in different libraries, the library that has the highest priority among them are selected.

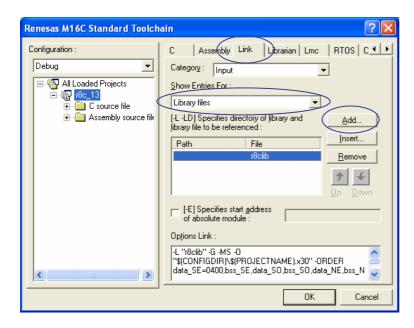
When changing the priorities, select a library and click the buttons marked with [1]. A library with higher priority is listed higher.

3.2.3 When Using Earlier Version than HEW V.4.02

- When using an earlier HEW version than 4.02, follow the steps below to specify libraries.
- From the build menu, select [Renesas M16C Standard Toolchain...] for M16C/Tiny and R8C/Tiny, or select [H8S,H8/300 Standard Toolchain...] for H8/300H Tiny.

🖗 r8c_13 - High-performance Embedded Workshop			
File Edit View Project Build Debug Setup Tools Test Window Help			
Renesas M16C Standard Toolchain	💌 🐴 🖓 🎒 🕮 🛗 👗 🛛 Debug	DefaultSession	- 7. 🛛
Build Eile Ctrl+F7			
E - 2 13 ∰ Build F7			
Build Multiple			
Serial Clean Current Project			
Update All Dependencies			
De Include/Exclude Build			
En. Build Phases			
실Op 말Pol Build ⊆onfigurations			
PolLink_age Order			
Sel Congrate Makafile			
State			
A DI DT AL AT 21 21 0 m H ?			
Build (Debug) Find in Files) Macro) Test) Version Control	1		>
The Long V read V running V radio V rost V version control	_/ III III III Default1 desktop		INS CAP

(2) Select the [Link] tab in the [Renesas M16C Standard Toolchain] dialog box, or select the [Optimization linker] tab in the [H8S,H8/300 Standard Toolchain] dialog box.



Select [Library files] ([Library files] for H8/300H Tiny) for [Show Entries For] and click [Add].

(3) Specify the directory containing the API libraries.

Re_13 - High-performance Embedded	Workshop			a a x
File Edit View Project Build Debug Setup T	ools Test Window Help			
	A	. 옮 옮 🗍 🧼 🕮 🛗 👗 Debug	DefaultSession	
Cource file Cource file Cource file Cource file Cource file Conseseration file Conseseratin file Conseseration file Conseseratin file Conseser	nesas M16C Standard Toolch: onliguration : Debug Call Loaded Projects Call Caded Call Caded Projects Caded Call Caded Caded Projects Caded Caded Cad	C Assembly Link Librarian Lmc R Category: Input Show Entries For : Library files (4. LD) Specifies directory of Jibrary and Jibrary file to be referenced : Path File	Add	[1]
A OI OT AL AT 21 21 0 0 0 E		OK	Cancel	
Build & Debug & Find in Files & Macro	o 入 Test 入 Version Control /			
		ET EZ EZ EA Default1 decktop		TAIC

[1] Select [Library directory] (for R8C/Tiny, M16C/Tiny, and M16C/60).

	Set library file directory	? 🛛
	Relative to :	ОК
(Custom directory	Cancel
	Directory :	
		<u>B</u> rowse

Relative to: Select [Custom directory].

Directory: Specify the directory containing the API libraries.

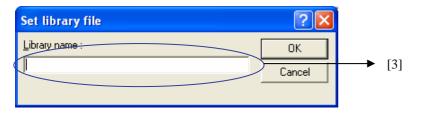
Example) C:\Renesas\PDG\lib\M16C_28

For H8/300H Tiny, specify the directory name and liblary file name.

(4) Specify the library name (For R8C/Tiny, M16C/Tiny, and M16C/60).

