RENESAS Tool News

RENESAS TOOL NEWS on May 1, 2004: RSO-M3T-CC32R_1-040501D

A Note on Using C-Compiler Package M3T-CC32R

--On the Power Functions in the Standard Library--

Please take note of the following problem in using the M3T-CC32R C-compiler package for the M32R family MCUs:

- On the power functions (powf5, powf, and pow) in the standard library

---

1. **Versions Concerned**
   M3T-CC32R V.4.20 Release 1 and 1A

2. **Description**
   Using the powf5, powf, and pow functions may generate incorrect results of operations.

   2.1 Conditions
   This problem occurs if the following two conditions are satisfied:
   (1) Any of the following functions are called:
   (a) a powf5 function
   (b) a powf function with the -m32re5 compile option selected (*1)
   (c) a pow function with the -float_only and -m32re5 compile options selected (*1)
   (2) When x and y represent the first and second input arguments of the function in (1), none of the following cases apply to them:
   (a) Either of x and y is 0.0.
   (b) The value of x is negative (including -0.0) and y is not an integer (*2).
   (c) Either of x and y is infinity (∞) or not a number (NaN) (*2 and *3).

NOTES:

*1. Selecting these options replaces the calling of the powf or pow function with that of the powf5.

*2. If these cases are applied, no power operation is performed; macro name "EDOM" is assigned to external variable "errno", which holds an error number.

*3. Infinity (∞) and not a number (NaN) are the special symbols returned if illegal operations are performed concerning floating-point operations (for example, division by 0.0). So don't use them in usual operations.

2.2 Example

Source file: sample.c

```
-----------------------------------------------------------------
#include <math.h>
#include <mathf.h>
float ans1, ans2, ans3, ans4;
void func(void)
{

/* Conditions (1)-(a) and (2) */
   ans1 = powf5(4.0, 0.5);

/* Conditions (1)-(b)(-m32re5 selected) and (2) */
   ans2 = powf(10000.0, 0.25);

/* Conditions (1)-(c)(-float_only and -m32re5 selected) and (2) */
   ans3 = pow(27.0, 1.0/3.0);

/* No condition is met */
   ans4 = powf5(0.0, 1.0);
}
-----------------------------------------------------------------
```

3. Workaround

This problem can be circumvented in either of the following ways:

(1) Immediately after the include statements of mathf.h and mathf.h, put two lines for replacing powf5 with powf.

Modification of the source file sample.c

```
-----------------------------------------------------------------
 #include <math.h>
 #include <mathf.h>

 #undef powf                /* Added */
 #define powf5 powf           /* Added */

 float ans1, ans2, ans3, ans4;
 void func(void)
 {
    ans1 = powf5(4.0, 0.5);
    ans2 = powf(10000.0, 0.25);  /* No problem arises
                              even if -m32re5 selected */
    ans3 = pow(27.0, 1.0/3.0);   /* No problem arises
                 even if -float_only and -m32re5 selected */
    ans4 = powf5(0.0, 1.0);
 }
-----------------------------------------------------------------
```

(2) Use the fixed_powf5 function.
Download the file named fixed_powf5.zip (3.01KB) of the fixed_powf5 function (the modification of powf5) and use it following the instructions of "How to Use the fixed_powf5 Function." described below.
The fixed_powf5.lzh file consists of the following files:

small/fixed_powf5.mo	...	the fixed_powf5 function for the small memory model
small/fixed_powf5.stk	...	the file for calculating stack size for the small memory model
medium/fixed_powf5.mo	...	the fixed_powf5 function for the medium memory model
medium/fixed_powf5.stk	...	the file for calculating stack size for the medium memory model
large/fixed_powf5.mo	...	the fixed_powf5 function for the large memory model
large/fixed_powf5.stk	...	the file for calculating stack size for the large memory model

How to Use the fixed_powf5 Function

1. Search for all the functions that satisfy Condition (1) above.
2. Replace every function involved with the fixed_powf5 one.

This explicitly indicates that the replacement has been made to circumvent the problem.

3. Be sure to add the declaration shown below immediately after the include statements of math.h and mathf.h.

   ```
   ---------------------------------------
   extern float fixed_powf5(float,float);
   ---------------------------------------
   ```

4. To create a load module, link to it the fixed_powf5.mo file (*) included in the downloaded fixed_powf5.lzh file.
   NOTE:
   * Select the file appropriate to the memory model you use out of the three.

4. Schedule of Fixing the Problem

We plan to fix this problem in our next release of the product.
