

## M3T-NC30WA ご使用上のお願い --標準ライブラリの文字列操作関数について--

M16C/60, M16C/30, M16C/Tiny, M16C/20, M16C/10およびR8C/Tinyシリーズ用 Cコンパイラパッケージ M3T-NC30WA の使用上の注意事項を連絡します。

- 標準ライブラリの文字列操作関数に関する注意事項

### 1. 該当製品

M3T-NC30WA V.5.00 Release 1 ~ V.5.30 Release 02

### 2. 内容

標準ヘッダ<string,h>で宣言されているライブラリ関数を呼び出すソースをコンパイルすると、Windows版ではアプリケーションエラー、EWS版およびLinux版ではコアダンプが発生し、コンパイラが強制終了されることがあります。

#### 2.1 発生条件

以下の条件をすべて満たす場合に発生することがあります。

なお、アプリケーションエラーまたはコアダンプが発生しなかった場合は、生成されたコードに問題はありません。

- (1) コンパイルオプション -OR を使用している。
- (2) コンパイルオプション -Ono\_stdlib(-ONS) を使用していない。
- (3) 1つの関数から2回以上同じライブラリ関数を呼び出している。
- (4) (3)で2回以上呼ばれるライブラリ関数は、以下のいずれかである。

memcmp, memcpy, memicmp, memmove, strcat,  
strcmp, strcoll, strcpy, strcspn, strcmp, strncat, strncmp,  
strncpy, strnicmp, strpbrk, strspn, strstr, およびstrxfrm  
(上記の関数はすべて、その引数に2つのポインタ型引数を含む。)

- (5) (3)のライブラリ関数を呼び出す箇所のいずれかで、2つのポインタ型引数両方にfarポインタ(アドレス定数を含む)を渡している。
- (6) (3)のライブラリ関数を呼び出す箇所のいずれかで、2つのポインタ型引数の一方にはfarポインタ(アドレス定数を含む)を渡し、他方にはnearポインタを渡している。
- (7) (5)および(6)の呼び出し箇所の両方とも、(3)の関数の戻り値を参照していない。
- (8) (5)および(6)の2つの呼び出し箇所の後、処理の流れが合流している。(ソース上、明示的にそうになっていなくても、最適化でそのようになる場合がある。)

## 2.2 発生例

```
-----  
#include <string.h>  
  
char near  nearstr[10];  
char far   far1str[10], far2str[10];  
int       cond;  
  
void func(void)  
{  
    if (cond == 0x0001) {  
        strcpy(nearstr, far2str); /* 発生条件(3)、(4)、(6)および(7) */  
    } else {  
        strcpy(far1str, far2str); /* 発生条件(3)、(4)、(5)および(7) */  
    }  
    cond = 0;                /* 発生条件(8) */  
}
```

```
-----
```

## 3. 回避策

コンパイルオプション `-Ono_stdlib(-ONS)` を追加してコンパイルしてください。

## 4. 恒久対策

本問題は、次期バージョンアップ時に改修する予定です。

---

**【免責事項】**

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。ニュース本文中のURLを予告なしに変更または中止することがありますので、あらかじめご承知ください。

© 2010-2016 Renesas Electronics Corporation. All rights reserved.