<pre>r8c_13 - High-performance Embedded Workshop</pre>			_ 7 ×
File Edit View Project Build Debug Setup Tools Test Window			
	💌 🐴 🙀 🎽 🏙 🛗 👗 Debug	DefaultSession	
	C Assembly Link Librarian Lmc RTI ciects arce file mbly source file Path File Clipson Link: L'ifediti"-6-MS-0 "\$CONFIGDIR \\$!PROJECTNAME].x30"-ORDEP data_SE-0400.bss_SE.data_S0.bss_S0.data_NE.I	AddLbrary directory ubrary file	
			to the second
	💶 😰 🛐 🖽 Default1 desktop		INS

[2] Select [Library file].



- [3] Enter the name of the peripheral API library file to be used. Example) rapi_m16c28 (Do not add the .lib extension.)
- (5) Close all the dialog boxes to complete the settings.

3.2.4 Exculding Interrupt Vector Table

The PDG creates an interrupt vector simultaniously.

Since another interrupt vector is created when a workspace and a startup file are created at the same time by the HEW, the interrupt vector for the startup needs to be excluded from the compile target.

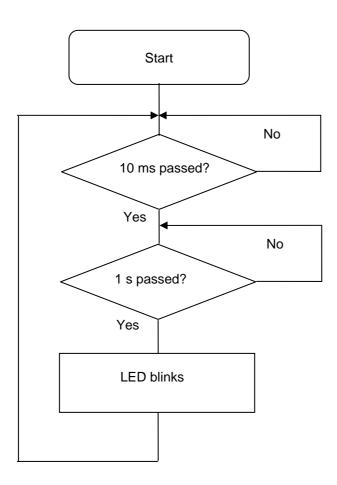
Note that when HEW V.4.02 or later is used, the interrupt vector is automatically excluded in the PDG.

4. Example of Creating an Application

This section describes a procedure for creating an application with the PDG.

4.1 Flow Chart of an Application to be created

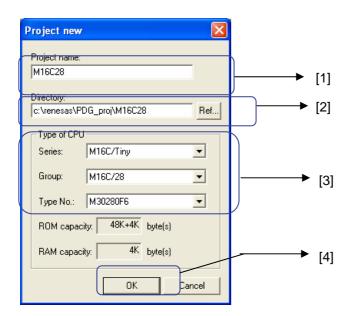
Create an application based on the flowchart below. Microcomputer used for the application is M16C/28.



4.2 Setting Peripherals with the PDG

4.2.1 Creating a Project

Select [File] -> [Create New Project] to open the new project window.



[1] Enter a project name.

Here, "M16C28" is chosen.

Note: The Peripheral Driver Generator creates a header file with "<project name>.h". If an existing file has the same name, enter another project name.

[2] Specify a directory to store the project file in.

By default, a directory with the same name as the project is created under c:\renesas\PDG_proj.

- [3] Select a CPU.
 Here, select the followings:
 Series: M16C/Tiny
 Group: M16C/28
 Type No: M30280F6
- [4] Click the [OK] button to complete creating a project.

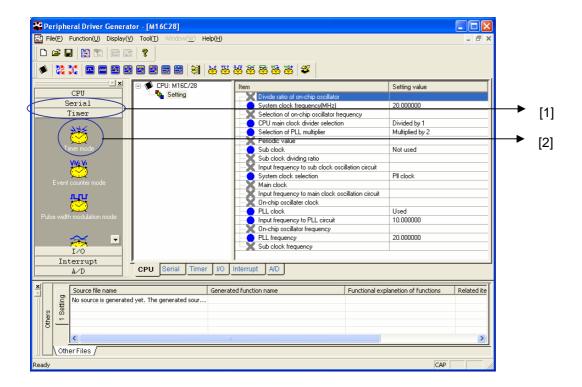
4.2.2 Setting Clocks

The [CPU clock setting] window appears after the steps for creating a project in section 4.2.1 is completed.

