ID Code Retrieval Methods for On-Chip Debugging Emulator E8/E8a/E1/E20

(applies to M16C, incl. R32C, and R8C Family MCUs)

Renesas Electronics Corporation
MCU Tool Product Marketing Department

2011/06/14

ID Code Check Function Overview

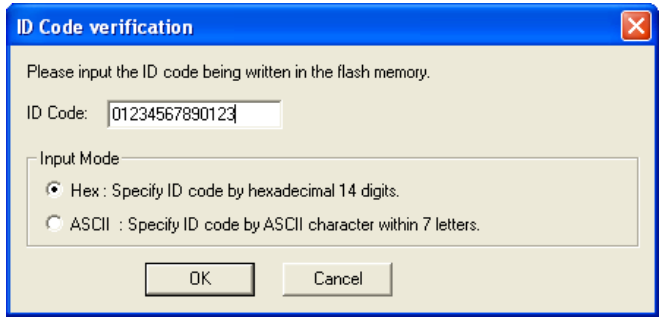
- The ID code check function in M16C and R8C Family MCUs prohibits execution of read and erase operations to the MCU built-in flash memory.
- After the ID code is written to the built-in flash memory by the user, the user must input the same ID code in order to use a debugger or a flash programmer. If the input ID code and the ID code written to the Flash memory do not match, the flash memory cannot be read or erased.

※Some newer devices have a forced erase function; check the hardware manual of the MCU in use to find out if this function is supported.

- Because this is a built-in function, if the user forgets the ID code after writing it to the Flash memory, there is nothing Renesas can do to retrieve it.
- However, if the user knows which program was last downloaded, the ID code set for that program can be retrieved.
  This document introduces the method for retrieving an ID code in such circumstance.

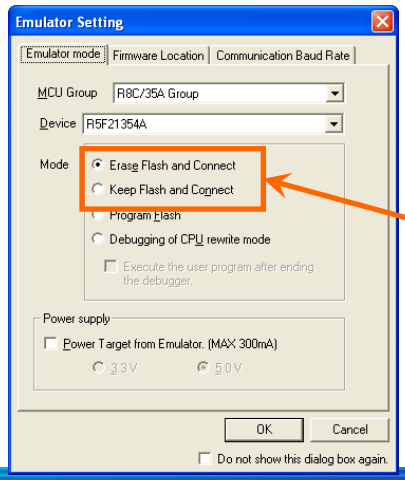
How the ID Code for E8/E8a/E1/E20 Debugger works

- The ID Code Check Function is built-into the MCU to prevent the Flash memory from being easily read or overwritten with an on-board programmer, etc. The 7-byte ID code previously written to the Flash memory is input by the user in the dialog to the right. (**This is not the dialog for setting the ID code.)**



※When the ID code written to the Flash memory is all FF, the debugger automatically checks the ID.

The ID code written to the Flash memory is the value first set in the user program when the program is downloaded. However, when a program is downloaded in a debug mode (the first 2 modes listed in the E8/E8a startup dialog/ [debug mode] in the E1/E20 startup dialog), the user ID code is not reflected in the Flash memory and is always all FF. ※E1/E20 debugger supports only some MCU groups in R8C family.



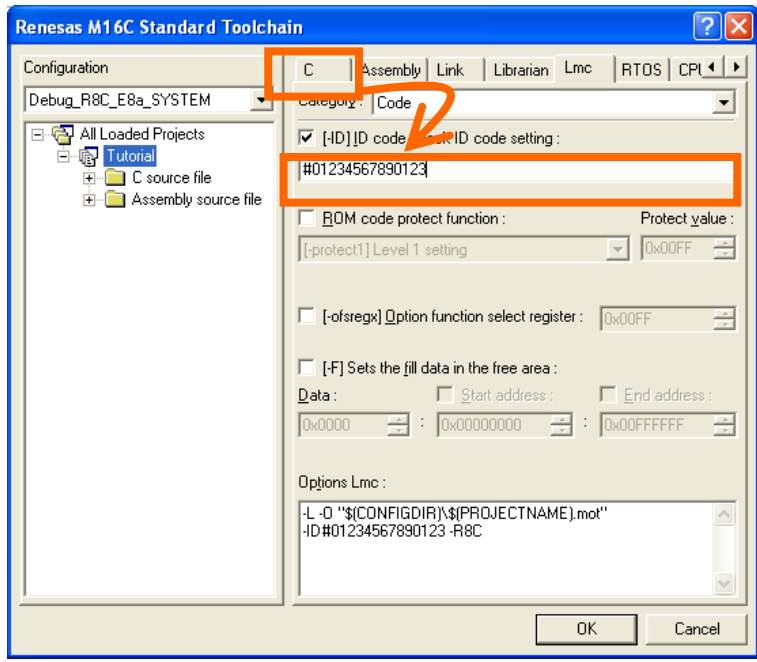
※Left figure shows the dialog for E8a debugger

