RENESAS Tool News

RENESAS TOOL NEWS on February 16, 2012: 120216/tn6

Note on Using Integrated Development Environment High-performance Embedded Workshop

When using High-performance Embedded Workshop, take note of the following problem:

- With selecting build options

High-performance Embedded Workshop is included with the compilers, emulator debuggers, and other software products that it manages.

1. Product and Versions Concerned

High-performance Embedded Workshop V.4.09.00 and later

2. Tools Involved

When used together with the product concerned, either of the following compilers are involved in this problem:
- C/C++ compiler package for SuperH family V.9.04 Release 01
- C/C++ compiler package for RX family V.1.02 Release 00

3. Description

If you specify the linking order of modules by using the function for customizing linking orders, parameters of the binary or input option of the optimizing linkage editor (linker) selected in High-performance Embedded Workshop may not properly be passed to the linker during build. As a result, object files may not correctly be created.

4. Conditions

This problem arises if either of the following is satisfied:

(1) For the binary option

The following procedures are performed to select a binary file to be inputted to the linker:

(1.1) On the Build menu, select the ***** Standard Toolchain command to open the ***** Standard Toolchain dialog box. Here, ***** denotes SuperH RISC engine or RX according to your compiler.

(1.2) Select Input from the Categories list in the Optimizing Linker tab.

(1.3) Select Binary Files from the Show Entry for pull-down list.

(1.4) Click Add button to open the Add binary file dialog box.

(1.5) Type the binary file's name into File Path and the section's name into Section.

Example:

If you type "f.bin" into File Path and "sec2" into Section, the section's name is not passed to the linker as shown below, and then a link error occurs during build:

```
-binary="f.bin"
```

The correct output is:

```
-binary="f.bin"(sec2)
```

(2) For the input option

The following procedures are performed to select the library file, in which the module's name is set, to be inputted to the linker:

(2.1) On the Build menu, select the ***** Standard Toolchain command to open the ***** Standard Toolchain dialog box. Here, ***** denotes SuperH RISC engine or RX according to your compiler.

(2.2) Select Input from the Categories list in the Optimizing Linker tab.

(2.3) Select Relocatable Files/Object Files from the Show Entry Pull-down list.

(2.4) Click Add button to open the Add relocatable file or object file dialog box.

(2.5) Type the library file's name into File Path and the module's name within the library file into Module Name.

Example:

If you type "test.lib" into File Path and "mod" into Module Name, the option's name "library" is passed in place of "Input" to the linker during build but the module's name is not passed as shown

below:
-library="test.lib"
As a result, object files are not created correctly.

The correct output is:
-input="test.lib"(mod)

5. Workaround

To avoid this problem, do either of the following:
(1) Change the method of selecting the option concerned as follows:
(1.1) In the ***** Standard Toolchain dialog box, cancel Binary File or Absolute/Relocatable/Library selected from the Show Entry pull-down list in the Optimizing Linker tab.
(1.2) Select Other from the Categories list.
(1.3) Select Absolute/Relocatable/Library from the User Defined Options list.
(1.4) Type the names of the option and its parameters into the edit box of the User Defined Options list.
Examples:
To select the binary option:
-binary="f.bin"(sec2)
To select the input option:
-input="test.lib"(mod)

(2) Disable the function for customizing linking orders as follows:
(2.1) On the Build menu, select the Linkage Order command to open the Linkage Order dialog box.
(2.2) Clear the Use custom linkage order check box.

6. Schedule of Fixing Problem

We plan to include the problem-fixed High-performance Embedded Workshop with the next versions of both compilers.