U clock setting			×	3
System clock selection:	Pll clock	•)
System clock frequency:		20.000000	MHz	
Main clock				
Use as peripheral function clock so				
Input frequency to main clock circuit:		20.000000	MHz	
- On chip oscillator clock				
Use as peripheral function clock so	urce			
Frequency selection:		<u></u>		
Periodic value:		0		
Divider selection:		_		
On-chip oscillator frequency:		0.000000	MHz	
- Pll clock				
Use as peripheral function clock so	urce		t	H
Input frequency to PLL circuit:		10.000000	MHz	
Selection of multiplication:	Multiplied by 2	•		\mathbf{H}
PLL frequency:		20.000000	MHz	
Sub clock				
Use as peripheral function clock so	urce			
Input frequency to sub clock circuit:		0.032768	MHz	-
Sub clock divider		_		
Sub clock		0.000000	MHz	J
CPU main clock divider	Divided by 1	•)
Internal period:		50.000000	ns	
	Modify set	ting Car	ncel	

- [1] Select [PLL clock] in the [System clock selection].
- [2] Specify the frequency as 10 MHz.
- [3] Select [Multiplied by 2] for [Selection of multiplication].
- [4] Do not use sub-clock.
- [5] Select [Devided by 1] for [CPU main clock divider].
- [6] Click [Modify setting] to complete the settings.

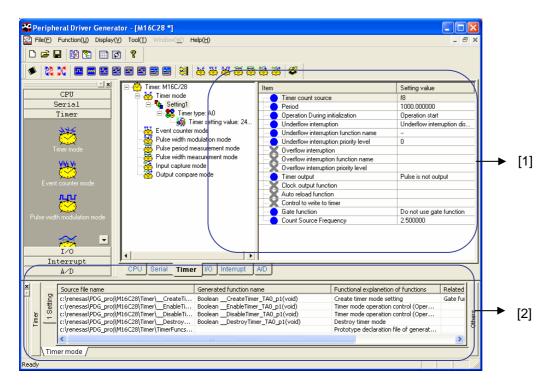
4.2.3 Setting Timer Mode



- [1] Select the [Timer] tab.
- [2] Select and click [Timer mode].

	Timer mode setting		X	
[1] 🗲	Timer type: A0	Operation during initializatio	n: Operation start	→ [4]
	Frequency of count source: 2.500000 MHz	Interruption: 🔲 Enable U	nderflow interruption	
[2] 🗲	Count source: f8	Interruption	priority level:	
[3] 🗲	Period: 1000.000000 microsec.	Interruption function name:	TimerIntFunc	
	Result of calculation	Timer output:	Pulse is not output	→ [5]
	The values in this frame are set. Period: 1000.000000 micro sec.	Clock output function:	··· 💌	
	(Error: 0.000000 %)	Auto reload function:		
	Setting value:	Control to write to timer:	··	
	Timer setting value 2499	Gate function:	Do not use gate function	→ [6]
		Generate batch	source[M] Modify setting Cancel	
		4	•	
		[7]	[8]	

- [1] Select [A0] as the timer.
- [2] Select [f8] for the internal count source.
- [3] Specify the counter value as 10 ms.
- [4] Select [Operation start] for the operation after initialzation.
- [5] Select [No pulse is output] for the timer output.
- [6] Select [Do not use gate function] for the gate function.
- [7] Check [Generate batch source] to generate sources reflecting these settings.
- [8] Click [Setting] to complete the settings.



After the settings of the timer mode are completed, the settings are displayed in [1], and the generated source files, generated function names, and functional explanations are listed in [2].

4.3 Creating a Program

A program is created with the HEW.

4.3.1 Creating a Workspace

Welcome!	? 🛛	
Options: Create a new project workspace Create a new project workspace Create a new project workspace	OK Cancel Administration	▶ [1]
C:\Renesas\PDG\sample\m16c_28\		

[1] In the [Welcome!] dialog box, select [Create a new project workspace] and click [OK].

New Project Workspace		X
Projects		
Project Types	Workspace Name: m16c_timer	• [2]
C source startup Application Empty Application	Project Name: m16c_timer	
Elibrary Debugger only - M16C Simulat	Directory: C:\WorkSpace\m16c_timer Browse	[3]
	CPU family:	▶ [4]
<	Tool chain: Renesas M16C Standard	→ [5]
Properties		
	OK Cancel	Ĩ

[2] Select [C source startup Application].

