To our customers,

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.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

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

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

User's Manual

Peripheral Driver Generator V.1.04

Guide Book

Renesas Electronics
www.renesas.com

Rev.1.00 2009.05

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

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.

For inquiries about the contents of this document or product,
e-mail to your local distributor.

Renesas Tools Homepage http://www.renesas.com/en/tools

Contents

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, R8C/13, SH7125, and H8S/20103.

(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.
These options are registered automatically when using certain version of HEW.

(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.

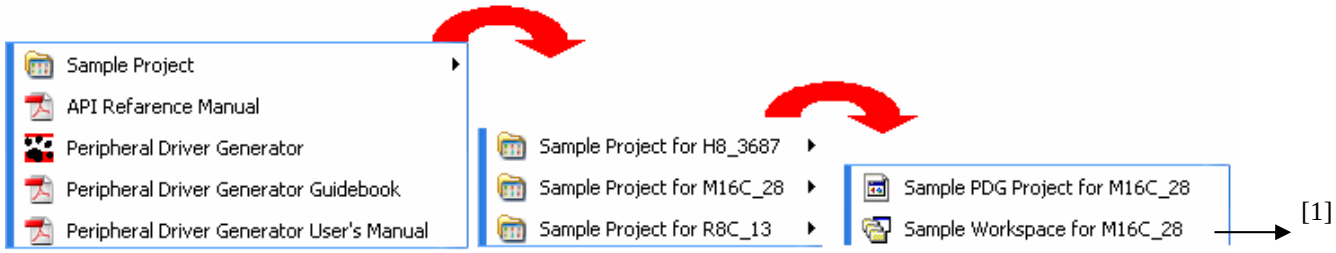
* Backup copies of the attached sample project file originals are stored in sample.bak under the PDG install directory (When recovering the files, copy the directories under sample.bak into the sample directory).

2.1 Creating a Workspace with the HEW [HEW]

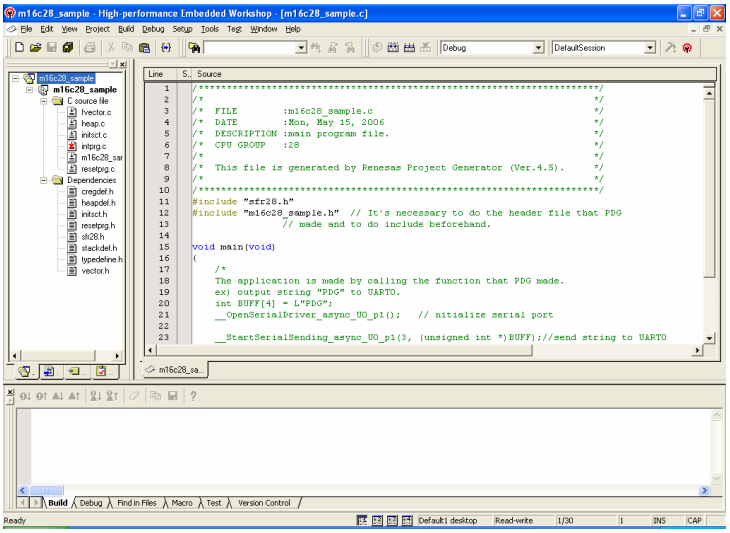
In this section, you use an empty workspace for the HEW, which is included in the PDG package.
Select [Start] -> [Renesas] -> [Peripheral Driver Generator] -> [Sample Projects]. Then, open any one of the followings:

- Sample workspace for H8/3687
- Sample workspace for M16C/28
- Sample workspace for R8C/13
- Sample workspace for SH7125
- Sample workspace for H8S/20103

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.



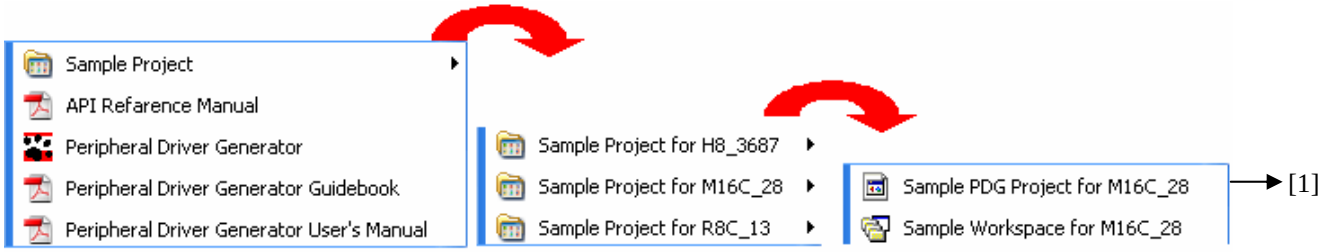
Note: H8/3687, M16C/28,R8C/13 sample workspaces are created with HEW V.4.00.03. SH7125 and H8S/20103 sample workspace are created with HEW V.4.04.01. These workspaces cannot be opened with earlier version of HEW. When opening with later version of HEW, the message box opens. Click [OK] to open the workspace.

2.2 Creating a Project with the PDG [PDG]

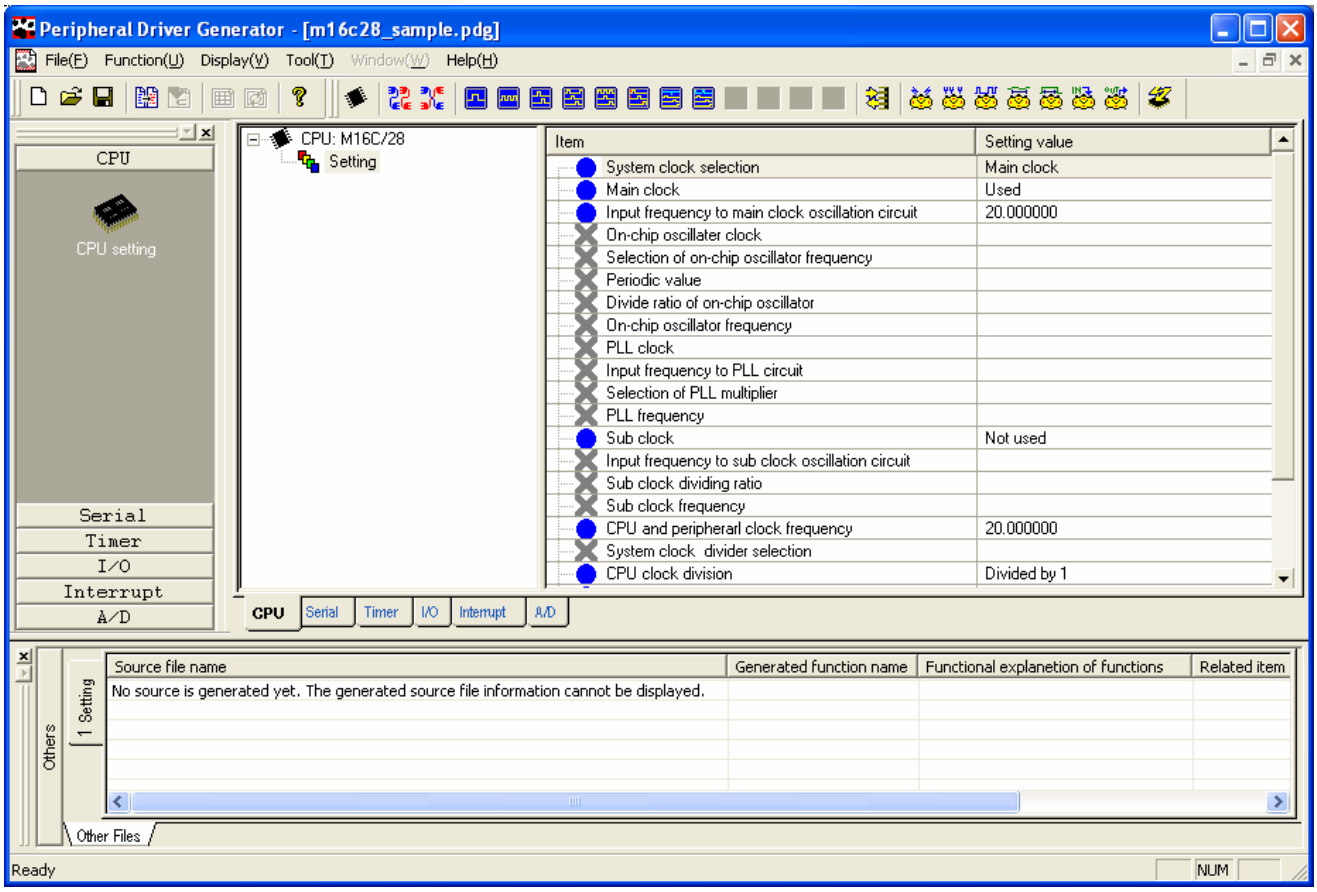
In this section, you use an empty project for the PDG, which is included in the PDG package.
Select [Start] -> [Renesas] -> [Peripheral Driver Generator] -> [Sample Projects]. Then, open any one of the followings:

- Sample PDG project for H8/3687
- Sample PDG project for M16C/28
- Sample PDG project for R8C/13
- Sample PDG project for SH7125
- Sample PDG project for H8S/20103

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



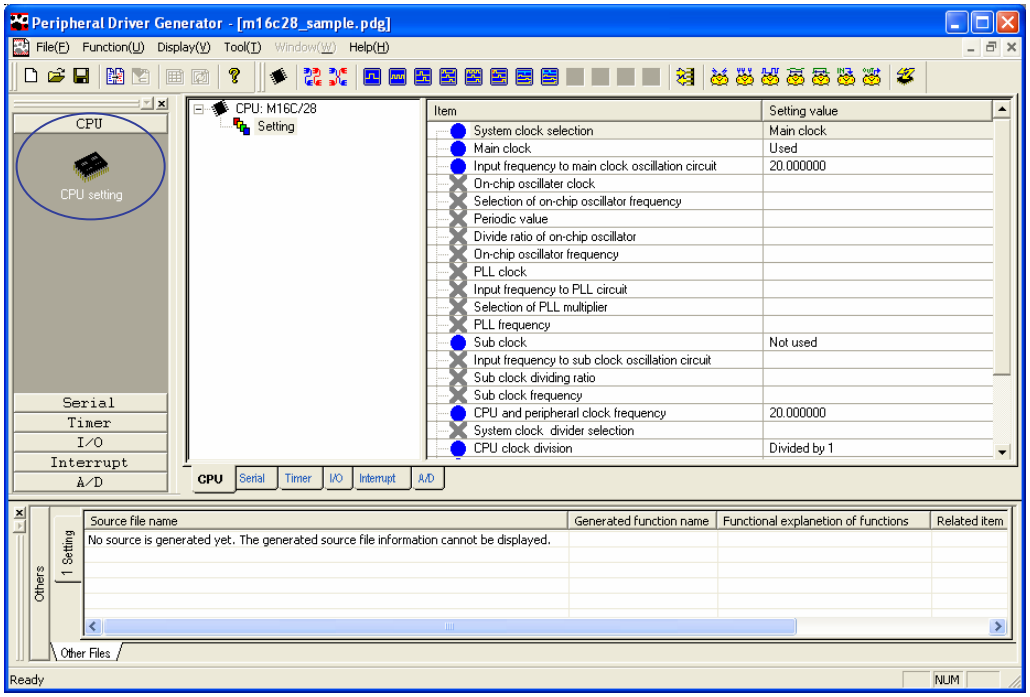
The PDG will be launched after [1] is selected.

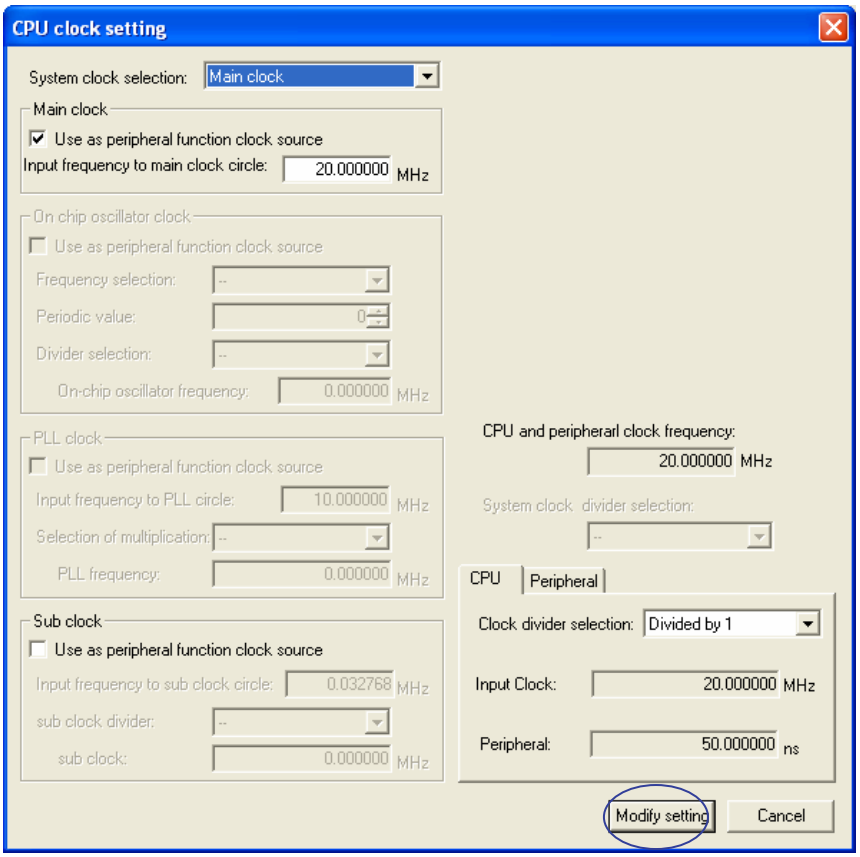


2.3 Setting the Peripheral I/O Modules [PDG]

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

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





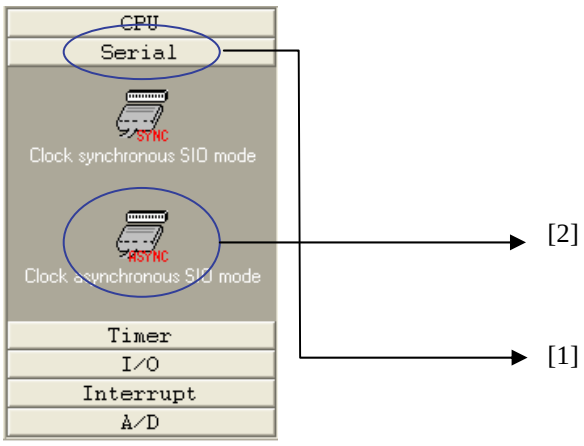
In the [CPU clock setting] dialog box, perform settings according to the program to be created and click [Modify setting] to complete the settings.

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].



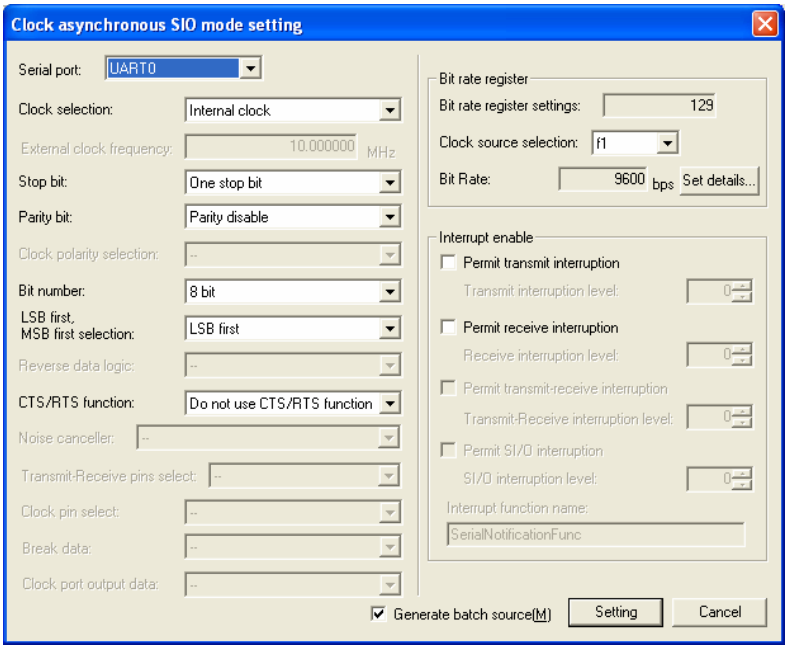
In the [Clock asynchronous SIO mode setting] dialog box, specify the items (bit number, parity bit, baud rate, etc,) and select a serial port (UART0, UART1, etc,).

Check [Generate batch source (M)] and click [Setting].

C source files reflecting these settings will be generated.

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

* When using SH/Tiny series and H8S/Tiny series, C source files cannot be generated by above procedure. To generate C source files, it is necessary to setup the serial ports by the following operations.