Called “Download emulator firmware” and “Does not emulator firmware”, respectively, in version previous to E8 V.2.09

	R32C	M16C	R8C
ID code 1st byte	FFFFFFE8h	FFFDFh	FFDFh
ID code 2nd byte	FFFFFFE9h	FFFE3h	FFE3h
ID code 3rd byte	FFFFFFEAh	FFFEbh	FFEbh
ID code 4th byte	FFFFFFEBh	FFFEFh	FFEFh
ID code 5th byte	FFFFFFECh	FFFF3h	FFF3h
ID code 6th byte	FFFFFFEDh	FFFF7h	FFF7h
ID code 7th byte	FFFFFFEEh	FFFFBh	FFFBh

How to Set the ID Code

- The ID code can be set with any of the 3 following methods:
  - Use an Assembler directive command [.ID]【**Recommended**】
    Example: Set the ID code to 0x1234567890abcd
    (1) Assembler source
    .id “#1234567890abcd”
    (2) C source
    _asm(“ .id “”¥” #1234567890abcd¥””);
  - Use the load module converter LMC30 [-ID] option
    Select HEW menu [Build] →
    [Rensas M16C Standard Tool Chain]
    then set the ID code in the load module converter tab in the dialog shown on the right.



※The C/C++ compiler package for the M16C series and the R8C family (M3T-NC30WA) V.6.00 Release 00 and later do not support –ID option, so you can not use this method.

- Set directly with Assembler
  This method sets the ID code to the corresponding high level position when defining the fixed vector as follows:
  org 00FFDCH
  .lword int_und | (55000000h) ; UND
  :

How to Set the ID Code

Method 1:

Please note that the manner in which values are reflected to x30 and mot files differs based on use of the ".ID" Assembler instruction method or use of the "-ID" option of the LMC30 command. Therefore, we recommend using the ".ID" Assembler instruction to set the ID code.

Specifying with .ID	The value is reflected in both x30 and mot files.
Specifying with –ID	The value is only reflected in the mot file.

Due to this difference, if the "-ID" option is used and a write operation is executed to the Flash using the x30 file, the ID codes set in the user program will not be reflected in the actual chip, causing problems in programming later.

※The C/C++ compiler package for the M16C series and the R8C family (M3T-NC30WA) V.6.00 Release 00 and later do not support –ID option, so this note does not apply to you if you use these.

Method 2:

If the user does not specify an ID code, the highest address in the fixed interrupt vector is allocated to the ID code, which is normally zero. If the vector that stores the ID code is not described in the user program, the initial value of the Flash will be FF.

When the corresponding fixed vector is declared → the corresponding ID code bytes become 00.
When the corresponding fixed vector is declared → the corresponding ID code bytes become FF.

How to Set the ID Code

Method 3:

Register settings for selecting ID code and option functions using Assembler directive commands ".id" or ".ofsreg" may not be executed successfully due to certain complier restrictions.

※The following restrictions are improved on the C/C++ compiler package for the M16C series and the R8C family (M3T-NC30WA) V.6.00 Release 00 and later, so this note does not apply to you if you use these.