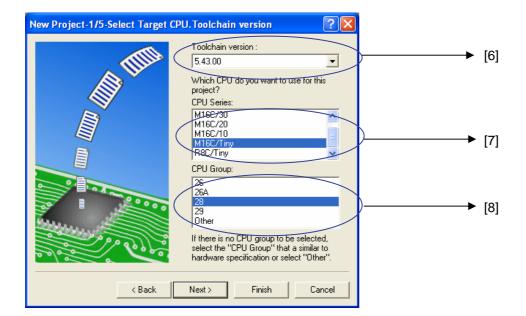
(When using an assembler startup, select [Application].)

[3] Enter a workspace name.

Here, [m16c_timer] is selected.

- [4] Select [M16C] for the CPU family.
- [5] Select [Renesas M16C Standard] for the tool chain.

Click the [OK] button.



- [6] Specify the compiler version as 5.40.00 or later.
- [7] Select [M16C/Tiny] for [CPU Series:].
- [8] Select [28] for [CPU Group].

Click the [Next] button to proceed.

Operate the new project creation wizard to compelet creating the new workspace.

4.3.2 Creating a Program

Code a program on [1] of the HEW.

🖗 m16c_timer - High-performance Embedded Workshop - [m16c_timer.c]	
🗇 Elle Edit View Broject Build Debug Setyp Tools Tegt Window Help 🗧 🖉 🗙	
D 🖆 🖬 🕼 🕹 🖇 📴 🙀 🔄 🖓 🔄 🖓 🔄 🖓 🐨 🏙 🛱 🎽 Debug 💽 DefaultSession 💿 🥕 🐢	
Image: Image	→ [1]
Ă 0i 0t At At 21 2t 2 □ B ?	
Comment Applied A Debug A Find in Files A Macro A Test A Version Control /	
Total And the second se	

When coding a program, use the functions created with the PDG.

🚰 Peripheral Driver Generator - [M16C28 *] File(F) Function(U) Display(V) Tool(T) Help(H) 🗃 🗖 🔛 🔀 8 2 [1] * 22 🗖 🖬 🖬 🗃 🗃 🗃 😽 💥 💥 🖉 🖏 🥳 🌾 r: M16C/28 - x Setting value CPU Timer count sourc Setting1
Setting1
Strimer type: A0
Strimer setting value: 24..
Event counter mode Inner count source Period Operation During initialization Underflow interruption Underflow interruption function nar Underflow interruption priority level Oraciflow interruption priority level Serial 1000.000000 Timer Operation start Underflow interruption dis 245 se width modulation mode 0 Pulse period measurement r Pulse width measurement m nput capture mode Ordentiow interruption plotting feve Diverflow interruption function nan Overflow interruption priority level Timer output Clock output function Auto reload function Control to write to timer Gate function VV V utput compare mode Pulse is not output ιų Gate function Count Source Frequency Do not use gate function 2.500000 \sim I/0 Interrupt ial Timer 1/0 Inte A∕E CPU Se upt A/D Functional explanetion of functions Source file name Generated function name Related Source rune name civrenesas/PDG proj/M16C28\Timer_CreateTi... civrenesas/PDG proj/M16C28\Timer_EnableTi... civrenesas/PDG proj/M16C28\Timer_DisabeTi... civrenesas/PDG_proj/M16C28\Timer\DisabeTi... civrenesas/PDG_proj/M16C28\Timer\TimerFuncs... Boolean _CreateTimer_TA0_p1(void) Boolean _EnableTimer_TA0_p1(void) Boolean _DisableTimer_TA0_p1(void) Boolean _DestroyTimer_TA0_p1(void) Create timer mode setting Timer mode operation control (Oper Timer mode operation control (Oper Detection and control (Oper Setting Gate fu imer Destroy timer mode Prototype declaration file of genera Timer mode eady

The generated functions can be viewed as Excel data as well as in the PDG's graphical user interface.

By clicking the button, Excel is launched and the functions are listed.

Note that since the extension of the output files is .csv, *.csv files must be associated with Excel. (Otherwise, the files are displayed as text files.)