When using SH/Tiny series and H8S/Tiny series, it is necessary to select pin functions to setup the peripheral I/O modules from the I/O port setting dialog box.

For example, if you setup SCI/channel 0 using SH/Tiny, the corresponding pins need to be set to SCI function by the following operations.

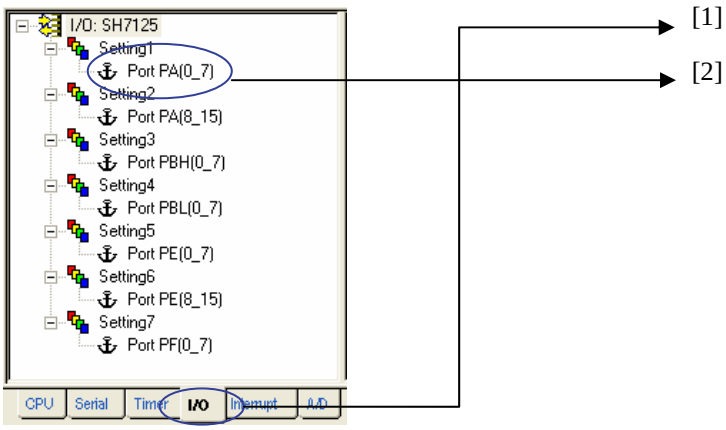
[1] Select the [I/O] tab in the peripheral setup window.

[2] The pins corresponding to SCI/channel 0 are as follows.

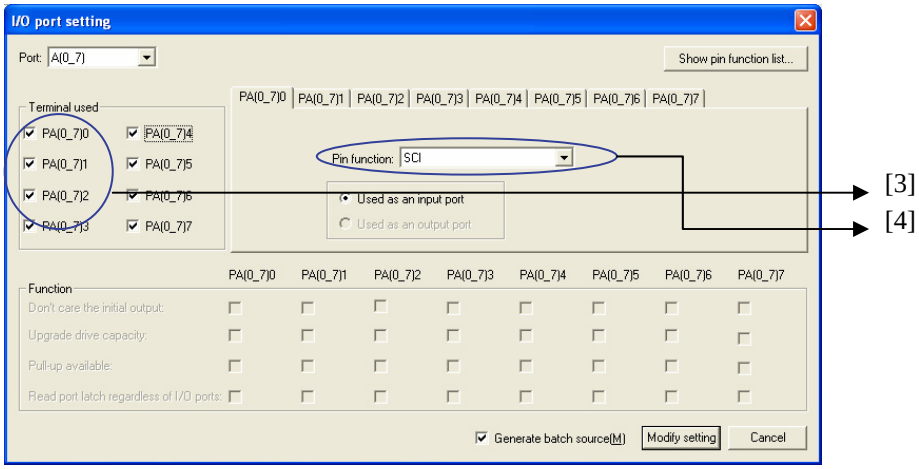
RXD0 : PA0, PA10, or PE1
TXD0 : PA1, PA11, or PE2
SCK0: PA2, PA12, or PE3.

When setupping the SCI, SCK and RXD, or SCK and TXD need to be selected.
In the example below, PA0, PA1, and PA2 are selected as RXD0, TXD0 and SCK0.

Double-click PortPA(0_7) on the peripheral setup window to open the [I/O Port setting] dialog box.



[3] Check [PA(0_7)0], [PA(0_7)1], and [PA(0_7)2] in [Terminal used]. These checkboxes corresponds to PA0, PA1, and PA2. By checking the checkboxe, setting tab for each pin appears.



[4] Select [SCI] as pin function in each tab.

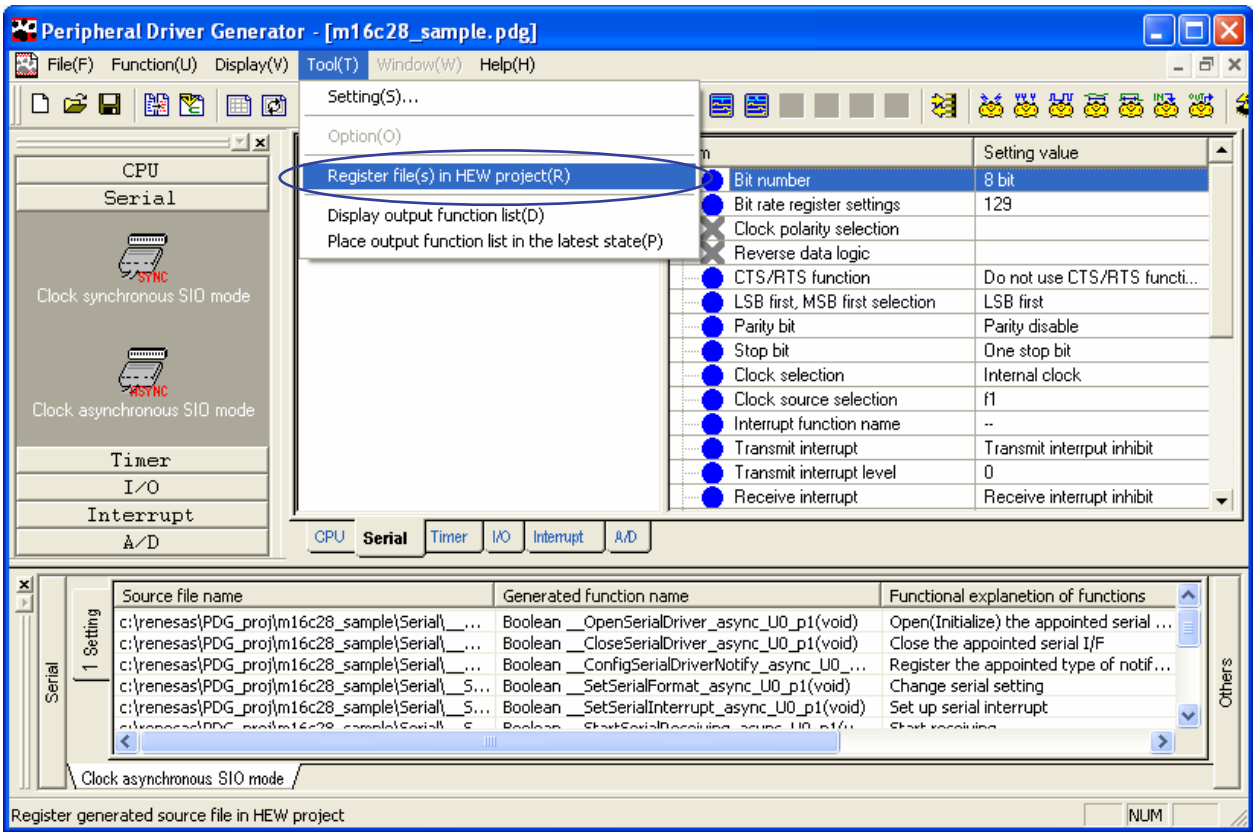
[5] Click [Modify setting] to generate C source files.

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.

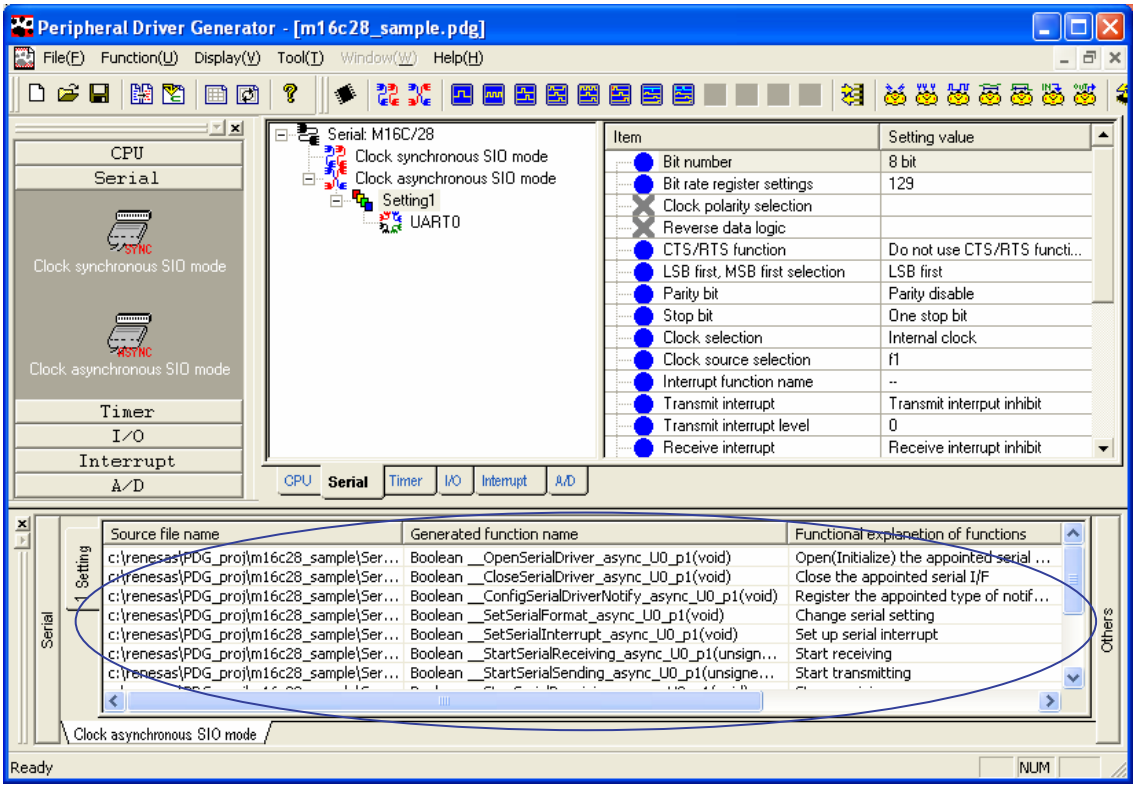


Note:

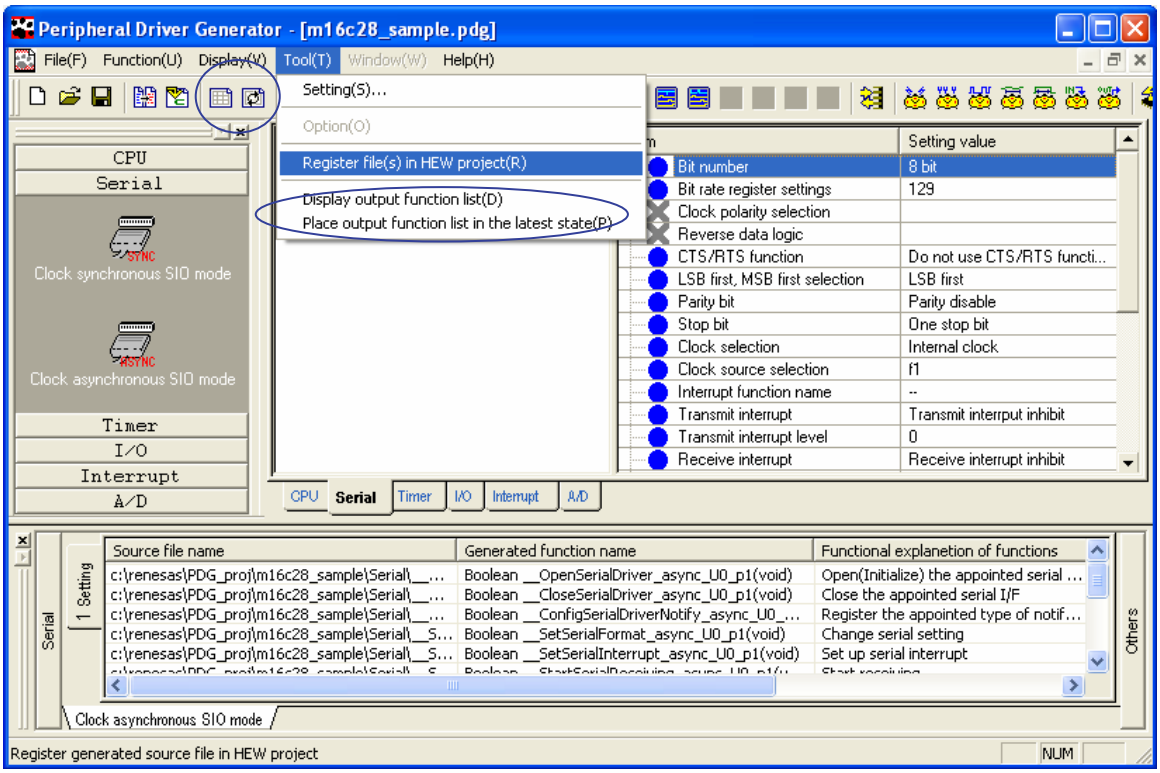
- Once the source files generated by the PDG are registered in the HEW, you cannot cancel their registration via the PDG. Cancel the registration via the HEW, if necessay.
- Make sure that the CPU group selected for the HEW workspace is the same as for the PDG project before registering the source files.

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.

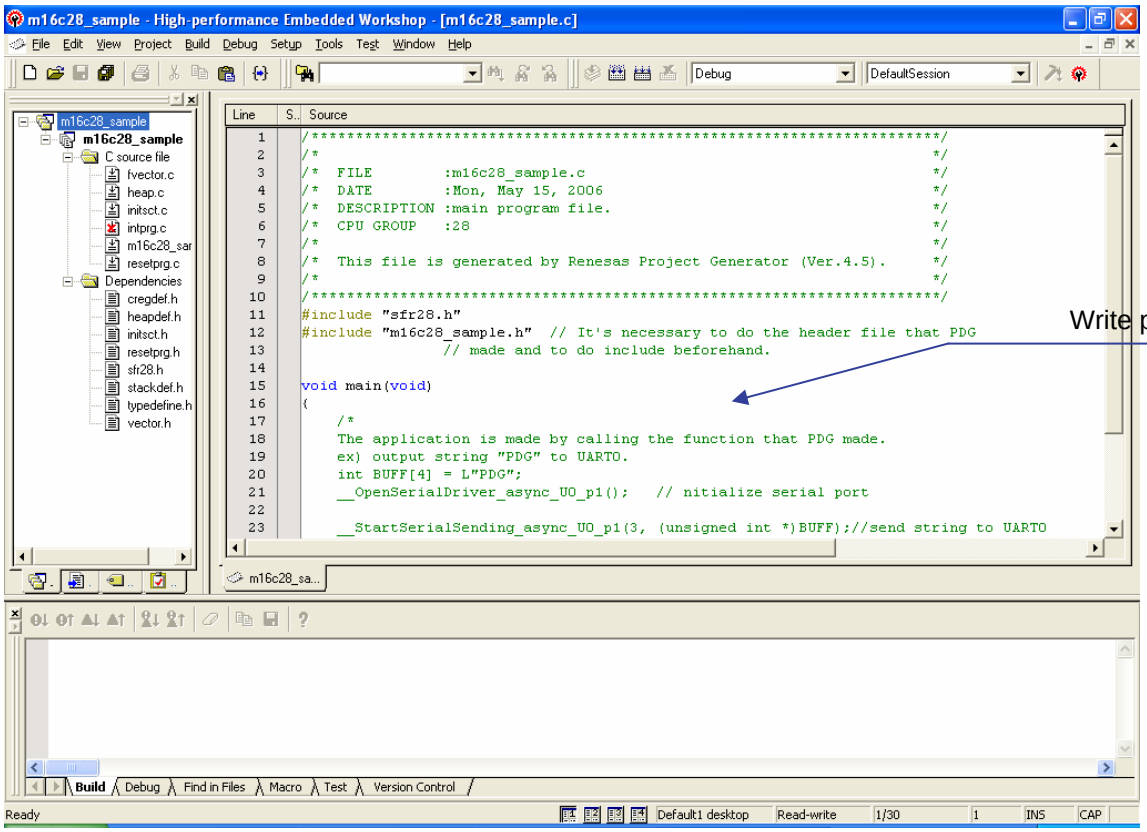


To list output functions in CSV file format, select [Display output function list] from the tool menu. The application associated with the *.csv extension will be launched.



2.6 Creating the Application [HEW]

The application is created by using the HEW.