FAQ 104698 The following restrictions apply to source program file names and directory names:

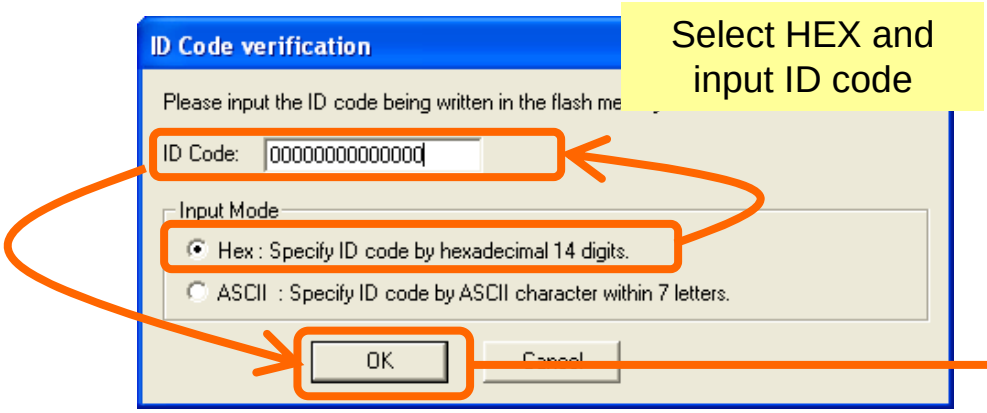
- Only ASCII characters can be used for directory names, file names, and workspace names.
- For example, kanji and katakana cannot be used.
- Blank spaces cannot be used in directory names in version older than V.5.40 Release 00 of M3T-NC30WA, M3T-NC308WA
- Only one period [.] can be used in a file name.
- Network path names cannot be used. Please assign a drive name.
- Shortcuts cannot be used.
- "..." cannot be used to specify two or more directories.
- Files names including a path specification cannot be longer than 128 characters.

If any of the restricted items are used, one of more of the following phenomena may occur.

- A value set using Assembler instruction command .id, .orfreg, protect, .rvector or .svector will not operate normally. This may result in problems occurring when trying to correctly set the ID code or options function selection register value.
- Call Walker or STKViewer, which reference the stack size usage amount, will not be displayed correctly.
- MAPViewer, which references the map information of the absolute module file, will not be displayed correctly.
- The contents set by one of the above Assembler instruction commands will not be displayed in the ".map" file.
- "Can't open file" or other compiler errors may occur.
- The linker will output the "Due to an error lnxx.exe will be ended due to an error" message end abnormally.
- The variable vector table automatic create function will not operate normally.

How to retrieve a lost ID code

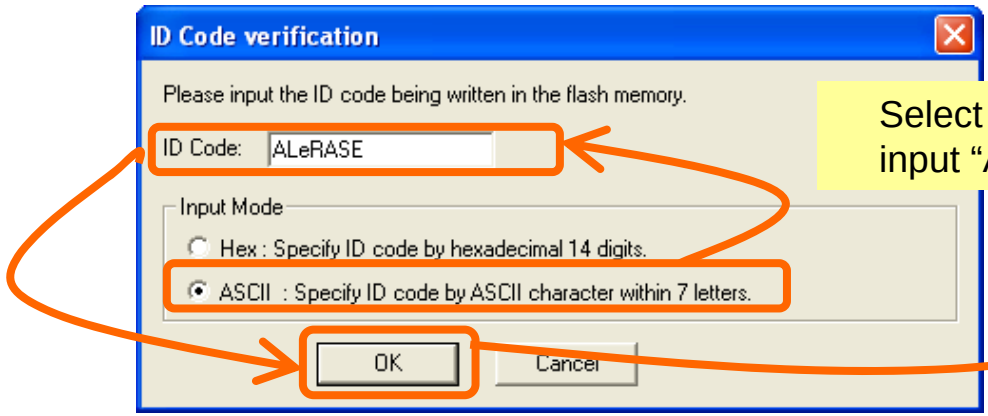
- Try 0x00000000000000
  →If an ID code has not been set, it will sometimes be assigned as 0x00000000000000.



Select HEX and input ID code

If the ID codes do not match, this dialog will reappear.
In this case, cancel the dialog, power down the user system and then restart it.
(In general the system will not allow repeated input of different IDs.)

- If the MCU has a *forced erase function*, try inputting the ID code for a forced erase: AleRASE



Select ASCII and input “ALeRASE”

How to retrieve a lost ID code

- Using the Assembler instruction command ".ID" or load module converter LMC30 "-ID" option when the last downloaded program is known:
  → In this case, an ID file with the ".ID" extension was created. Reference the contents of this file to retrieve the ID code.