Microsoft Excel - Microsoft Ex		ools <u>D</u> ata <u>Wi</u> ndow <u>H</u> elp Acro <u>b</u> at			_ 므
		🖹 🖻 🝼 🔊 • Ω - 🍓 Σ 🏂 ફੈ↓ 🕌 🚜 100% 🔹 🕄 🖕	Arial • 10 • B /	ॻ ॾ ॾ ॿ छ \$ % , ‰ ∉	
]		••••••••••••••••••••••••••••••••••••••
A25 -					
A25 •	= B	C	D	E	F
	M16C28		U	E	г
2	NITOCZO				
3					
4 Timer					
5 Mode	Setting name	Source file name	Generated function name	Functional explanetion of functions	Related ite
5 Timer mode	Setting1	c:\renesas\PDG proj\M16C28\Timer\ CreateTimer TAD p1.c	Boolean CreateTimer TAD p1(void)		Gate funct
7		c:\renesas\PDG proj\M16C28\Timer\ EnableTimer TAD p1.c	Boolean EnableTimer TAD p1(void)	Timer mode operation control (Operation start)	
3		c:\renesas\PDG_proj\M16C28\Timer_DisableTimer_TAD_p1.c	Boolean DisableTimer TAO p1(void)	Timer mode operation control (Operation stop)	
9		c:\renesas\PDG proj\M16C28\Timer\ DestroyTimer TAD p1.c	Boolean DestroyTimer TAO p1(void)	Destroy timer mode	
0		c:\renesas\PDG_proj\M16C28\Timer\TimerFuncs_Timer_TA0_p1.h		Prototype declaration file of generated function	
1					
2					
3					
4					
5					
6					
7					
8 9 Others					
9 Otners 20 Mode	Setting name	Source file name	Generated function name	Functional explanetion of functions	Related ite
21 -	Setting name	c:\renesas\PDG proj\M16C28\M16C28.h	Generated function name	Header file for all output source files	Related Its
22	Setting	c. wenesas in Do_proj winoczo winoczo.n		rieader me for an output source mes	
23					
24					
5	1				
26					
4)) M16C28			[+]		
eadv	/			NUM	•

The figure above is the displayed Excel data.

4.3.3 Sample Programs

The header file and the functions shown in red are generated by the PDG.

```
#include "sfr28.h"
#include "m16c28.h"
                                  // When the functions generated by the Peripheral Driver Generator are used,
                                  // this header file must be included.
#define _1SCNT (1000/10)
#define PLL_WAIT_1MS 10000 /* 1msec @10MHz */
#define PLL_WAIT_CNT 20
                                  /* 20msec */
void main(void)
{
       int
             counter = _1SCNT;
       int
             onoff = 1;
       unsigned int i,j;
       /* PLL clock setting */
       prcr = 0x01;
                                  /* protect register off */
       cm2 = 0x00;
                                  /* system register2 Initialize */
       cm07 = 0;
       cm1 \&= 0x3f;
       cm06 = 0;
                                  /* 2 multiplying */
       plc0 = 0x11;
                                  /* 2 wait */
       pm20 = 0;
                                  /* PLL operation */
       plc07 = 1;
       for (i = 0; i < PLL_WAIT_CNT; i++) {
                                                      /* about 20ms wait */
             for (j = 0; j < PLL_WAIT_1MS; j++) { /* Main clock 10MHz */
              }
       }
       cm11 = 1;
                                  /* protect register on */
       prcr = 0x00;
                                  /* protect register off */
       prcr = 0x04;
       pacr = 0x03;
                                  /* 80pin type */
       prcr = 0x00;
                                  /* protect register on */
\downarrow
[1]
```

```
[1]
\downarrow
      p0 = 0xff;
      p1 = 0xff;
      pd0 = 0xff;
      pd1 = 0xff;
      if( __CreateTimer_TA0_p1() == TRUE ) /* timer setting */
      {
             if( __EnableTimer_TA0_p1( ) == TRUE ) /* timer start */
             {
                   while(1)
                    {
                          while( (talic & 0x08 ) == 0 ); /* 10ms? */
                          ir_ta1ic = 0;
                          counter--;
                          if (counter == 0)
                                                          /* 1s? */
                          {
                                 p1 = 0xfe;
                                if( onoff )
                                       p0 = 0xf9; /* LED1 on */
                                 else
                                       p0 = 0xff;
                                                    /* LED1 off */
                                 onoff ^= 1;
                                 counter = _1SCNT; /* counter reset */
                          }
                   }
             }
      }
      return;
}
/* end */
```