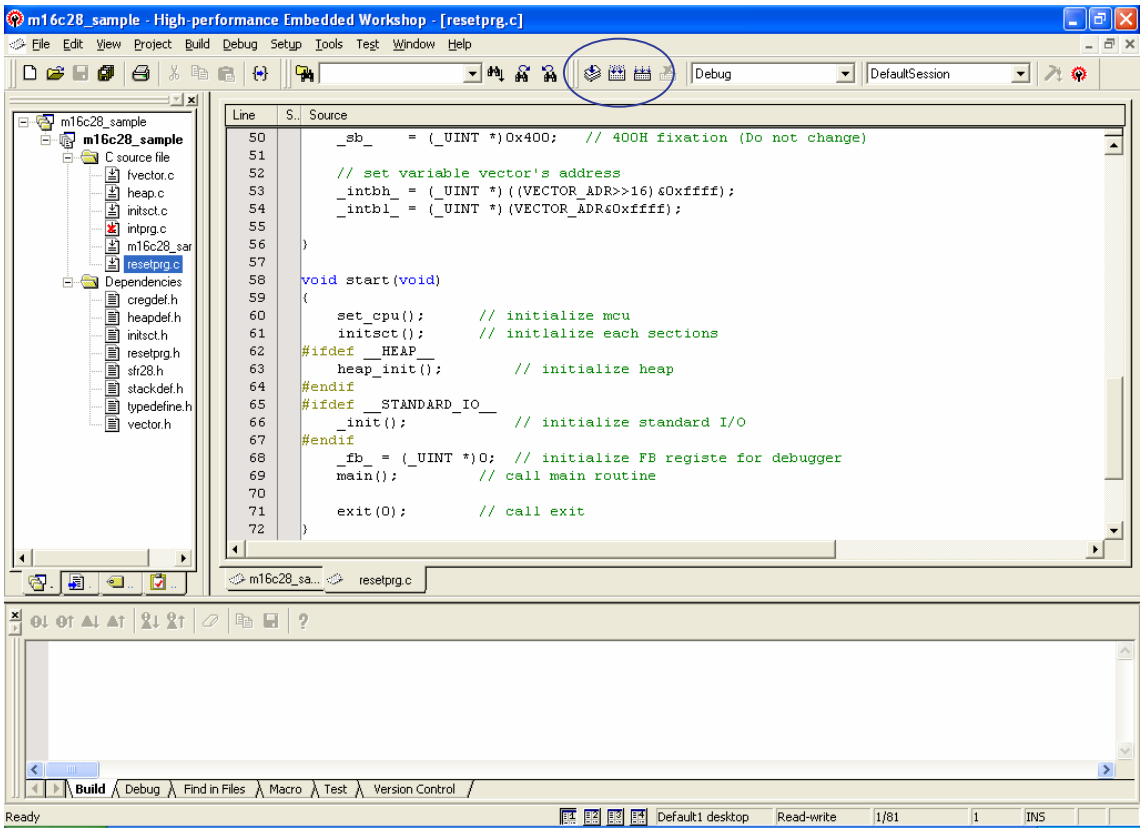
For example, to send data to the UART0 serial port, call the following function:

__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, lib\R8C_Tiny, lib\SH_Tiny, or lib\H8S_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.

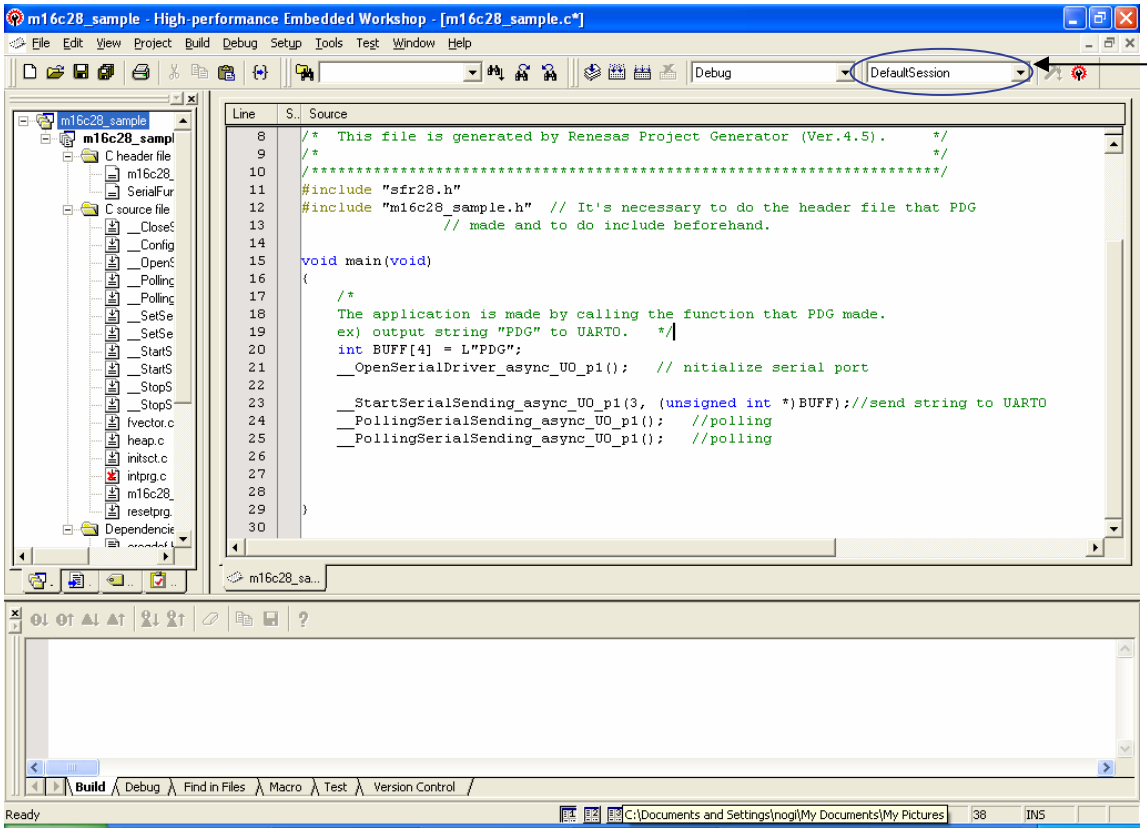
```
int BUFF[4] = L"PDG";
__OpenSerialDriver_async_U0_p1();                          //Initializes UART0

__StartSerialSending_async_U0_p1(3, (unsigned int *)BUFF);  //Sends to UART0
__PollingSerialSending_async_U0_p1();                      //Polling processing
__PollingSerialSending_async_U0_p1();                      //Polling processing
```

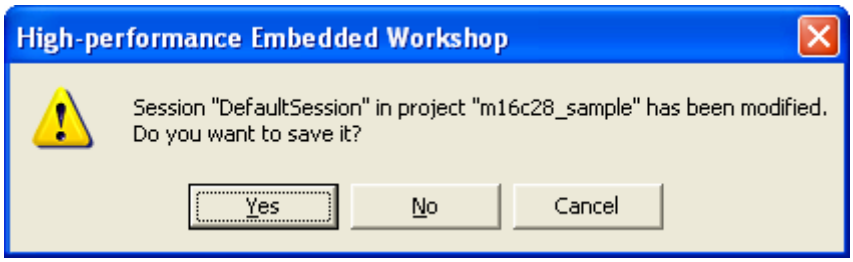
(2) Use the PDG to register the generated source files in the HEW.

(3) Perform a build.

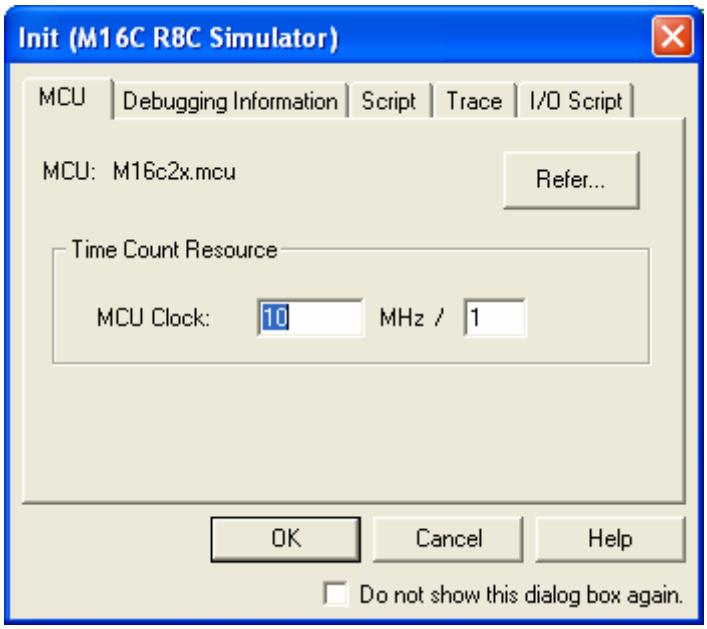
- Executing the simulator



[1] Select [SessionM16C_R8C_Simulator] for session.
[2] Select [Yes].



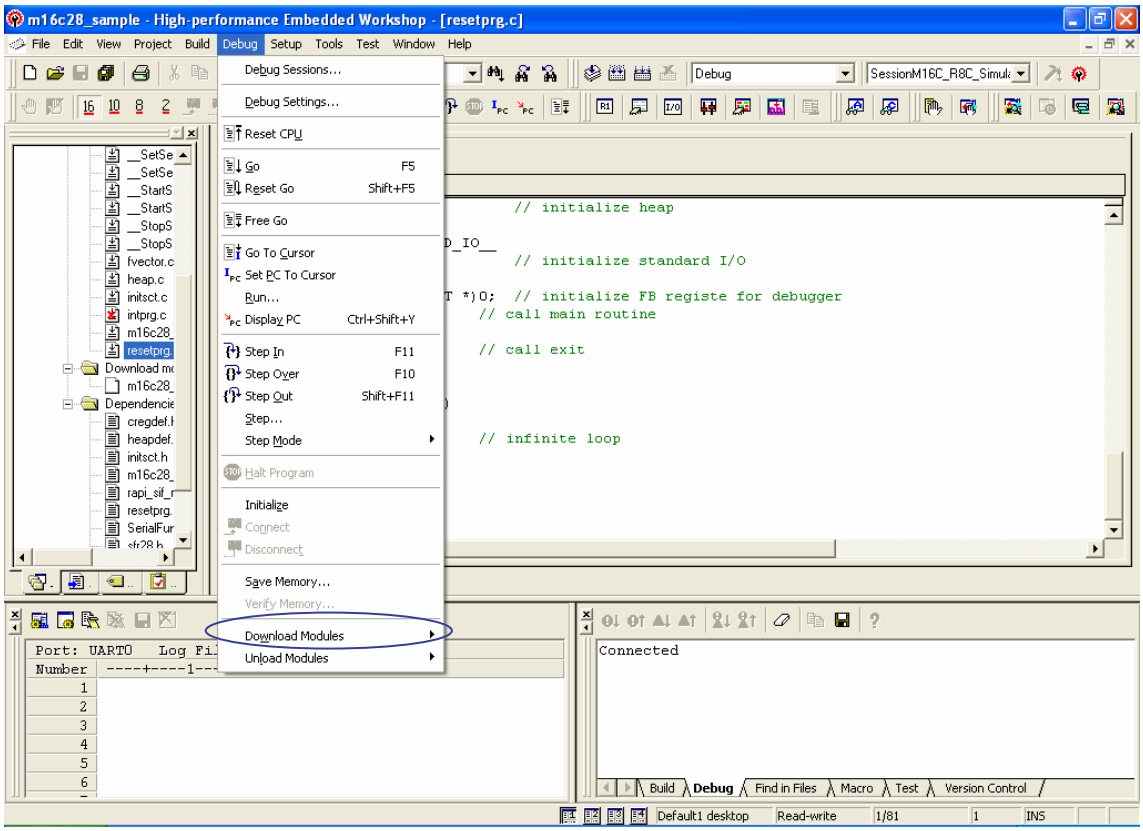
[3] Select [OK].



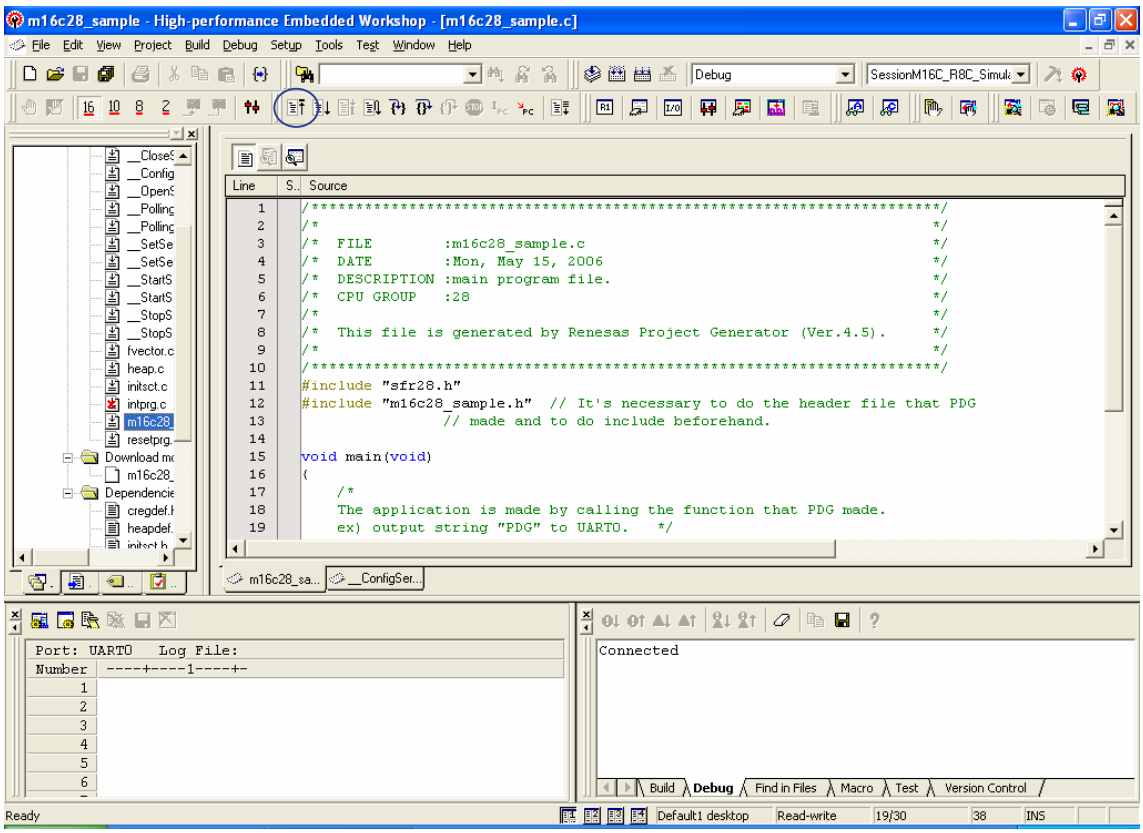
[4] Select [Download Modules] to download the following file.

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.

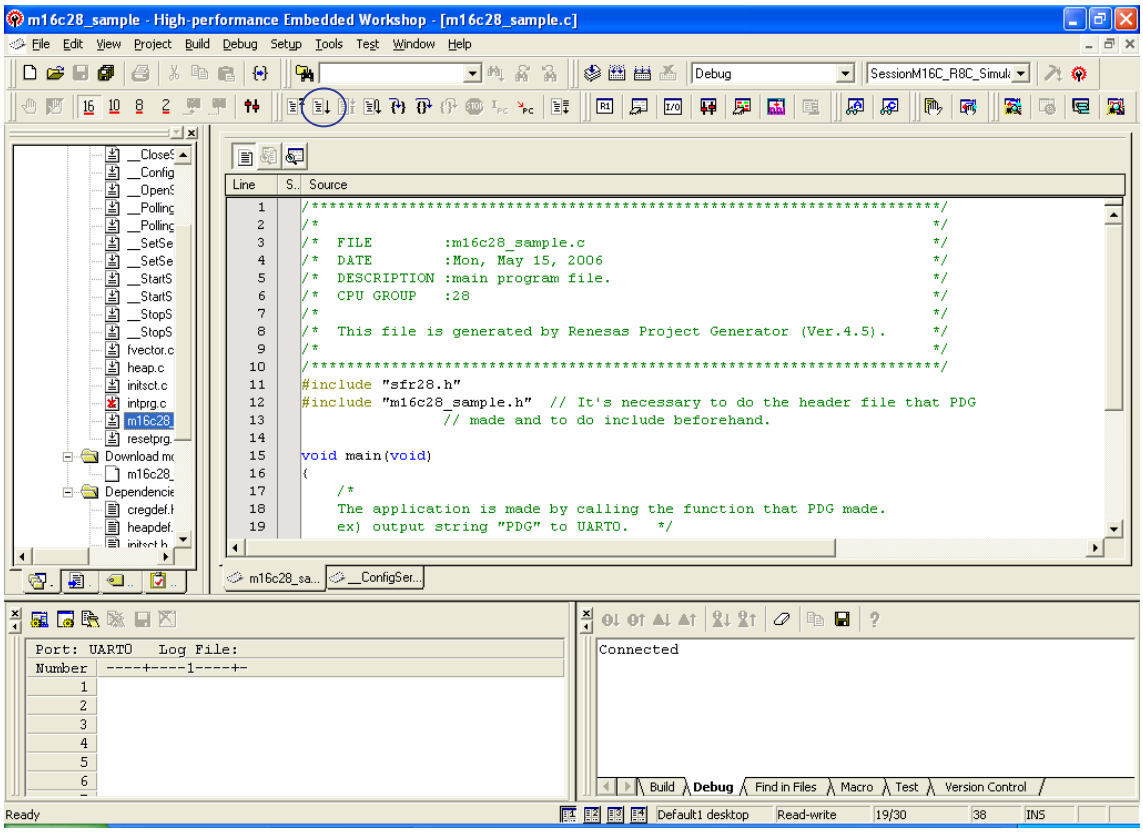


[6] Set a breakpoint at the place where exit(0) is called in resetprg.c.

