RENESAS Tool News

RENESAS TOOL NEWS on June 5, 2012: 120605/tn1

Notes on Using CubeSuite+ Integrated Development Environments for RX Family

When using the CubeSuite+ for the RX family of MCUs, take note of the following problems:

- With operating the library generator
- With using compile options and link options (build options)

1. Products Concerned

- The CubeSuite+ for RX Family licensed pack
- The evaluation edition of the CubeSuite+ integrated development environment with the license for the RX family

Use of the common program V1.01.00, V1.01.01, or V1.02.00 to the CubeSuite+ in combination with the CubeSuite+ CC-RX compiler (the C/C++ compiler package for the RX family) provides the basic condition under which the problems described below arise.

2. Problem with Operating the Library Generator

2.1 Description

In the Properties panel of CubeSuite+, if you make either of the following changes of options and then execute the build, the library generator does not operate:

- In the Common options tab, you make a change to the value of the Address value of base register that sets the address value text box.
- In the Library generate option tab, you select the Enables string.h(C89/C99) property and click the drop-down arrow at the right end of the row (at the second column of the row, YES or NO has been selected) to open a drop-down list.
  Then you click YES or NO to make a change to the above selection.

2.2 Workaround

If you have made a change to either of the above options, execute the build after deleting the library file.

To search for the library file, follow these steps:

(1) Go to the directory whose name has been typed in the Path of the output folder text box in the Library generate option tab.

(2) Search for the file whose name has been typed in the Output file name text box. It is the library file searched for.

2.3 Schedule of Fixing Problem

We plan to fix this problem in the next version of the common program to the CubeSuite+.

3. Problem with Using Compile Options and Link Options (Build Options)

3.1 Description

If you execute a build by using an option to optimize the accesses to external variables, incorrect code may be generated.

3.2 Conditions

This problem arises if the following conditions are all satisfied:

(1) The Compile options tab is selected in the Properties panel of CubeSuite+; then the Yes(Optimizes the inter-module)(-map) option is selected in the Optimizes accesses to external variables property in the Optimization category.

(2) The Link options tab is selected; then the Yes(-Map) option is selected in the Outputs the external symbol-allocation information file property in the Output category.

(3) Any of the source files that have been registered on the project is modified.

Example:

Source file main.c

```
-------------------------------------------------------------------
    int a, x, b;  // Definition of variables changed from int a, b;
// Condition (3)
    int   c;

    void main(void)
    {
      c = a + b;
    }
-------------------------------------------------------------------
```

If the build of the above is made, incorrect code is generated from the source file sub.c below, and variables b and c reference to incorrect addresses.

Source file sub.c

```
--------------------------------------------------------------------
 extern int a, b;
 extern int c;
     void sub(void)
     {
      c = a + b; // Variables b and c reference incorrect addresses.
     }
--------------------------------------------------------------------
```

When the options in Conditions (1) and (2) are selected, the build should be executed as follows:

1. The modified source file main.c is compiled.
2. At the first linking, the Outputs the external symbol-allocation information file is updated.
3. By using the updated Outputs the external symbol-allocation information file, all the source files, including main.c and sub.c, are recompiled.
4. The second linking is executed.

If the problem arises, however, only the main.c file (the modified source file) is recompiled in step 3 above. So from main.c, the object file using the updated Outputs the external symbol-allocation information file is generated, and from sub.c, the object file using the original Outputs the external symbol-allocation information file is done.
As a result, main.c and sub.c are different from each other in the allocation information of variables after the second linking is executed.

3.3 Workaround

When using the project involved in this problem, perform the rebuild project function.

3.4 Schedule of Fixing Problem

We plan to fix this problem in the next version of the common program to the CubeSuite+.

[Disclaimer]