- When the last downloaded program is know but the type of ID code set is not known:
  → Try the methods described on the following pages.

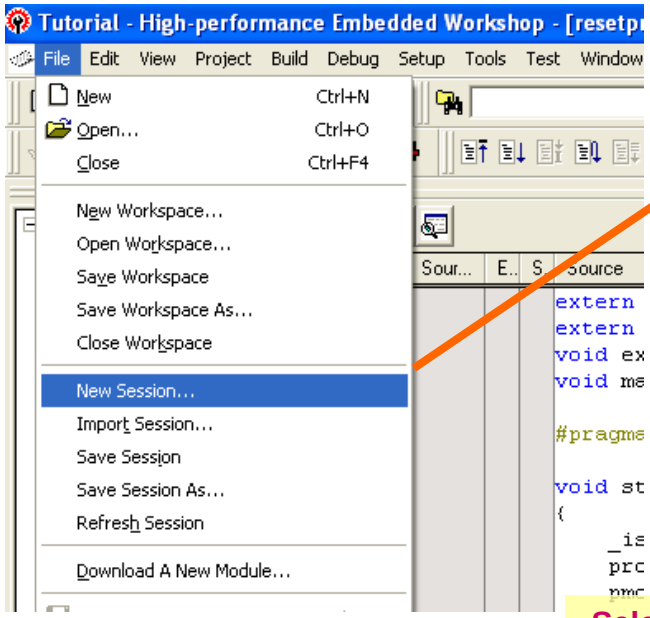
If none of these suggestions work, there is nothing else Renesas can do to retrieve the ID code.

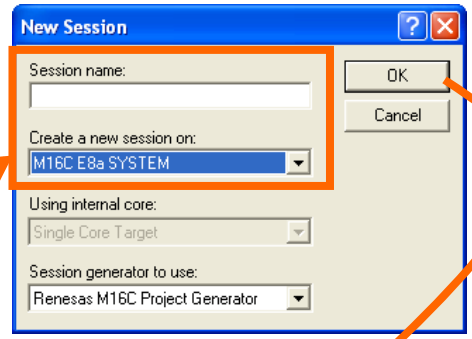
How to confirm an ID code with a simulator

- If you know the last program that was downloaded, you can read that program with a simulator, look at the area the ID code is stored in, and confirm the status of the ID.

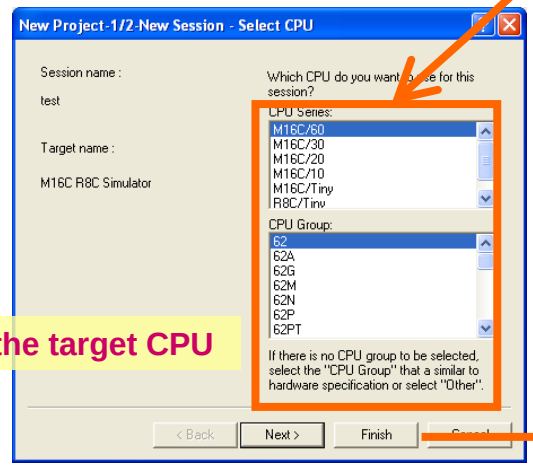
Step (1) Add a simulation

Add a simulation to the created workspace, then go to step (3).
If there is already a simulator session set up (SessionM16C_R8C_Simulator, SessionR32C_100_Simulator), proceed to step (2).

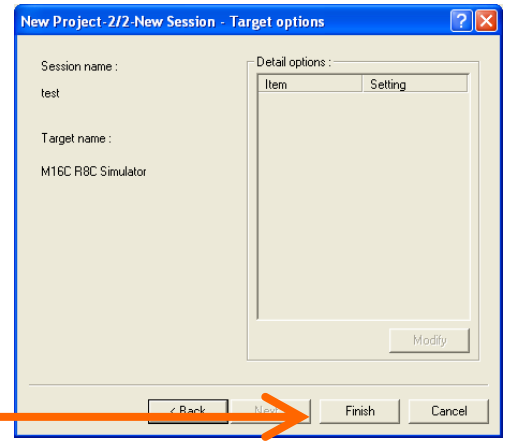




Input the session name and select the target of the simulator session.