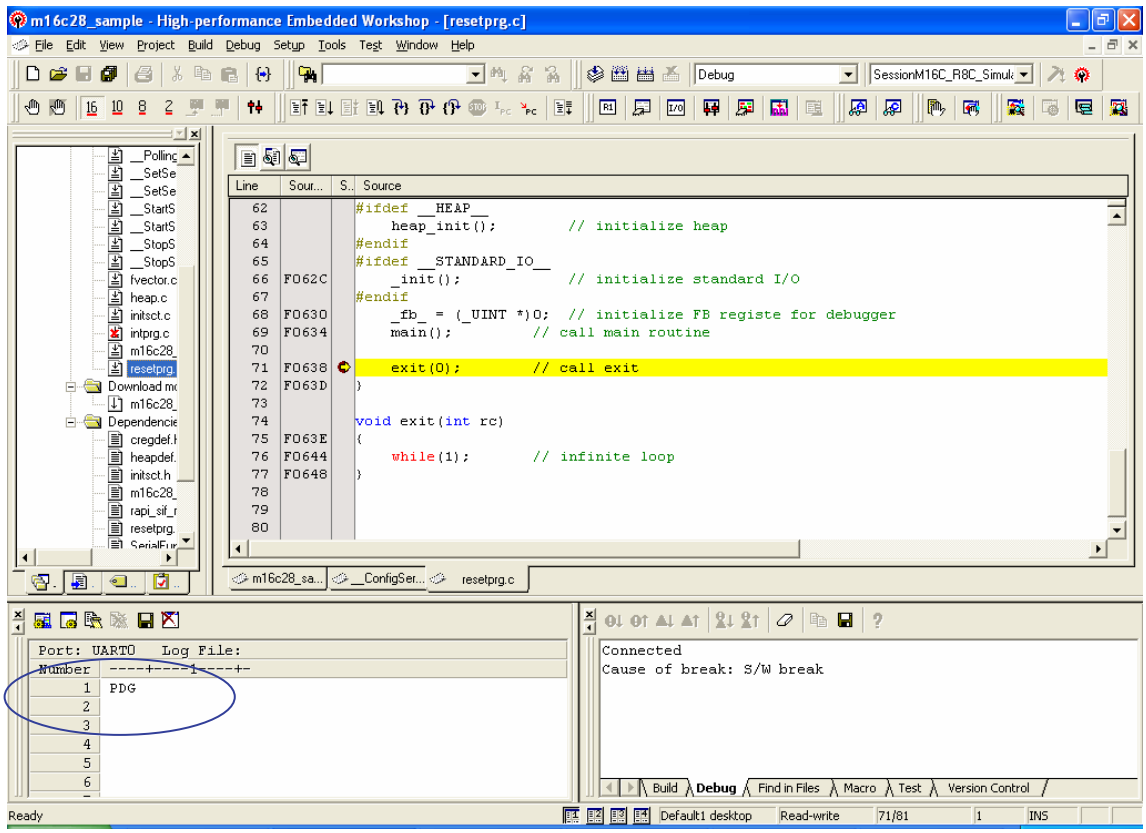
```
_fb_ = (_UINT *)0;          // initialize FB registe for debugger
main();                     // call main routine

exit(0);                    // call exit
```

[7] Click the run button.



[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)** *1
- **Setting for linking API libraries (using –L option)** *2

*1 When the High-performance Embedded Workshop V.4.05 or later is used, PDG specifies -I option automatically when PDG registers the source files. It is necessary to specify –I option manually when using High-performance Embedded Workshop V.4.04 or earlier version or adding source file to the project after source file registration from PDG.

*2 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.

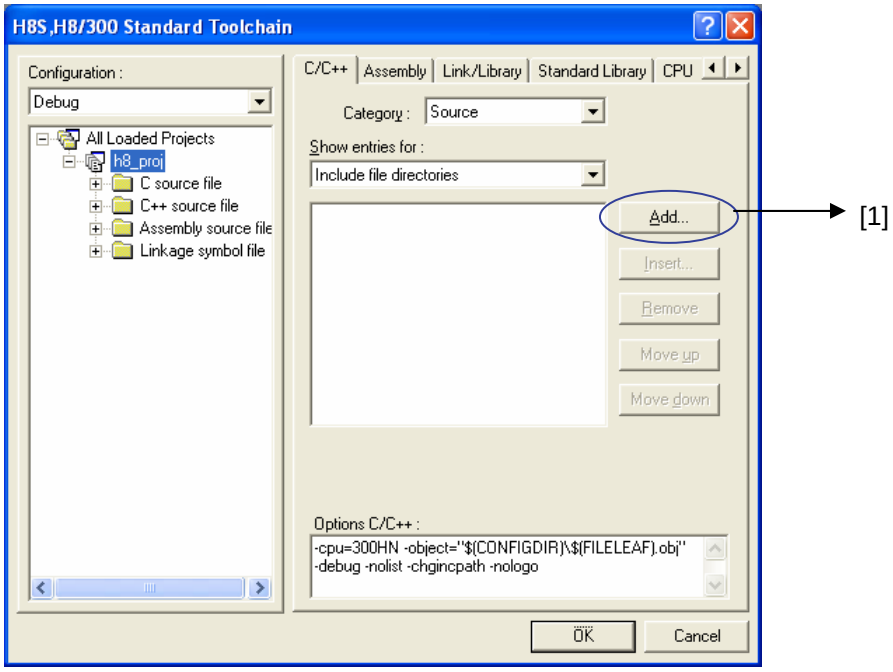
A header file name is <project name>.h.

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.

```
<sample.c>
#include "m16c28_sample.h"

void main(void)
{
        __OpenSerialDriver_sync_U0_p1();
        __ConfigSerialDriverNotify_sync_U0_p1();
        __SetSerialFormat_sync_U0_p1();
                :
                :
                :
}
```

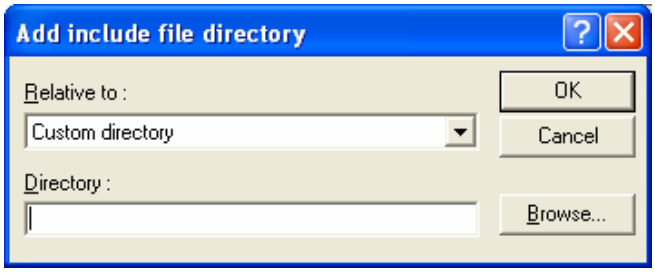
(1) From the build menu, select “Renesas M16C Standard Toolchain…” for M16C/Tiny and R8C/Tiny, select “H8S,H8/300 Standard Toolchain…” for H8/300H Tiny and H8S/Tiny, or select “SuperH RISC engine Standard Toolchain …” for SH/Tiny.



(2) Select the [C/C++] tab ([C] tab for M16C/Tiny and R8C/Tiny).
When H8/300H Tiny, H8S/Tiny, or SH/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.



In the [Add include file directory] dialog box, select the relative path category and enter the directory name.
For H8/300H Tiny, H8S/Tiny, or SH/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

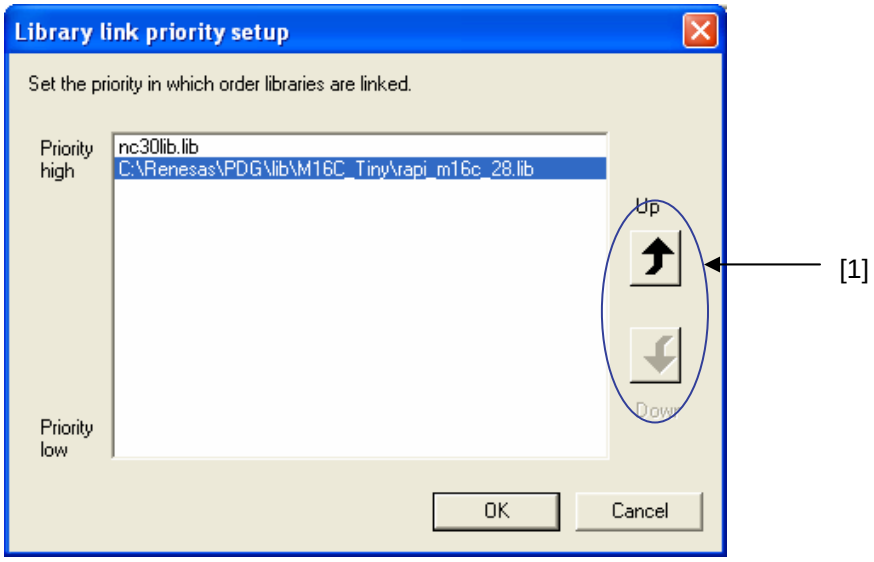
3.2.1 List of Libraries for Sample Projects

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
H8/3687	lib\H8_Tiny	rapi_h8_3687.lib
R8C/13	lib\R8C_Tiny	rapi_r8c_13.lib
M16C/28	lib\M16C_Tiny	rapi_m16c_28.lib
SH7125	lib¥SH_Tiny	rapi_sh7125.lib
H8S/20103	lib¥H8S_Tiny	rapi_h8s_20103.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.

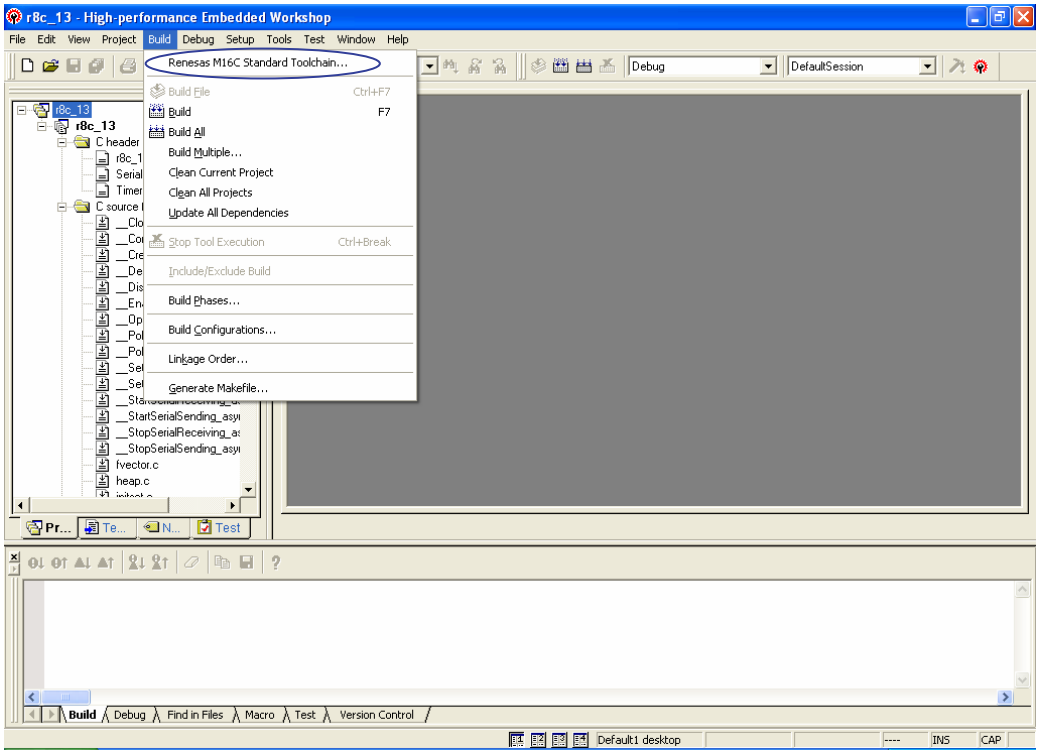


In this dialog box, determine the priorities of the libraries.
When the same symbol name exists in different libraries, the library that has the highest priority among them are selected.
When changing the priorities, select a library and click the buttons marked with [1]. A library with higher priority is listed higher.

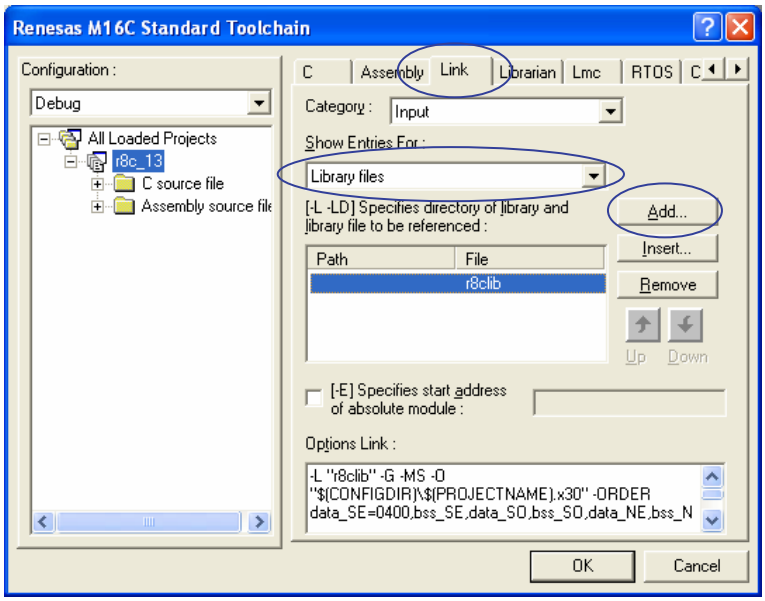
3.2.3 When Using Earlier Version than HEW V.4.02

When using an earlier HEW version than 4.02, follow the steps below to specify libraries.

(1) From the build menu, select [Renesas M16C Standard Toolchain…] for M16C/Tiny and R8C/Tiny, select [H8S,H8/300 Standard Toolchain…] for H8/300H Tiny and H8S/Tiny, or select [SuperH RISC engine Standard Toolchain …] for SH/Tiny.

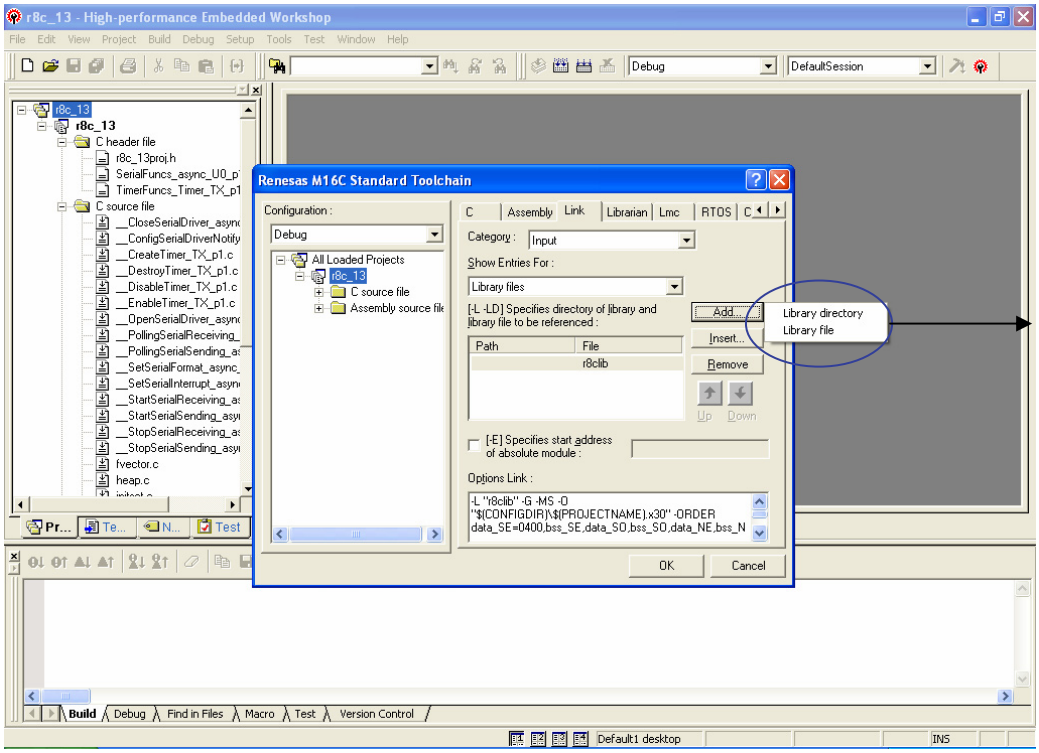


(2) Select the [Link] tab in the [Renesas M16C Standard Toolchain] dialog box, or select the [Optimization linker] tab in the [H8S,H8/300 Standard Toolchain] dialog box or [SuperH RISC engine Standard Toolchain …] dialog box.