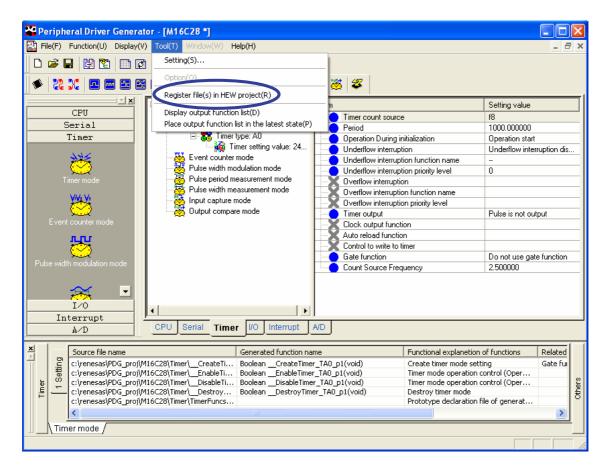
4.4 Build Work

In order to perform a compile/link after completing coding a program, the following steps are required:

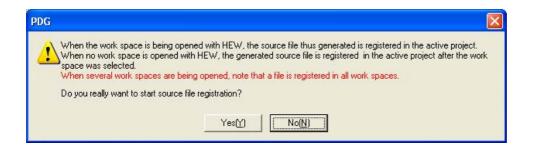
- Registering the files generated by the Peripheral Driver Generator in the High-performance Embedded Workshop.
- Setting the options necessary for a compile (specifying the header file include destination)
- Setting the options necessary for a link (specifying Renesas Embedded Library)

4.4.1 **Registering the Generated Filies**

You will register the files generated by the Peripheral Driver Generator in a workspace of the High-performance Embedded Workshop.



Select [Tool] -> [Register file(s) in HEW project].



The dialog box will appear asking whether to register the files or not. Select [Yes]. The files will be registered in the project currently opend by the HEW.

When registering them in another project, close the currently opend project.

When you attempt to register them while no workspaces are opend, the dialog box appears prompting you to select a workspace for the registration. Select one in the dialog box.

Open		? 🛛
Look jn: 问	m16c_timer	- 🖬 🏪 🛨
Cim16c_time	r	
File <u>n</u> ame: Files of <u>t</u> ype:	HEW Workspaces(*.hws)	 Cancel

When HEW V.4.02 or later is used, the [Library link priority setup] dialog box appears. Move the libraries up and down in the dialog box, according to their priorities. When [OK] is clicked, the files begin to be registered in the HEW project that you selected. *

* When several HEW workspaces are opened, files are registered in all active projects, as stated in the dialog box that asks whether to register the files. Close workspaces that you do not register the files in before performing registration.

Peripheral Driver Generator

Library li	ink priority setup	
Set the pri	ority in which order libraries are linked.	
Priority high	nc30lib.lib C:\renesas\PDG\lib\M16C_Tiny\rapi_m16c_28.lib	
		Up
		1
		F
Priority		Dowr
low		
	OK	Cancel

The message dialog box appears telling you that the registration is completed.

PDG	
⚠	The source file has completely been registered.
	OK

4.4.2 Setting Compile Options

In order to call the functions generated by the PDG, their header file must be included.

#include "sfr28.h"	
#include "m16c28.h"	//When the functions generated by the Peripheral Driver Generator are used, //this header file must be included.
#define _1SCNT (1000/10)	// uns neader the must be mended.

Compiling source files that includes this header file requires to specify its directory.

