

## Outline

When using the C/C++ compiler package for SuperH RISC engine family V.1 to V.9, note the following point.

1. When a string is converted to a numeric value by using a standard library function (SHC-0098)

Note: The number which follows the description of a precautionary note is an identifying number for the precaution.

## 1. When a String is Converted to a Numeric Value by Using a Standard Library Function (SHC-0098)

### 1.1 Applicable Products

All versions of C/C++ compiler package for SuperH RISC engine family

(V.1.01 Release 00 to V.9.04 Release 03)

### 1.2 Details

If a string is converted to a numeric value by using a standard library function such as `strtol()`, a pointer that does not comply with the standard might be set in the second argument.

### 1.3 Conditions

This problem may arise if both of the following conditions (1) and (2) are met.

- (1) One of the following standard library functions is used.

- `strtol()`
- `strtoll()`
- `strtoul()`
- `strtoull()`
- `strtod()`

- (2) A string that meets both of the following conditions (a) and (b) is specified in the first argument for (1).

(a) At least one white-space character exists at the beginning of the string.

(b) The white-space characters in (a) are followed by a character that does not represent an integer<sup>(Note)</sup>.

Note: If the value of the third argument is from 10 to 36, some or all the alphabetic characters are handled as characters that represent an integer.

## 1.4 Example

Red text indicates the parts corresponding to the above conditions.

### ■ Example source

```
#include <stddef.h>
#include <stdio.h>
#include <stdlib.h>
long test;
const char* str = "□ Hello World!"; /* (Conditions (2-1) (2-2)) */
void main(void) {
    char* endptr = NULL;
    test=strtol(str, &endptr, 0); /* (Condition (1)) */
    printf(" str=%s\n endptr=%s\n", str, endptr);
}
```

### ■ Correct output result

According to the specifications, a pointer is set so that endptr is equal to str, and information is correctly output as follows.

```
str="  Hello World!"
endptr="  Hello World!"
```

### ■ Actual output result

A pointer skipping the white-space characters is set, and information is incorrectly output as follows.

```
str="  Hello World!"
endptr="Hello World!"
```

➤ Supplement: Standard of strtol()

The numeric part at the beginning of the string is converted to long type in the radix specified for base. If the string contains a character that cannot be converted, the pointer to that character is stored in endptr. If the space is followed by a numeric value, the data is skipped up to the end of the numeric value. However, if the space is followed by a non-numeric value, the data is stored in endptr without skipping the space.

For details about strtol(), refer to the URL below.

<https://www.renesas.com/search/keyword-search.html#genre=document&q=r20ut0704>

SuperH C/C++ Compiler Package V.9.04 User's Manual  
10.4.1 Standard C Libraries

■ How to Use the Function

```
long strtol(
    const char *nptr, /* String */
    char **endptr,
    /* Pointer to the storage area containing a pointer to the first character that does not represent an integer */
    int base /* Radix */
);
```

■ Example source used

```
#include <stddef.h>
#include <stdio.h>
#include <stdlib.h>
long test;
const char* str = " 123 Hello 456 World!";
void main(void) {
    char* endptr = NULL;
    test=strtol(str, &endptr, 0);
    printf(" str=%s\n endptr=%s\n test=%ld\n",str,endptr,test);
}
```

■ Output result

```
str=" 123 Hello 456 World!"
endptr=" Hello 456 World!" /* The address following the numeric value is stored in the pointer. */
test=123/* The value converted as the return value is returned in long type. */
```

## 1.5 Workaround

To avoid this problem, use the following method.

Make sure that a white-space character is not contained at the beginning of the string that is passed to the first argument in Condition (1).

■ Example:

```
const char* str = "   Hello World!"; /* Three spaces are contained. */
void main(void) {
    char* endptr = NULL;
    (void)strtol(&str[3], &endptr, 0);
    /* By prohibiting spaces, Condition (2) is avoided.*/
    ...
}
```

## 1.6 Schedule for Fixing the Problem

We do not plan to fix the problem.

**Revision History**

Rev.	Date	Description	
		Page	Summary
1.00	Aug. 1, 2017	-	First edition issued

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061 Japan  
 Renesas Electronics Corporation

■Inquiry

<https://www.renesas.com/contact/>

Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

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.

All trademarks and registered trademarks are the property of their respective owners.