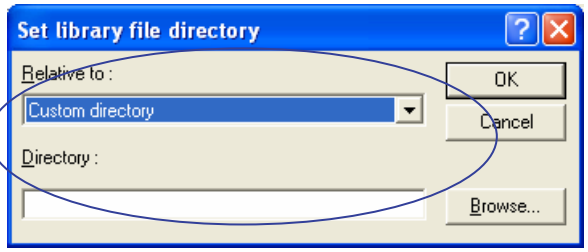


Select [Library files] ([Library files] for H8/300H Tiny and H8S/Tiny) for [Show Entries For] and click [Add].

(3) Specify the directory containing the API libraries.



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



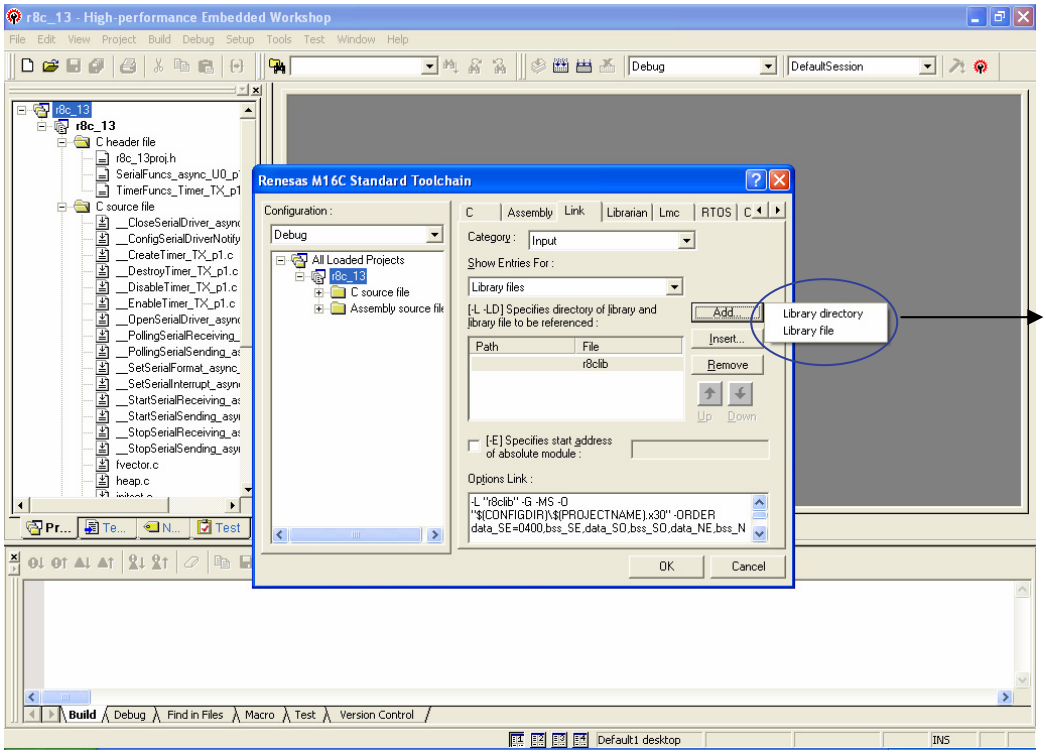
Relative to: Select [Custom directory].

Directory: Specify the directory containing the API libraries.

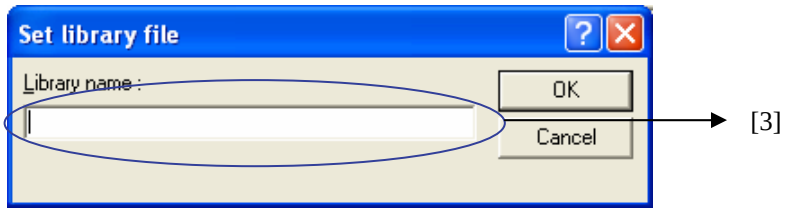
Example) C:\Renesas\PDG\lib\M16C_28

For H8/300H Tiny, H8S/Tiny, or SH/Tiny, specify the directory name and liblary file name.

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



[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.

3.3 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.
The method to use interrupt vectors is as follows.

3.3.1 When Using H8/300H Tiny, H8S/Tiny, M16C/Tiny, and R8C/Tiny

The startup program of HEW includes the source file intprg.c that declares interrupt vector functions. The source file created by PDG also includes interrupt vector functions. To avoid duplication of interrupt functions, intprg.c is excluded from the project of HEW when PDG registers the source files. (Please exclude it manually when using HEW V.4.02 or earlier version.) If you want to add interrupt operation, please add another file that includes interrupt function.
When using M16C, if transmit interrupt of SCI/UART0 is enabled both intprg.c and m16c_28.c created by PDG has interrupt function of vector 17. Therefore intprg.c is excluded.
For example, if you want to add DMA0 interrupt operation, please add file of DMA0 interrupt function.

InterruptTxU0_m16c_28.c (Created by PDG)

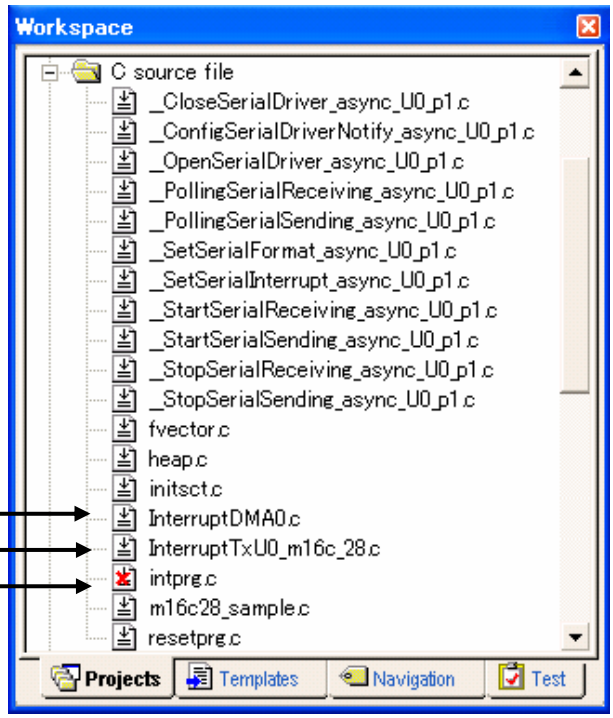
```
#pragma INTERRUPT _InterruptTxU0(vect=17)
void _InterruptTxU0(void)
{
      _SifdComTxU0();
}
```

intprg.c (Startup programs)

```
      ⋮
#pragma interrupt _uart0_trance(vect=17)
void _uart0_trance(void){}
      ⋮
```

例：InterruptDMA0.c (User file)

```
#pragma interrupt _Interrupt_dma0(vect=11)
void _Interrupt_dma0(void)
{
      //DMA0 interrupt operation
}
```



Workspace window of HEW

3.3.2 When Using SH/Tiny

The startup program of HEW includes the source file intprg.c that declares interrupt vector functions. The source files intprg_sh7125_pdg.c and intprg_sh7125.c created by PDG also includes interrupt vector functions. To avoid duplication of interrupt functions, intprg.c is excluded from the project of HEW when PDG registers the source files. (Please exclude it manually when using HEW V.4.02 or earlier version.)
In intprg_sh7125_pdg.c, the interrupt functions PDG can handle are collected. This file is created and edited only by PDG. PDG overwrite this file every code generation.
In intprg_sh7125.c, the interrupt functions PDG doesn't handle are collected. User can edit this file because PDG doesn't overwrite this file.

intprg_sh7125.c (Created by PDG. User can edit.)

```
// 4 Illegal code
void INT_Illegal_code(void)
{
      //User can add interrupt operation
}

// 6 Illegal slot
void INT_Illegal_slot(void){}

// 9 CPU Address error
void INT_CPU_Address(void){}
```

intprg_sh7125_pdg.c
(Created by PDG. Edit only by PDG)

```
// 11 NMI
void INT_NMI(void){}

// 64 Interrupt IRQ0
void INT_IRQ0(void){}

// 65 Interrupt IRQ1
void INT_IRQ1(void){}
```

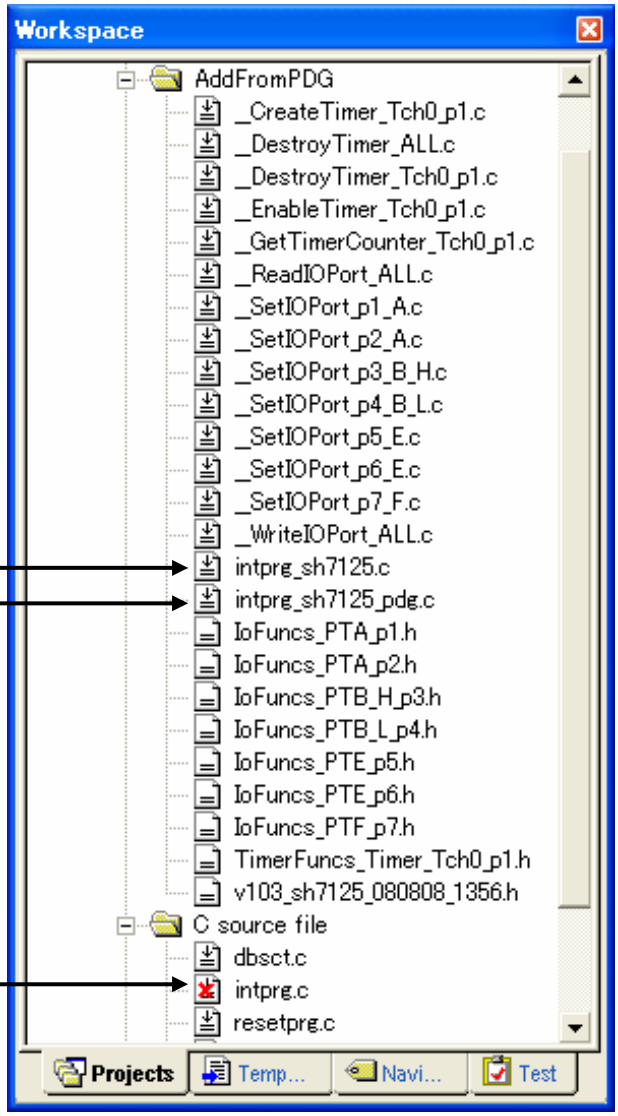
intprg.c (Startup program)

```
// 4 Illegal code
void INT_Illegal_code(void){}

// 6 Illegal slot
void INT_Illegal_slot(void){}

// 9 CPU Address error
void INT_CPU_Address(void){}

// 11 NMI
void INT_NMI(void){}
```



Workspace window of HEW

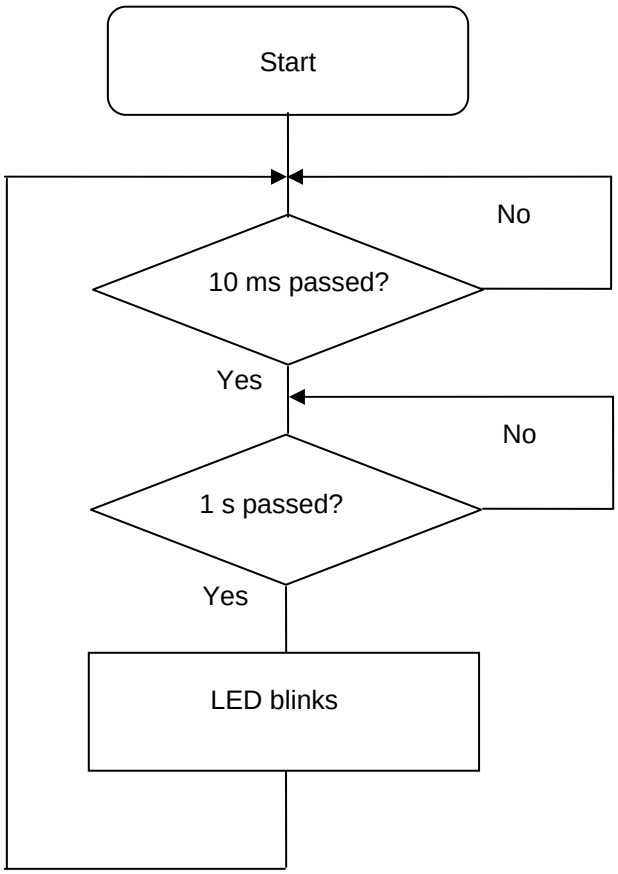
Note : When using SH7125 with HEW V.4.04 or earlier version, it is necessary to register the include file path of vect.h that is startup program of HEW manually for intprg_sh7125.c and intprg_sh7125_pdg.c. When using HEW V.4.05 or later, the include path is registered automatically.

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

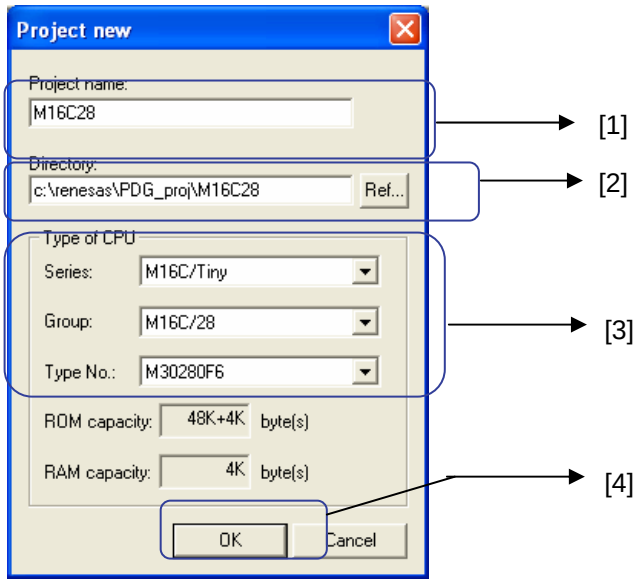
Create an application based on the flowchart below.
Microcomputer used for the application is M16C/28.



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.

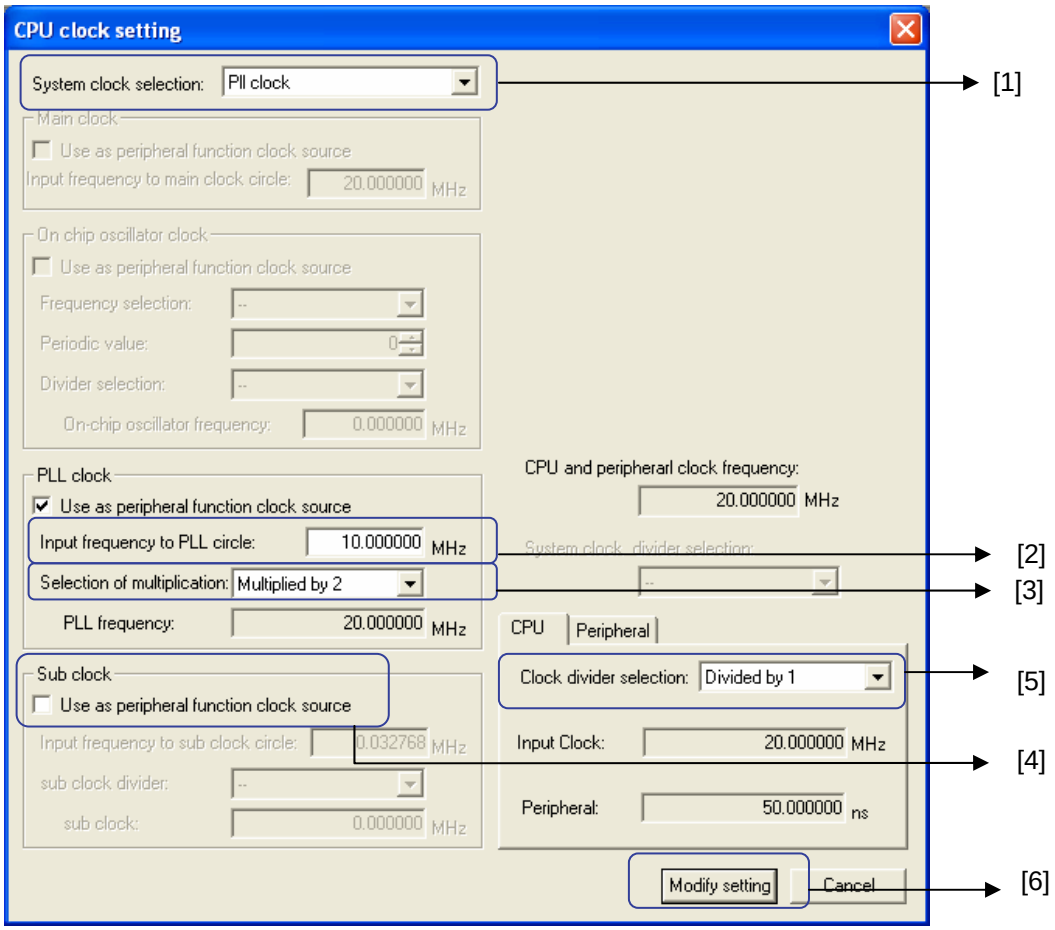
[2] Specify a directory to store the project file in.
By default, a directory with the same name as the project is created under c:\renesas\PDG_proj.

[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.