@m16c_timer - High-performance Embodded Workshop - [m16c_ti	mer.c*]	- 7 🛛
🥪 File Edit View Project Pad Debug Setup Tools Test Window Help		_ 8 ×
🗋 🗅 😅 🖬 🎒 😝 🐰 Renesas M16C Standard Toolchain	🟘 🙀 🙀 😻 🖽 📥 👗 Debug 💽 DefaultSession	- 2 0
Build File Ctrl+F7		1
E-🚭 m16c_timer 🛗 Build F7		
🖻 🖓 m16c_timer 🛗 Build All	***************************************	
Cheader file Build Multiple	c timer.c */	
TimerFung Clean Current Project	, Nov 16, 2007 */	
E C source file Clean All Projects	n program file. */	
Create1Ddate All Dependencies	*/	
Disable 🔏 Stop Tool Execution Ctrl+Break	erated by Renesas Project Generator (Ver.4.8). */	
当Enable 上 「 」 「 Yector.c 」 Include/Exclude Build	*/	
L bean c	-	
Build Phases	_	
실 initisct Build Phases 실 intprg.c 실 m16c_time Build Configurations		
·····································		
Dependencie Callingage Order Dependencie Callingage Order Generate Makefile		
📄 initset.h	/10)	
rapi_timer_m 19 #define PLL_WAIT_1MS		
Tesetprg.h Z0 #define PLL_WAIT_CNT sfr28.h Z1	1 20 /* 20msec */	
TimerFuncs 22 void main (void)		
typedefine.h 23 {		-
¥ 01 01 Al At 21 21 0		
		>
Build \bigwedge Debug \bigwedge Find in Files \bigwedge Macro \bigwedge Test \bigwedge Version Control	· · · · · · · · · · · · · · · · · · ·	<u> </u>
	III III Default1 desktop Read-write 76/78 2	INS

Select [Renesas M16C Standard Toolchain...] from the HEW build menu.

Renesas M16C Standard Toolch	ain 🔹 🔀	
Configuration : Debug All Loaded Projects Control m16c_timer Control cource file Assembly source file	C Assembly Link Librarian Lmc RTOS C Category: Source Show Entries For: Include file directories [4] Specifies the girectory containing the file(s) specified in #include. You can specify up to 8 directories: Add Remove Bemove D Covr	▶ [1
	Options C : -c -finfo -dir "\$(CONFIGDIR)"	

Select the [C] tab in the option dialog box and select the followings:

Category: Source

Show Entries For: Include file directories

Click the [Add] button marked with [1].

	Set include file directory	? 🗙
	<u>R</u> elative to :	ОК
	Custom directory	Cancel
	Directory :	
<	C:\renesas\PDG_proj\M16C28	Bowse

Specify the directory that contains the header file generated by the PDG. This is a project directory created by the PDG.

Click the [OK] button to complete the settings.

4.4.3 Setting Link Options

When HEW V.4.02 or later is used, it is not required to specify libraries.

(1) Select [Renesas M16C Standard Toolchain...] from the build menu.

Renesas M16C Standard Toolch	ain 🤶 🔀	
Configuration :	C Assembly Link Librarian Lmc RTOS C	
Debug	Category : Input Show Entries For : Library files [-L -LD] Specifies directory of Jibrary and Jibrary file to be referenced : Path File Nnc30lib Bernove Up Down	→ [1]
	[-E] Specifies start address of absolute module : Options Link : -L ''nc30lib'' -G -MS -0 ''\$(CONFIGDIR)\\$(PROJECTNAME).x30'' -ORDER data_SE=0400,bss_SE,data_S0,bss_S0,data_NE,bss_N OK	

Select the [Link] tab in the option dialog box and select the followings: Category: Input Show Entries For: Library files

Click the [Add] button marked with [1].

(2) Specify the directory that contains the API library.