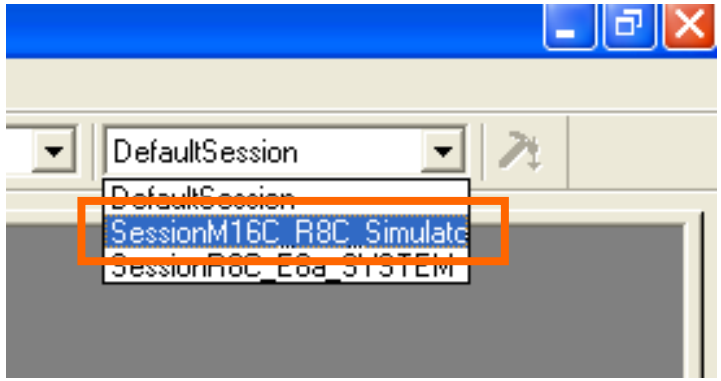
Select the target CPU



How to confirm an ID code with a simulator

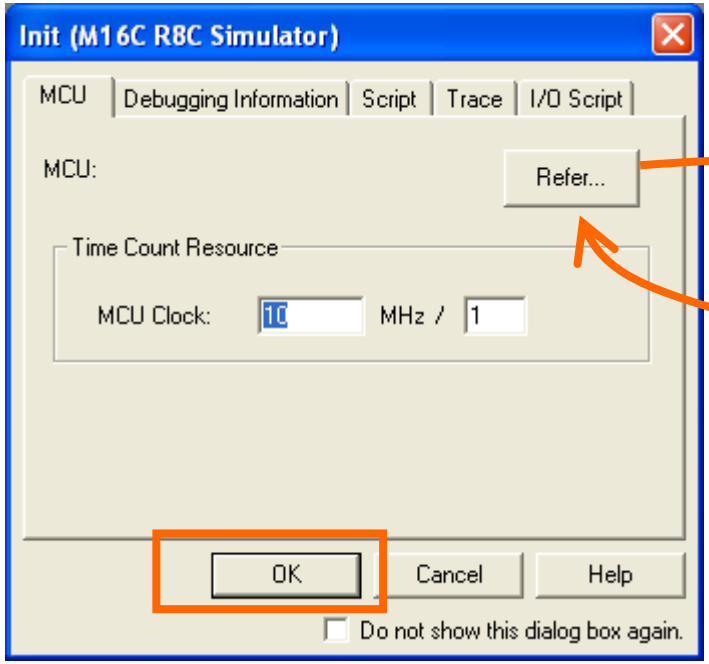
Step (2) Switch sessions

When there is already a simulator session set, switch the HEW session to the simulator session (SessionM16C_R8C_Simulator,SessionR32C_100_Simulator).

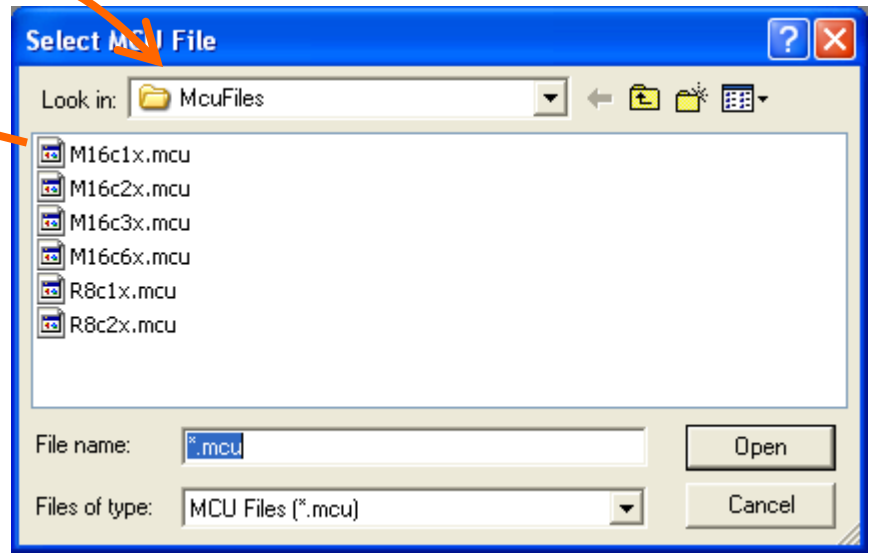


How to confirm an ID code with a simulator

Step (3) Simulator Settings
The following INIT dialog will be appear when you have created a simulator session or switched sessions.