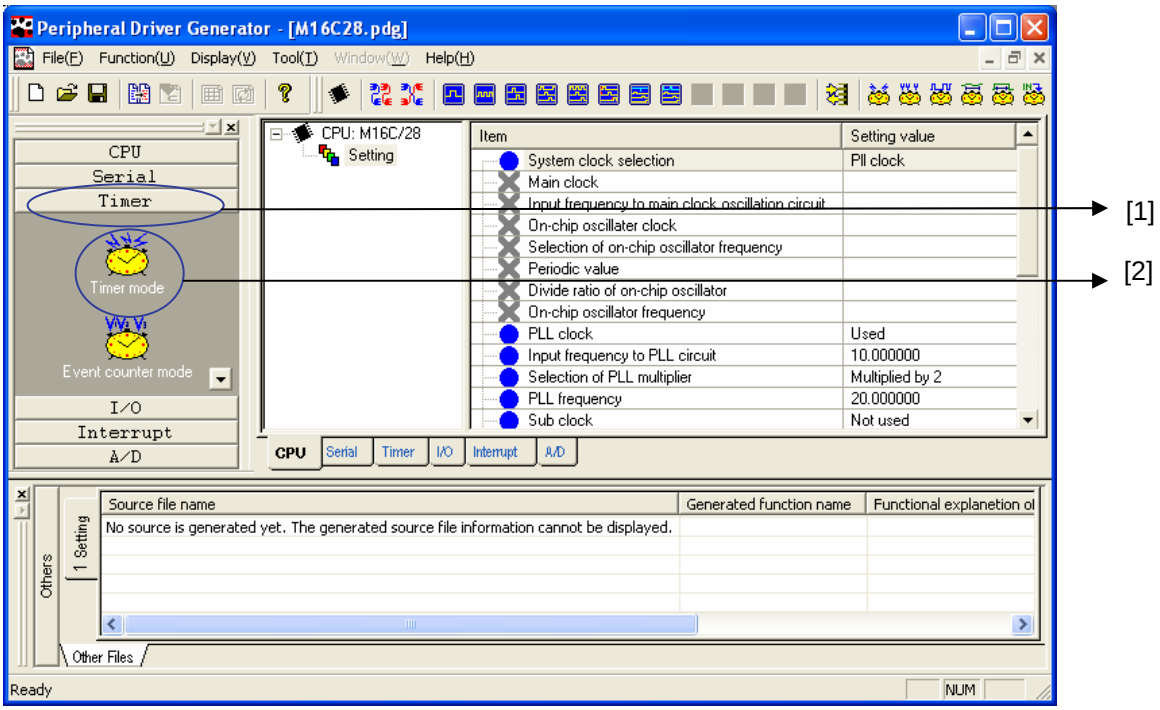
4.2.2 Setting Clocks

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



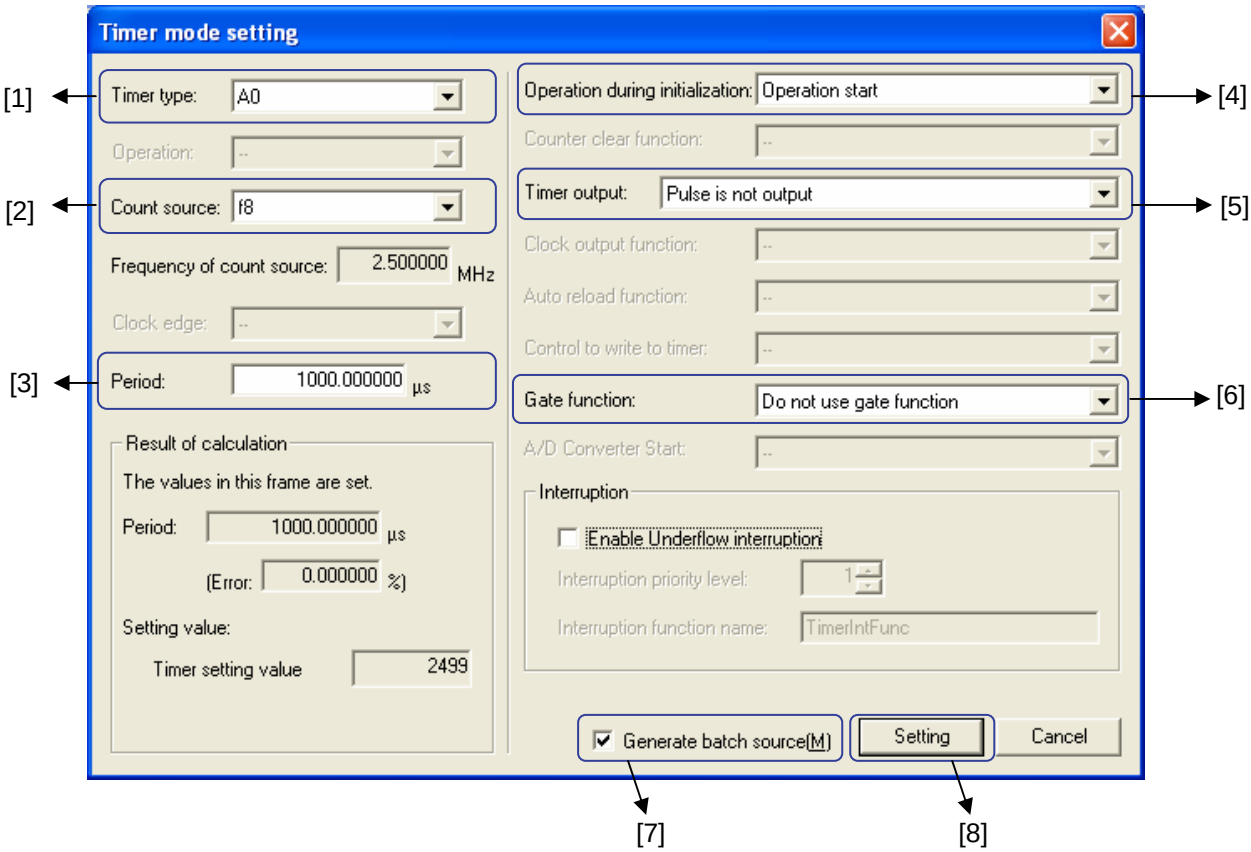
[1] Select [PLL clock] in the [System clock selection].
[2] Specify the frequency as 10 MHz.
[3] Select [Multiplied by 2] for [Selection of multiplication].
[4] Do not use sub-clock as peripherarl clock.
[5] Select [Devided by 1] for CPU clock divider.
[6] Click [Modify setting] to complete the settings.

4.2.3 Setting Timer Mode

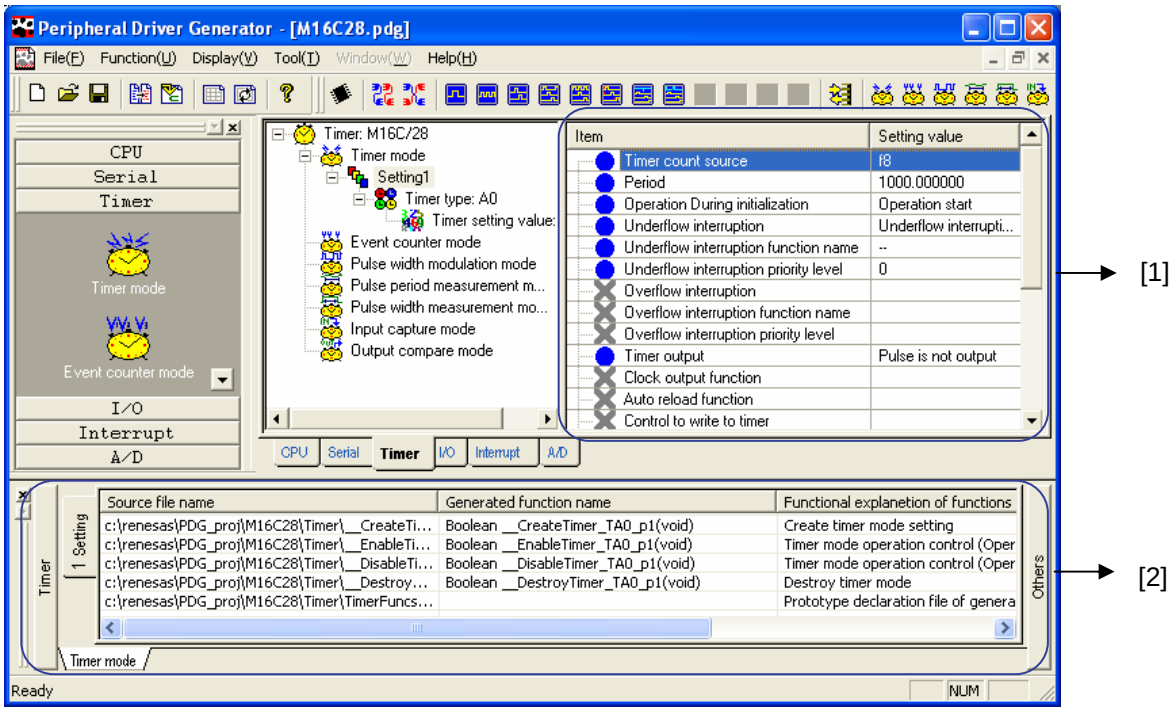


[1] Select the [Timer] tab.

[2] Select and click [Timer mode].



[1] Select [A0] as the timer.
[2] Select [f8] for the internal count source.
[3] Specify the period as 1000 micro sec.
[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.

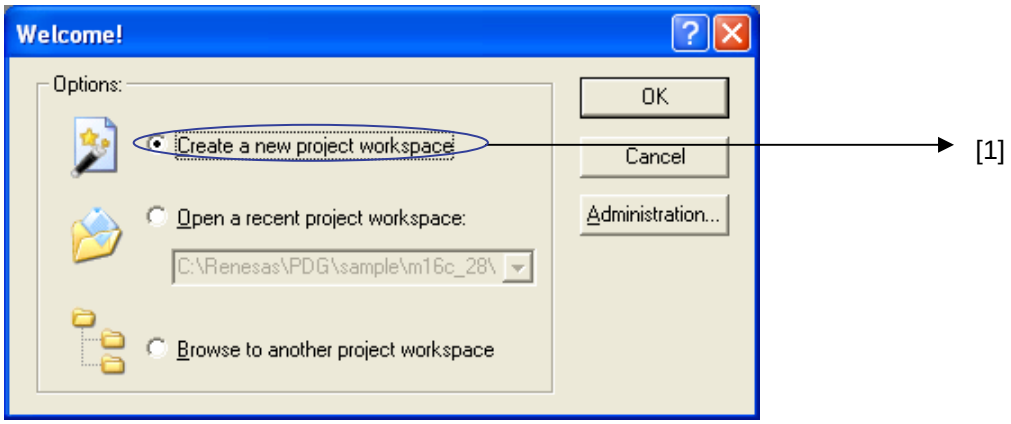


After the settings of the timer mode are completed, the settings are displayed in [1], and the generated source files, generated function names, and functional explanations are listed in [2].

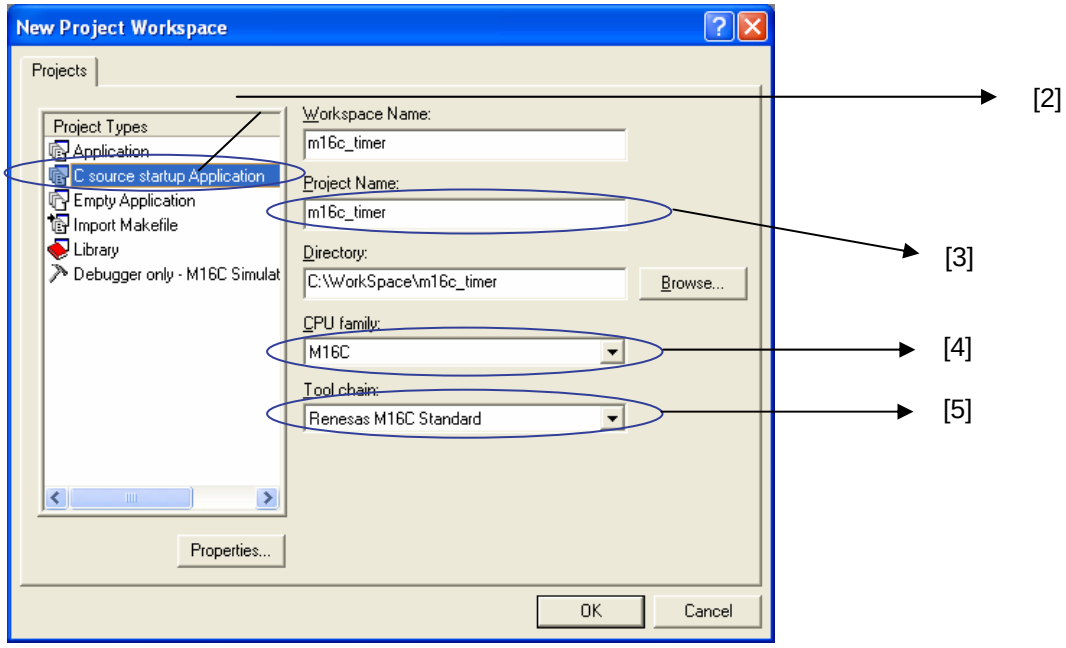
4.3 Creating a Program

A program is created with the HEW.

4.3.1 Creating a Workspace



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



[2] Select [C source startup Application].

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

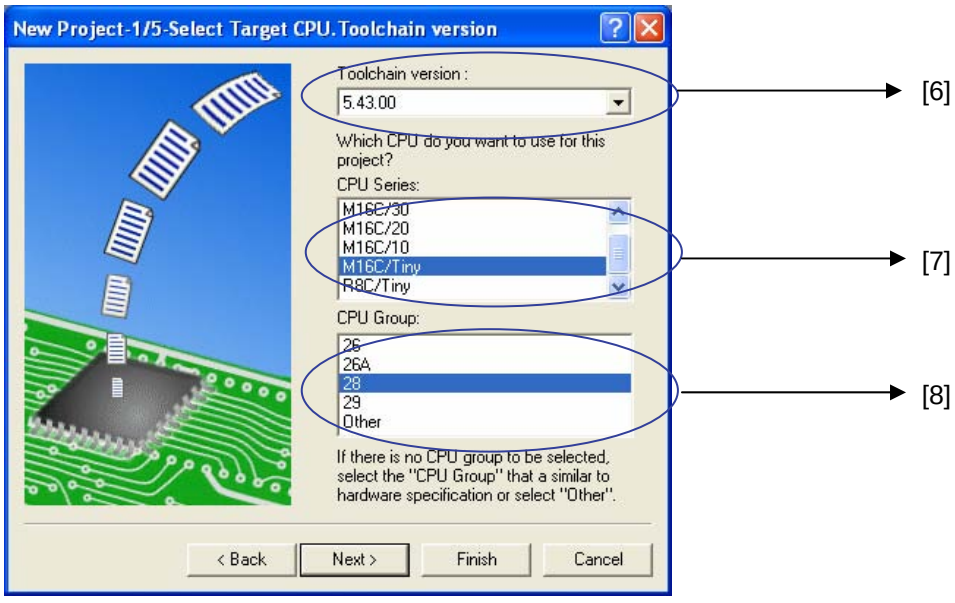
[3] Enter a workspace name.

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:].

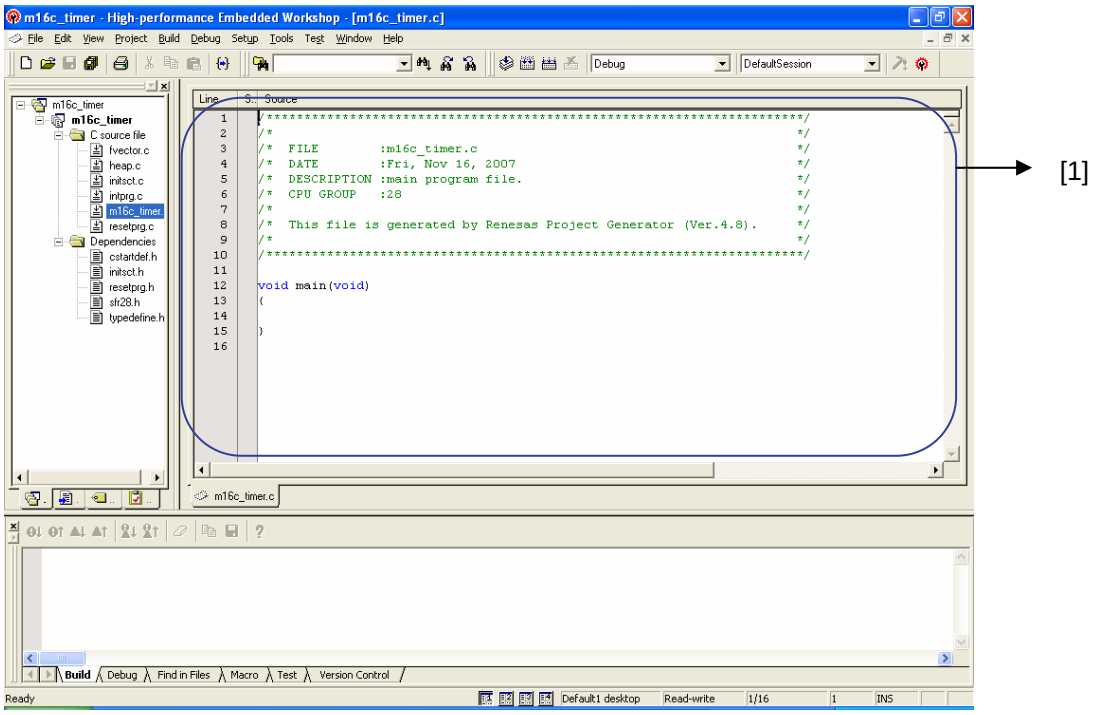
[8] Select [28] for [CPU Group].