🖗 m16c_timer - High-performance Embedde	ied Workshop - [m16c_timer.c] 📃 🗗 🗙	
🥵 File Edit View Project Build Debug Setup	Tools Test Window Help 💷 🗗 🗙	
🛛 🗅 😅 🖬 🎒 🏼 🎒 👗 🐚 💼 😽 🗍 🙀	🖡 💽 📲 🖓 🖉 🕮 🚟 🕇 Debug 💽 DefaultSession 💽 🥕 🙊	
Imile_timer Imile_timer Imile_timer Imile Imil	Source /* FILE :m16c_timer.c */ nesas M16C Standard Toolchain nfiguration: C Assembly Link Libraian Lmc RTOS C */ T Category: Input */ */ */ */ */ */ */ */ */ */ */ */ */	
Yrector.c Yrector.c	I Coaded Projects Show Entries For: Image: Show Entries For: Library files Image: Bit Image: Show Entries For: Library files Image: Bit Image: Bi	→ [1]
. .	L*mc30bc**G M5-0 *SCONFIGENTAS/PROJECTNAME1x30**ORDER data_SE-0400.bss_SE,data_S0.bss_S0.data_NE.bss_N	
× 01 01 AL AT 21 21 0 m .	OK Cancel	
<		
Build (Debug) Find in Files Macro		
Ready	💽 🔝 🖾 Default1 desktop Read-write 16/16 1 INS	

Select [Library directory] from [1].

Set library file directory	? 🔀
<u>R</u> elative to :	ОК
Custom directory	Cancel
Directory :	
C:\renesas\PDG\lib\M16C_Tiny	<u>B</u> rowse

Relative to: Select [Custom directory].

Directory: Specify the directory that contains the API library: C:\Renesas\PDG\lib\m16c_tiny

After entering the directory name, click [OK].

(3) Specify the API library name.

Click [Add] again and select [Library file] in [1].

Set library file	? 🛛
Library name :	ОК
rapi_m16c_28	Cancel

Enter the library name, "rapi_m16c_28." (Do not add the.lib extension.)

4.4.4 Excluding Interrupt Vector Files

Creating a workspace with the HEW registers intprg.c, which is one of several startup files. (When the H8 300H/Tiny is used, the workspace creation wizard asks whether to create intprg.c or not.)

Since the PDG creates interrupt vector functions, intprg.c is redundant for vectors. Therefore, intprg.c needs to be excluded from the build target.

Note that when HEW V.4.02 or later is used, the interrupt vector functions are automatically excluded from the PDG.

🖗 m16c_timer - High-performance Embedded Workshop - [m16c_timer.c]	- 7 🗙
i File Edit View Project Build Debug Setup Tools Test Window Help	_ 8 ×
D 😂 🖬 🖉 🎸 🍋 😭 😚 🦷 🐂 🔍 💌 ベ 谷 🏠 🔮 🕮 🚟 🎽 Debug 💽 DefaultSession 💌 考	@
Image: Section of the secti	
Image: Second	
Construction 29 cml ≤= 0x3f; Construction 30 cmc0 ≤= 0;	
Extraction 34 Wheelor Wheelor Wheelor Wheelor Wheelor Wheelor <td< td=""><td></td></td<>	
L = 1; C	
■ initicat = 0uin0 options * r = 0x04; /* protect register off */ = ■ rapiti Add Files INS r = 0x03; /* 80pin type */ = ■ resetp resetp resetp r = 0x00; /* protect register on */ = ■ rf2st; = 0x02; - /* protect register on */	
Exclude Build intrg.c Verson-Control	▼ ▶ //
S. B. Configure View	
X OL OT AL AT 21 Show Differences	
Alow Docking Hide	~
Properties	
Sumid & Debug & Find in Files & Macro & Test & Version Control /	>
Ready Read-write 74/74 1 INS	

Select intprg.c and right-click on it.

The pop-up menu opens. Select [Exclude Build intprg.c].

All the settings are completed.

Perfom a build and execute the application by using the debugger.

Peripheral Driver Generator V.1.02 Guide Book

Publication Date:	Nov. 28, 2007	Rev.1.00
Published by:	Sales Strategic Planning Div. Renesas Technology Corp.	
Edited by:	Microcomputer Tool Development Department Renesas Solutions Corp.	

© 2007. Renesas Technology Corp. and Renesas Solutions Corp., All rights reserved. Printed in Japan.

Peripheral Driver Generator V.1.02 Guide Book



Renesas Electronics Corporation 1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan