

## A Note on Using the C/C++ Compiler Package for the M32R MCU Family

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

- With using a function-like macro that takes the same function-like macro as an argument

### 1. Product and Versions Concerned

The C/C++ compiler package for the M32R family  
V.5.00 Release 00 and V.5.01 Release 00

### 2. Description

If a function-like macro that is spread over two or more lines takes the same function-like macro as an argument, the latter may not be expanded.

#### 2.1 Conditions

This problem occurs if the following conditions are all satisfied:

- (1) A function-like macro statement is spread over two or more lines.
- (2) The function-like macro in (1) takes the same function-like macro as an argument.
- (3) The function-like macro as an argument in (2) is placed in front of a line or just follows comments or a sequence of white-space characters in front of a line.

#### 2.2 Example

Source file (sample1.c):

```
-----  
#define SMP_MACRO(a,b) (a + b)  
int var = SMP_MACRO(1,                <- Condition (1)  
                SMP_MACRO(2,3));      <- Conditions (2) and (3)  
-----
```

Result of macro replacement (output of cc32R -E sample1.c):

```
-----  
#line 1 "sample1.c"
```

```
int var = (1 + SMP_MACRO(2,3)) ; /* SMP_MACRO(2,3) not expanded */  
-----
```

### 3. Workarounds

Avoid the problem in either of the following ways:

- (1) Concatenate the line containing the function-like macro as an argument in (3) to the preceding line by placing a ¥ character at the end of the preceding line.

Modification of sample1.c:

```
-----  
#define SMP_MACRO(a,b) (a + b)  
int var = SMP_MACRO(1, ¥ <- ¥ placed at the end of this line  
                SMP_MACRO(2,3));  
-----
```

- (2) Define another function-like macro with the same format and a different name; then use it as an argument.

Modification of sample1.c:

```
-----  
#define SMP_MACRO(a,b) (a + b)  
#define SMP_SAME_MACRO(a,b) (a + b) <- Same format as SMP_MACRO  
int var = SMP_MACRO(1,  
                SMP_SAME_MACRO(2,3)); <- SMP_MACRO replaced  
-----
```

---

#### [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.