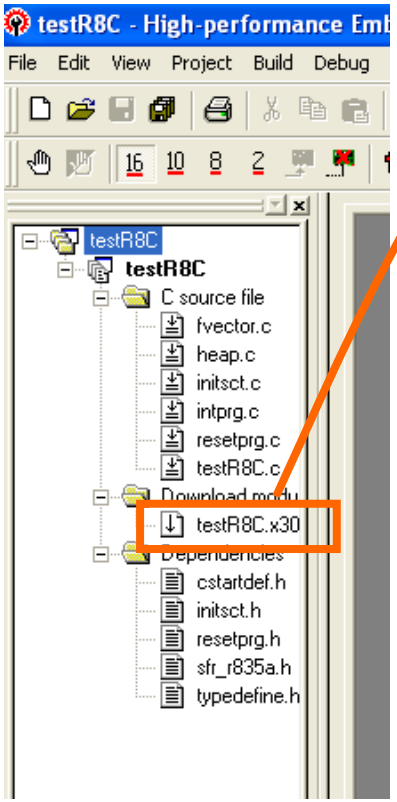
Click the Refer... button to display a list of MCU files. Select the target MCU.



Select the MCU file, then click OK.

How to confirm an ID code with a simulator

Step (4) If the user program was written to the MCU using an x30file, the memory window can be used after downloading the x30 file to confirm what was written in the ID code area. →If a mot file was used, refer to step (5).



After downloading the file, open the memory window to confirm the ID value.

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00FF60	00	C0	00	00	00	C0	00	00	00	C0	00	00	00	C0	00	00
00FF70	00	C0	00	00	00	C0	00	00	00	C0	00	00	00	C0	00	00
00FF80	00	C0	00	00	00	C0	00	00	00	C0	00	00	00	C0	00	00
00FF90	00	C0	00	00	00	C0	00	00	00	C0	00	00	00	C0	00	00
00FFA0	90	C0	00	00	96	C0	00	00	00	C0	00	00	00	C0	00	00
00FFB0	00	C0	00	00	00	C0	00	00	00	C0	00	00	00	C0	00	00
00FFC0	00	C0	00	00	00	C0	00	00	00	C0	00	00	00	C0	00	00
00FFD0	00	C0	00	00	00	C0	00	00	FF	FF	FF	FF	00	C0	00	01
00FFE0	00	C0	00	23	00	C0	00	00	00	C0	00	45	00	C0	00	67
00FFF0	00	C0	00	89	00	C0	00	AB	00	C0	00	CD	04	C1	00	FF

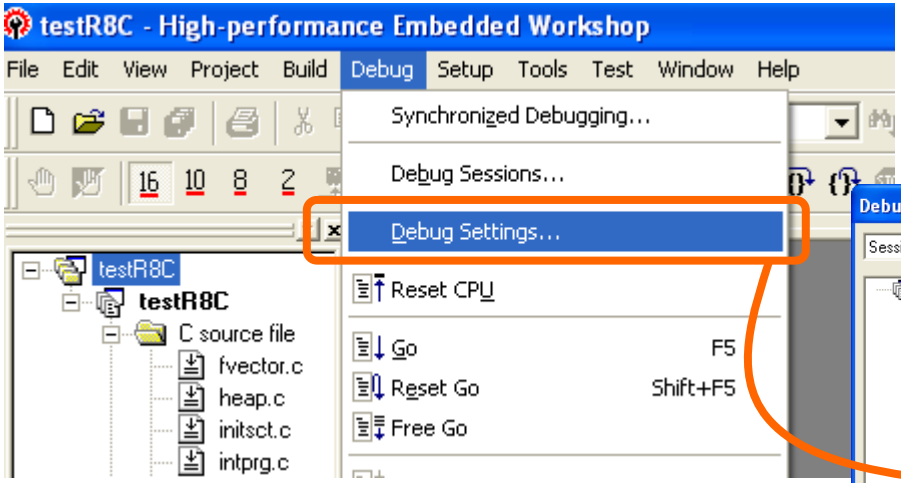
This example shows the case of R8C. The red characters are the ID code. In this case, the code is “0x1234567890abcd”.

