RENESAS TOOL NEWS on April 16, 2006: RSO-M3T-CC32R-060416D

A Note on Using the C/C++ Compiler Package --M3T-CC32R-- for the M32R Family of MCUs

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

- On comparing a pointer with an integer of 0 using the != operator

1. Versions Concerned

C/C++ compiler package M3T-CC32R V.4.00 Release 1--V.5.00 Release 00 (for the M32R family)

2. Description

If a pointer that is dynamically initialized to NULL or zero is compared with an integer of 0 using the inequality operator (!=), the result of the comparison always goes to TRUE.

2.1 Conditions

This problem occurs if the following conditions are all satisfied:

(1) Any of these optimizing options, -O7, -O6, -O5, and -O4, is selected; or -Ospace or -Otime alone is selected.

(2) A pointer that is dynamically initialized to NULL or zero exists in a function.

(3) The pointer in (2) is compared with an integer of 0 using the inequality operator within the same function, and the expression of the comparison is utilized as follows:

(a) as the controlling expression of an if statement or

(b) as the first operand of the conditional operator which is used by another operator or a statement

2.2 Examples

Example 1:

```
-----------------------------------------------------------
void func1(void)
{
   void (*p_func1)(void);
   p_func1 = 0;             /* Condition (2) */

   if (p_func1 != 0) {       /* Condition (3)-(a) */
      (*p_func1)();
   }
}
-----------------------------------------------------------
```

Example 2:

```
-----------------------------------------------------------
#include <stddef.h>
short *p_short;
void func2(void)
{
   p_short = NULL;           /* Condition (2) */

   if (0 != p_short) {       /* Condition (3)-(a) */
      *p_short = 100;
   }
}
-----------------------------------------------------------
```

Example 3:

```
-----------------------------------------------------------
#include <stddef.h>
int func3(void)
{
   int short *p_int = NULL;      /* Condition (2) */

   return (p_int!=0 ? 1 : 2);  /* Condition (3)-(b) */
}
-----------------------------------------------------------
```

3. **Workarounds**

This problem can be circumvented either of the following ways:

(1) Type cast the 0 of an operand of the inequality operator to void*, that is, add (void*) to the 0, or change the 0 to an NULL macro.

Circumvention of Example 1:

```
----------------------------------------------------------
---------
void func1(void)
{
    void (*p_func1)(void);
    p_func1 = 0;

    if (p_func1 != (void*)0) {     /* Change 0 to (void*)0 */
        (*p_func1)();
    }
}
----------------------------------------------------------
---------
```

Circumvention of Example 2:

```
----------------------------------------------------------
---------
#include <stddef.h>
short *p_short;
void func2(void)
{
    p_short = NULL;

    if (NULL != p_short) {       /* Change 0 to NULL */
        *p_short = 100;
    }
}
----------------------------------------------------------
---------
```

Circumvention of Example 3:

```
----------------------------------------------------------
---------
#include <stddef.h>
int func3(void)
{
    int short *p_int = NULL;
```

```
   return (p_int !=(void*)0 ? 1 : 2);  /* Change 0 to
(void*)0 */
}
------------------------------------------------------------
---------
```

(2) Suppress the optimization of level 4 either of the following ways:

  (a) If any of these options, -O7, -O6, -O5, and -O4, is selected, change it to any of these, -O3, -O2, -O1, and -O0.

  (b) If -Ospace or -Otime alone is selected, use any of these, -O3, -O2, -O1, or -O0, at the same time.

4. Schedule of Fixing the Problem

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

[Disclaimer]

The past news contents have been based on information at the time of publication. Now changed or invalid information may be included. The URLs in the Tool News also may be subject to change or become invalid without prior notice.