RENESAS Tool News

RENESAS TOOL NEWS on January 28, 2014: 140128/tn2

Note on Using C compilers for 78K0 of MCUs CA78K0 and CC78K0

When using the C compilers for the 78K0 of MCUs CA78K0 and CC78K0, take note of the following problems:

- With incorrect code being output for the processing of multiple casts of floating-point constants
- With the conversion of character strings by the strtol function producing incorrect numerical values

---

1. Problem with Incorrect Code Being Output for the Processing of Multiple Casts of Floating-Point Constants

1.1 Products and Versions Concerned

CA78K0 V1.20 to V1.30
(included in the integrated development environment CubeSuite+)
CA78K0 V1.00 to V1.11
(included in the integrated development environment CubeSuite)
CC78K0 V1.00 to V4.10
(bundled with the integrated development environment PM+)

1.2 Description

Multiple casting of floating-point constants produces incorrect results of operations.

1.3 Conditions

This problem arises if the following conditions are all met:
(1) A floating-point constant or constant cast to a floating-point type is cast to a floating-point type.
(2) The constant described in condition (1) above is cast to an integer type.
(3) The result of the operation described by condition (2) above is used in an operation other than the following:
- Simple assignment operation: =

- Logical operation: && or ||
- Conditional operation: ? :
- unary operation: !

1.4 Example

```
-------------------------------------------------------------
[*.C]
#define A ((long)((double)6031.0))  //Conditions (1) and (2)
void func(void)
{
   long x;
   x=A<<1;                    // Condition (3)
}
-------------------------------------------------------------
```

The result is not x = 12062 (= 6031 * 2).

1.5 Workaround

Do not cast a floating-point constant or constant cast to a floating-point type to a floating-point type.

```
-------------------------------------------------------------
#define A ((long)6031.0)
void func(void)
{
   long x;
   x=A<<1;
}
-------------------------------------------------------------
```

The result is x = 12062 (= 6031 * 2), which is correct.

2. Problem with the Conversion of Character Strings by the Strtol Function Producing Incorrect Numerical Values

2.1 Products and Versions Concerned

CA78K0 V1.20 to V1.30
(included in the integrated development environment CubeSuite+)
CA78K0 V1.00 to V1.11
(included in the integrated development environment CubeSuite)
CC78K0 V1.00 to V4.10
(bundled with the integrated development environment PM+)

2.2 Description

When character strings are converted into numerical values by the strtol function, in some cases, the value may overflow and be returned as conversion has not been performed correctly.

2.3 Conditions

This problem arises if the following conditions are all met:

(1) Character strings are converted into numerical values by the strtol function.

(2) In the processing of functions described in (1) above, conversion to numerical values is performed from the head of a character and a carry for each 0x10000 is produced during conversion.
A carry for each 0x10000 is produced when the conditions below is met while conversion is being performed.

((N * base) / 0x10000) != ((N * base + c) / 0x10000)

N: The value obtained by converting a character string from its head to the n-th character
base: The radix
c: The value of the n + 1-th character

Example:
When strtol ("65537", &err, 10) is executed, 6553 (the value obtained by converting up to the 4th character), 10 (the radix), and 7 (the 5th character) are obtained.

Left side = ((6553 * 10) / 0x10000) = 0
Right side = ((6553 * 10 + 7) / 0x10000) = 1

Left side != right side is true, and the value returned has overflowed.

2.4 Workaround

When the radix is 10 in the strtol function, replace it with the atol function.
If the method above does not work, there is no workaround.

3. Schedule for Fixing the Problem

Sorry we have no plan to fix these problems.