※ Fill the ID code area with FF (initial Flash value) before downloading.

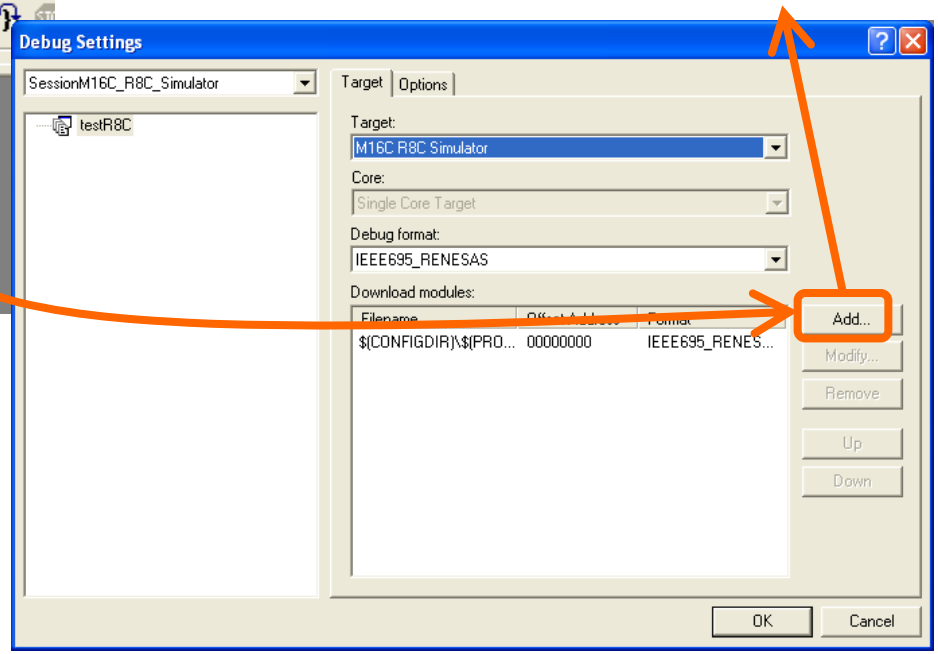
How to confirm an ID code with a simulator

Step (5) If a mot file was used to write the program, the mot file must first be registered in the HEW.

- Go to HEW “Debug” → “Debug” setting to open the debug selection dialog, then click “Add.”



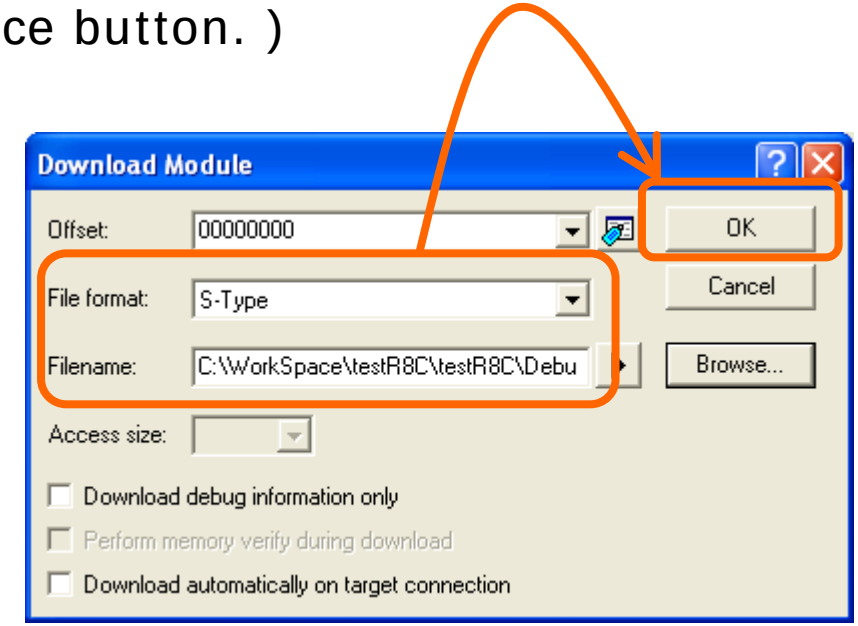
Click “Add” to open the download module dialog (see next page).