Click the [Next] button to proceed.

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

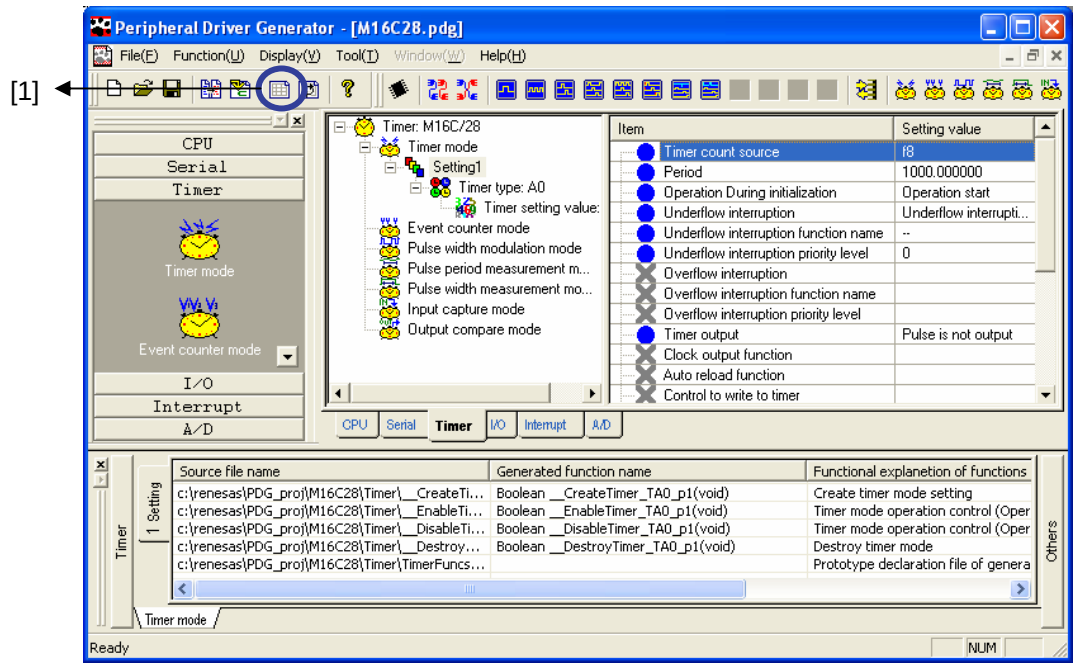
4.3.2 Creating a Program

Code a program on [1] of the HEW.



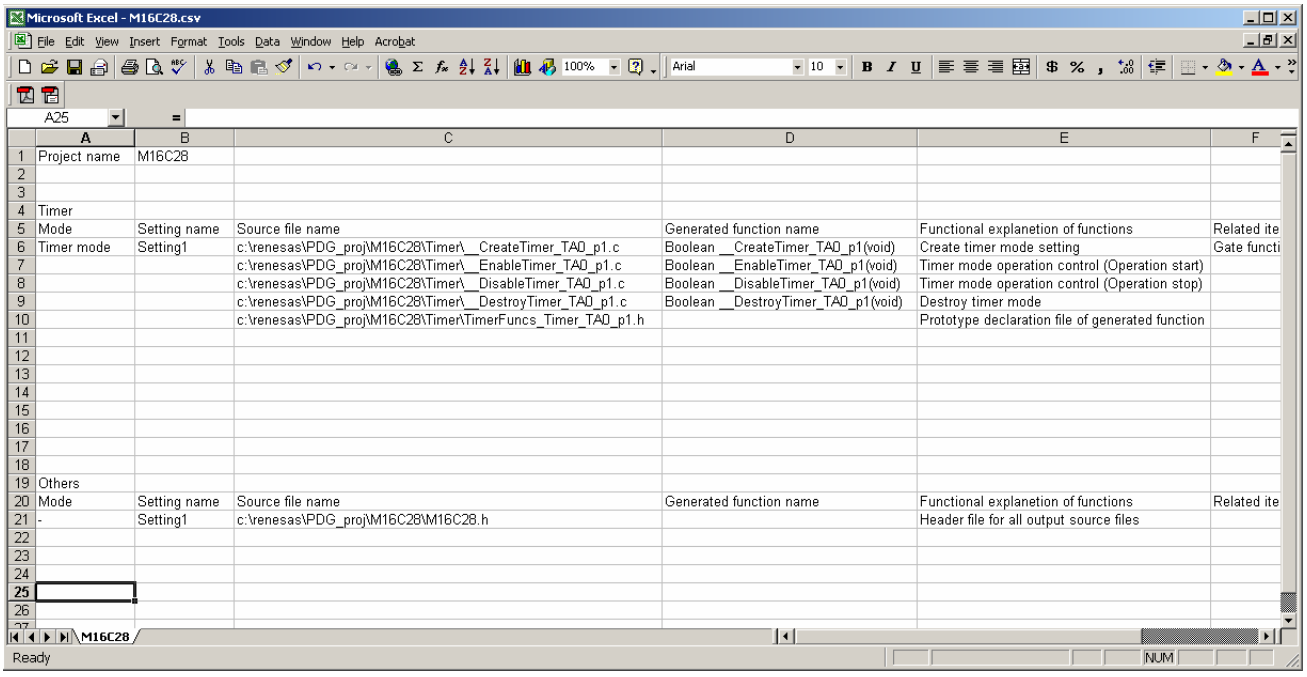
When coding a program, use the functions created with the PDG.

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.)



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;
        plc0 = 0x11;                /* 2 multiplying */
        pm20 = 0;                   /* 2 wait */
        plc07 = 1;                  /* PLL operation */

        for (i = 0; i < PLL_WAIT_CNT; i++) {          /* about 20ms wait */
                for (j = 0; j < PLL_WAIT_1MS; j++) {  /* Main clock 10MHz */
                }
        }
        cm11 = 1;
        prcr = 0x00;                /* protect register on    */

        prcr = 0x04;                /* protect register off */
        pacr = 0x03;                /* 80pin type */
        prcr = 0x00;                /* protect register on */
↓
[1]
```

```
[1]
↓
        p0 = 0xff;
        p1 = 0xff;
        pd0 = 0xff;
        pd1 = 0xff;

        if(   __CreateTimer_TA0_p1( ) == RAPI_TRUE    )          /* timer setting */
        {
                if( __EnableTimer_TA0_p1(   ) == RAPI_TRUE    )  /* timer start */
                {
                        while( 1 )
                        {
                                while( ( ta1ic & 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 */
```

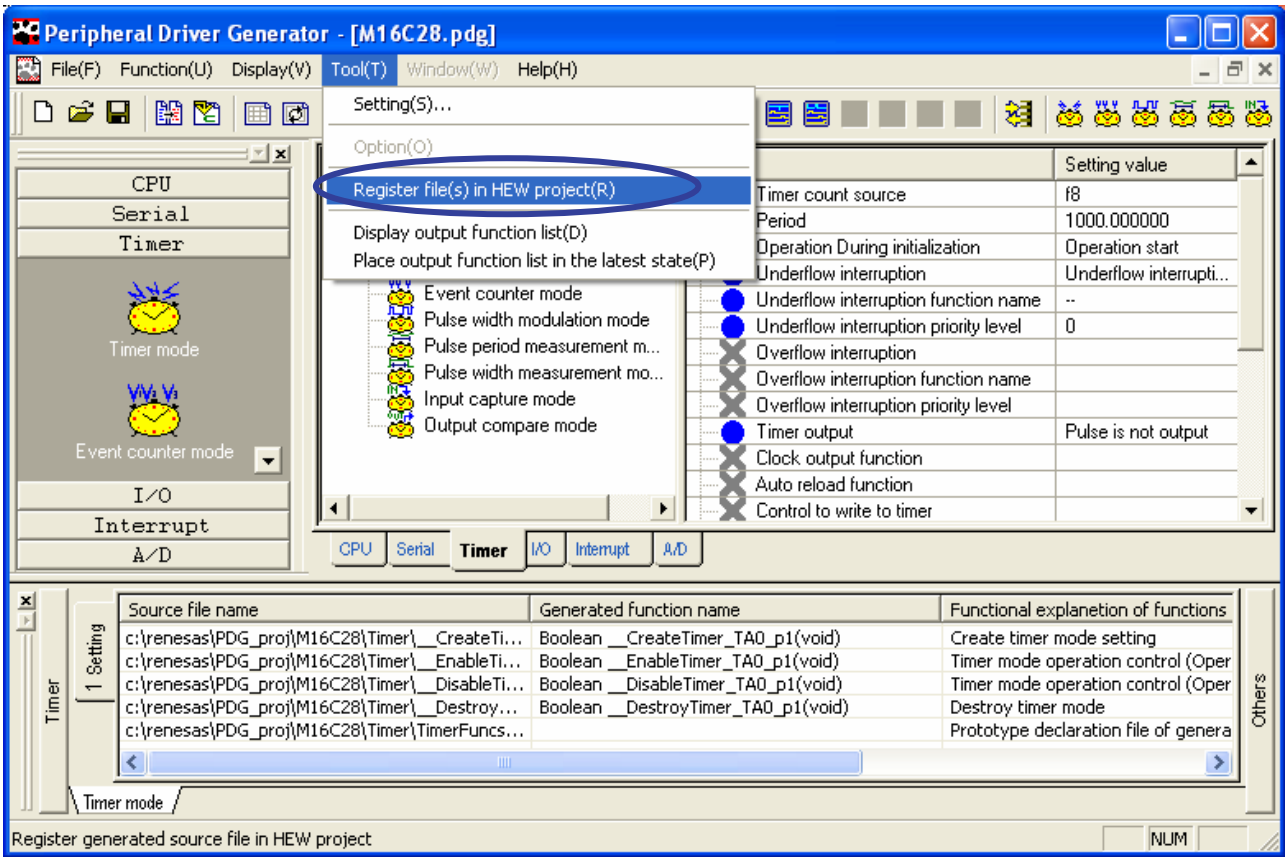
4.4 Build Work

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

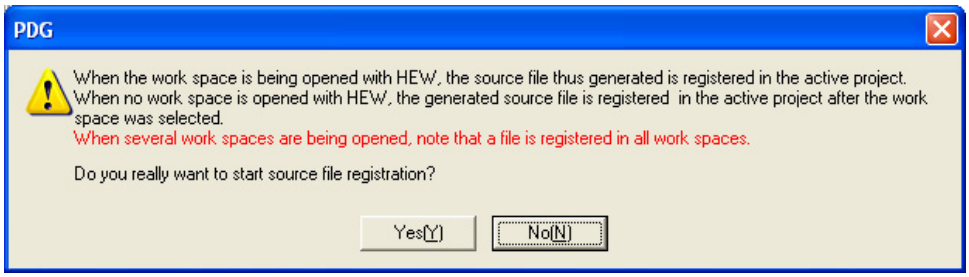
- Registering the files generated by the Peripheral Driver Generator in the High-performance Embedded Workshop.
- Setting the options necessary for a compile (specifying the header file include destination)
- Setting the options necessary for a link (specifying Renesas Embedded Library)

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.

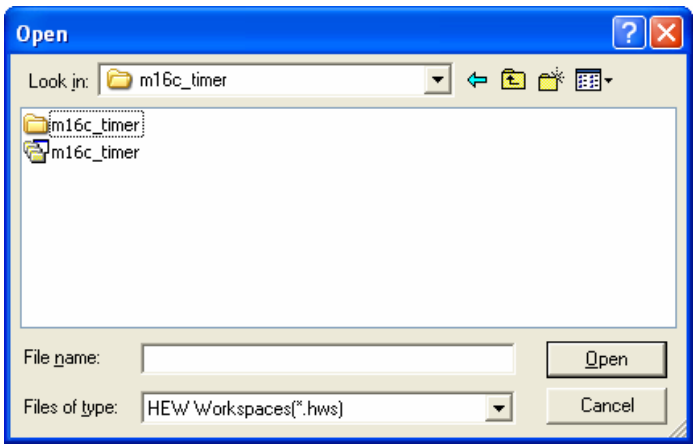


Select [Tool] -> [Register file(s) in HEW project].



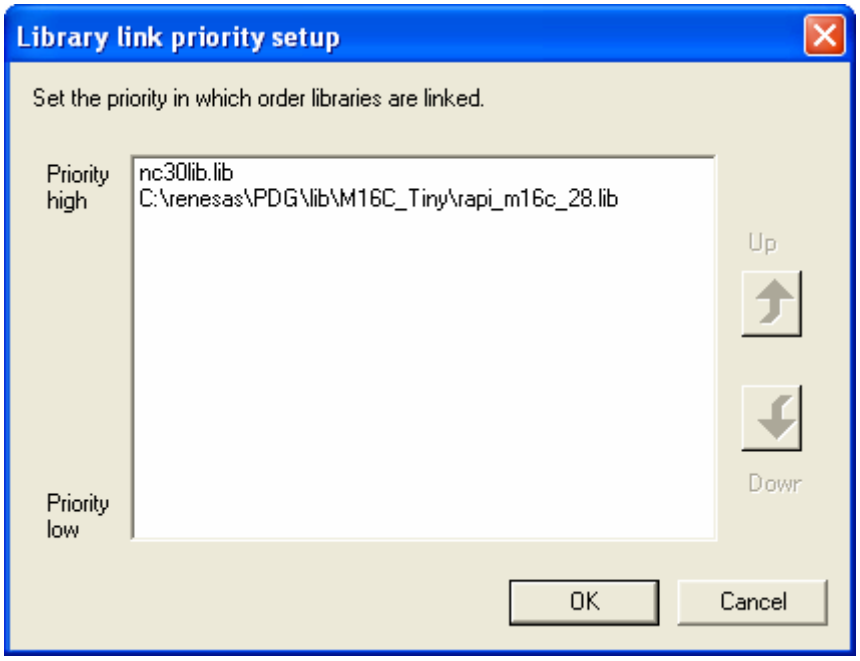
The dialog box will appear asking whether to register the files or not. Select [Yes].
The files will be registered in the project currently opend by the HEW.

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 regstration. Select one in the dialog box.

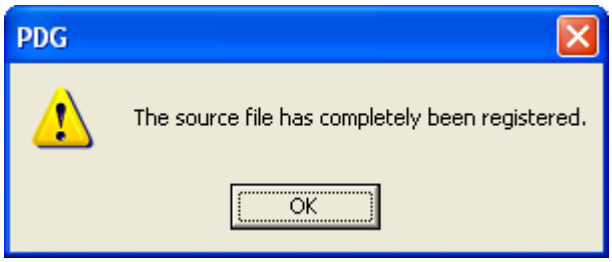


When HEW V.4.02 or later is used, the [Library link priority setup] dialog box appears. Move the libraries up and down in the dialog box, according to their priorities. When [OK] is clicked, the files begin to be registered in the HEW project that you selected. *

* 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.



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



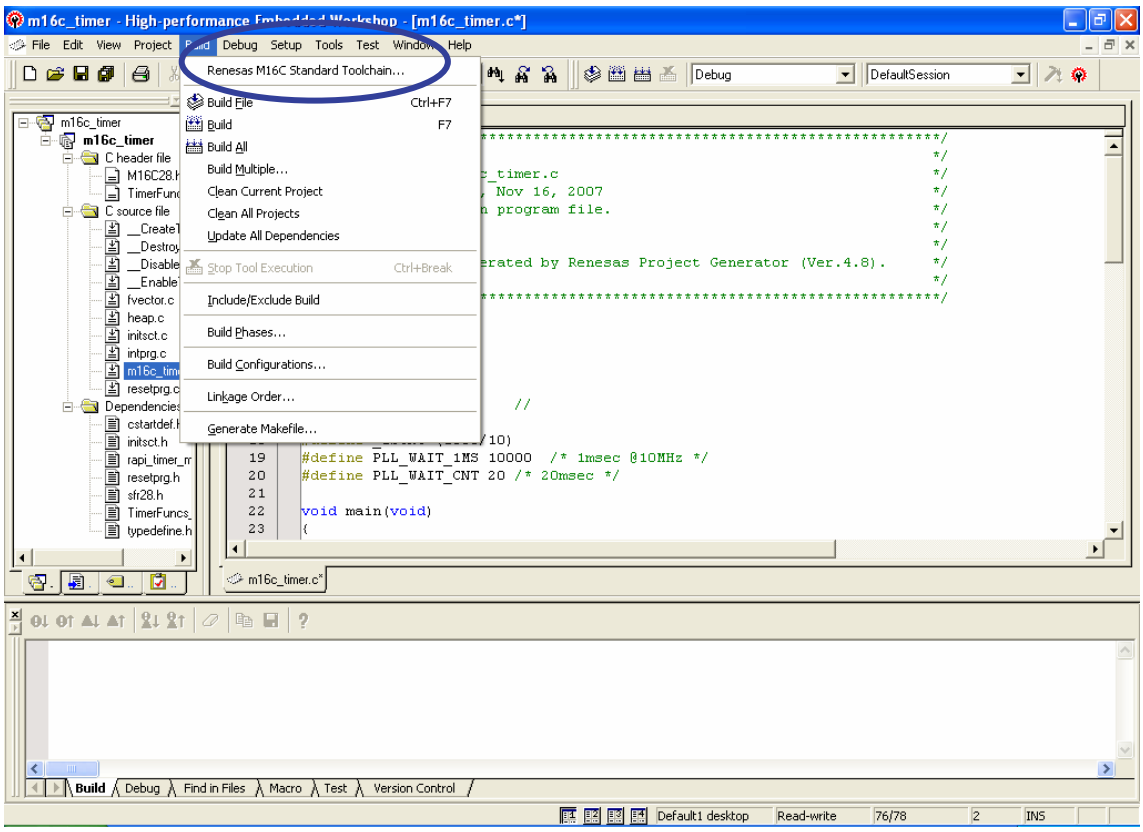
4.4.2 Setting Compile Options

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

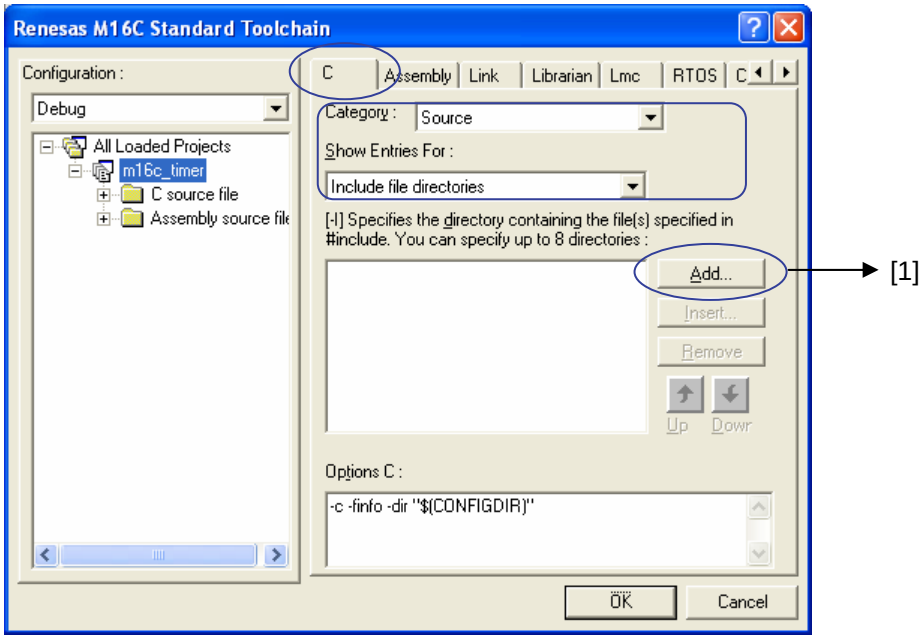
```
#include "sfr28.h"

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

Compiling source files that includes this header file requires to specify its directory. The method to specify the include file directory is as follows.
When setting H8/300H Tiny, H8S/Tiny, or SH/Tiny with HEW V.4.05 or later, this operation is not necessary for the source file registered by PDG or the file already registered because this option is specified automatically with source registration.



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

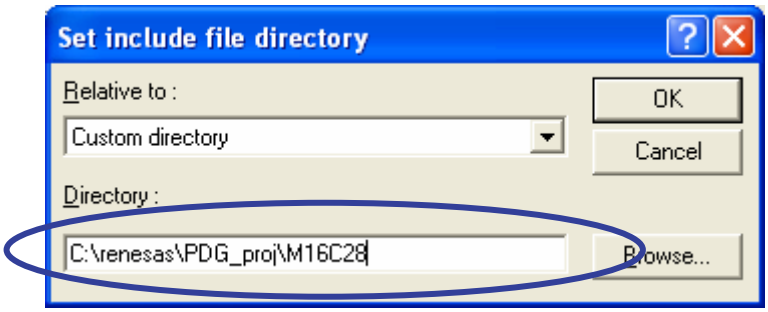


Select the [C] tab in the option dialog box and select the followings:

Category: Source

Show Entries For: Include file directories

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



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

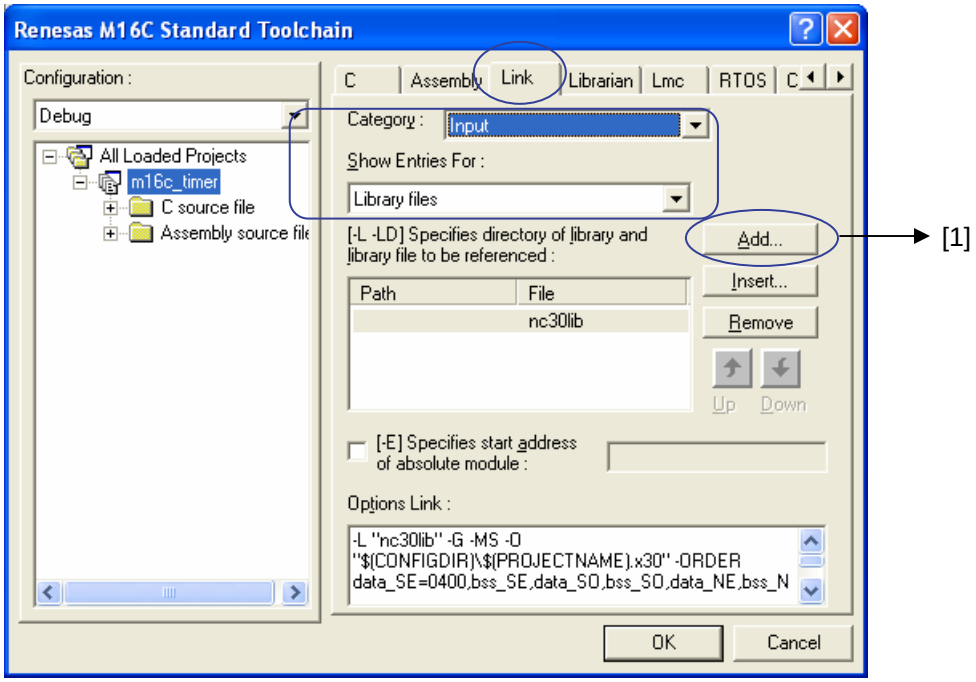
Click the [OK] button to complete the settings.

Note : When setting M16C/Tiny or R8C/Tiny with HEW V.4.05 or later, the message “Failed to set the include file option.” appears depending on the toolchain version. If this message appears, specify the include file directory by the above procedure.

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.



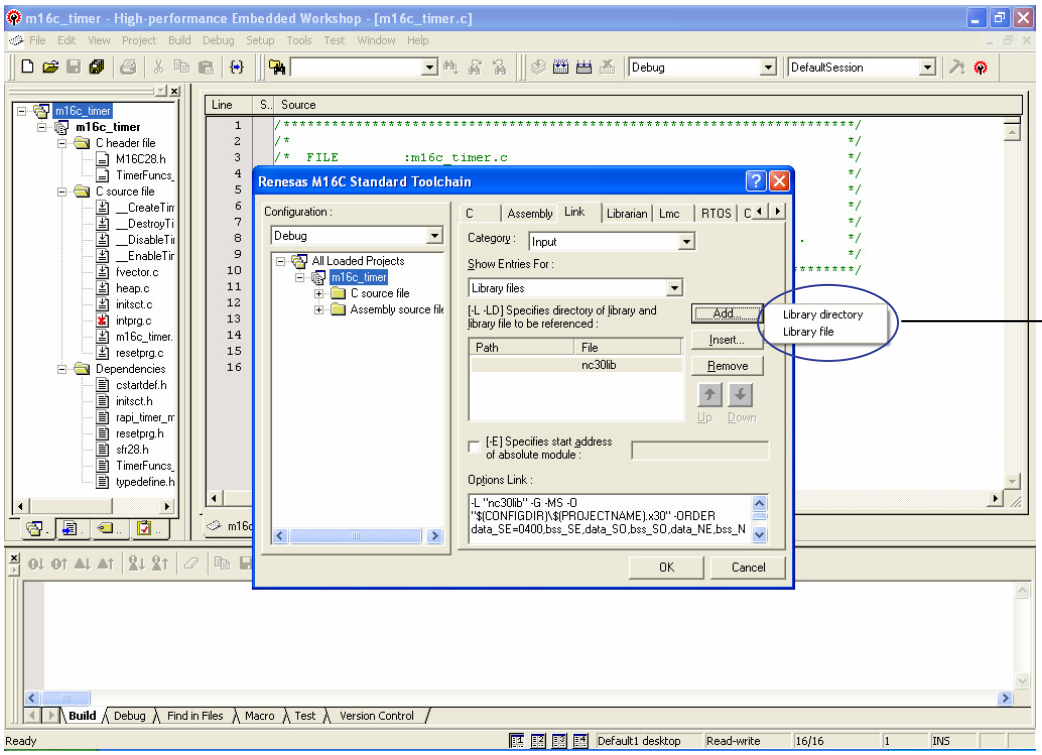
Select the [Link] tab in the option dialog box and select the followings:

Category: Input

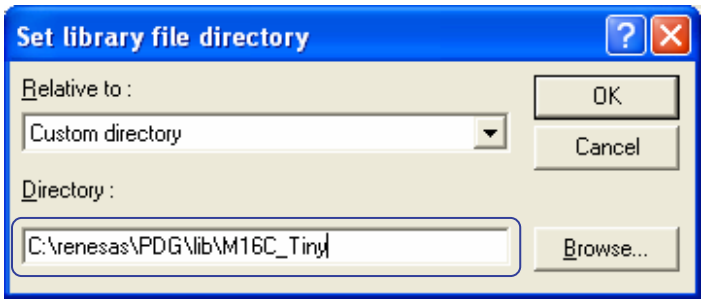
Show Entries For: Library files

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

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



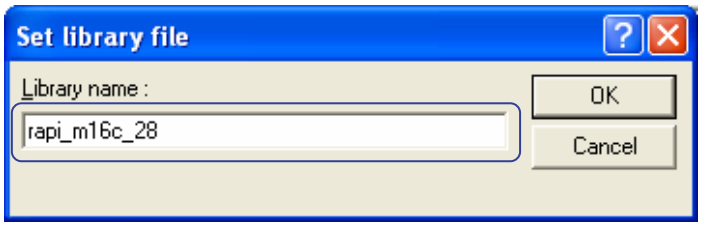
Select [Library directory] from [1].



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].

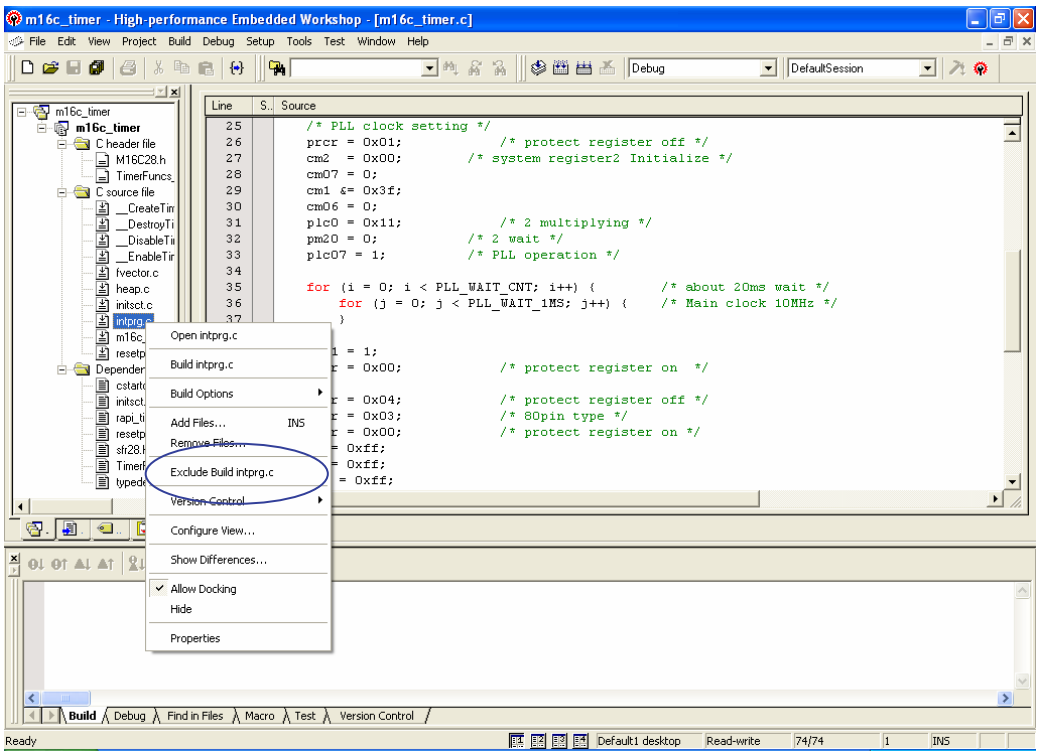
(3) Specify the API library name.
Click [Add] again and select [Library file] in [1].



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.

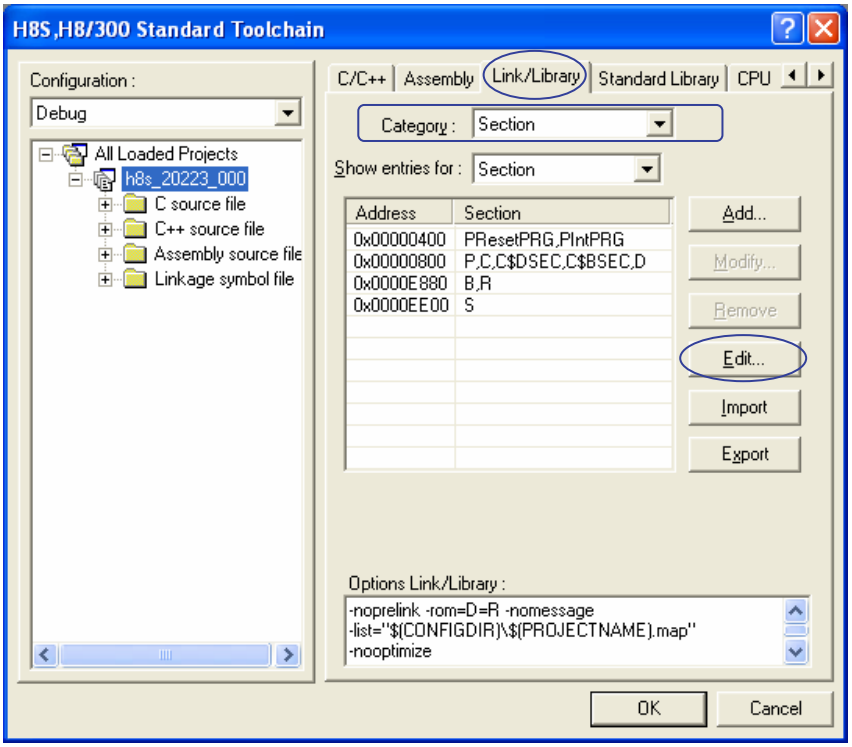


Select intprg.c and right-click on it.
The pop-up menu opens. Select [Exclude Build intprg.c].

4.4.5 Preventing Overlapping of the DTC Vector Table and Sections (for H8S/Tiny)

When you use the source code generated through the PDG to operate the DTC, the area from 400 to 4DD (at maximum) will be used as the DTC vector table. If any section overlaps with the DTC vector table, the address of the section must be changed.

Select [Build -> H8S,H8/300 Standard Toolchain...] to open the build options dialog box. Then select [Section] for [Category] on the [Link/Library] page. A list of sections is displayed. Change the addresses of the sections as required.



All the settings are completed.

Perfom a build and execute the application by using the debugge

.

Peripheral Driver Generator V.1.04 Guide Book

Publication Date:	May. 29, 2009 Rev.1.00
Published by:	Sales Strategic Planning Div. Renesas Technology Corp.
Edited by:	Microcomputer Tool Development Department Tool Business Division Renesas Solutions Corp.

Peripheral Driver Generator V.1.04
Guide Book

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

REJ10J2019-0100