This setting is stored in the project and only needs to be set once.

How to confirm an ID code with a simulator

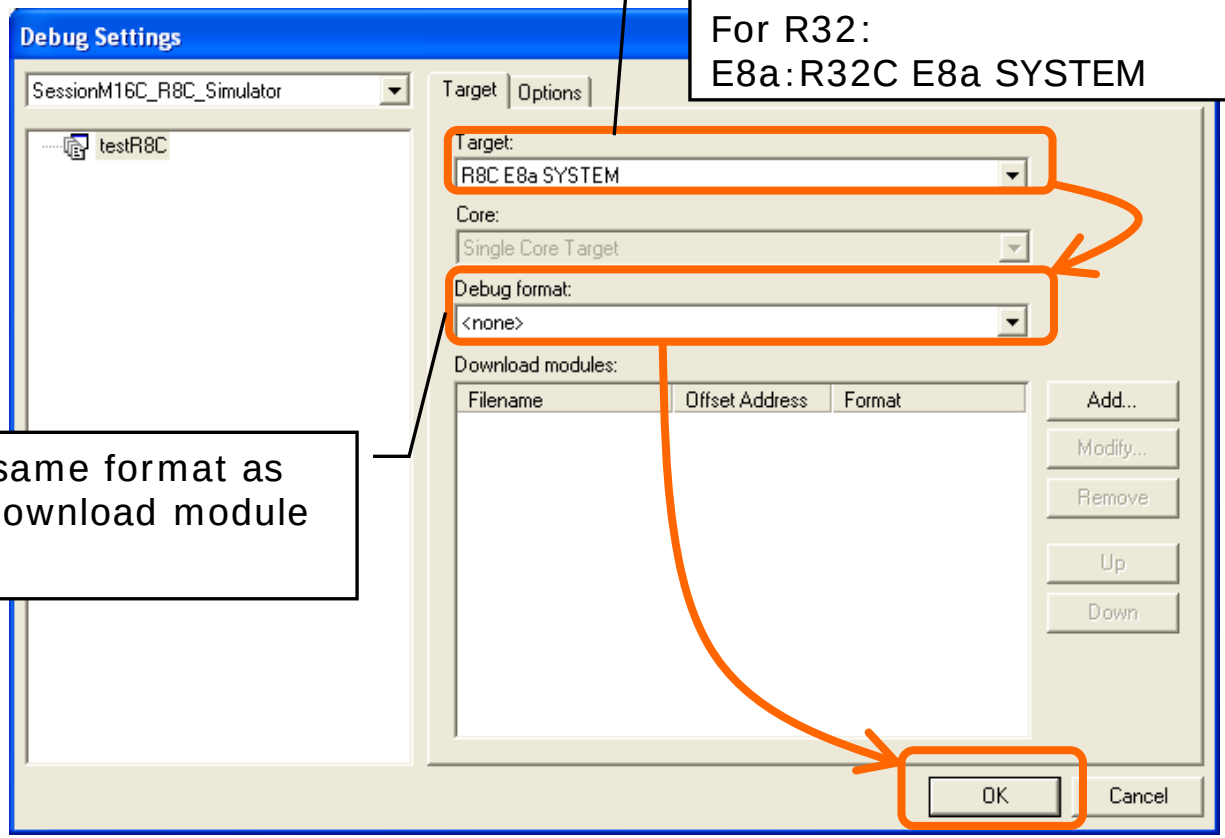
- Set the download module dialog as shown below.
  - Format: S-Record
  - File name: the downloaded Motorola S file name (select using the reference button.)



How to confirm an ID code with a simulator

- Return to the Debug Setting dialog and set the following:

Confirm the settings as follows:
For R8C/Tiny:
E8:R8C E8 SYSTEM E8a:R8C/E8a SYSTEM
E1/E20: R8C E1/E20 SYSTEM
For M16C:
E8:M16C E8 SYSTEM E8a:M16C E8a SYSTEM
For R32:
E8a:R32C E8a SYSTEM



Select the same format as set in the download module dialog.

Renesas Electronics Corporation