

## M16Cファミリ用Cコンパイラパッケージ ご使用上のお願い --構造体のメンバを、sizeof演算子を含む式で 初期化する場合の注意事項--

M16Cファミリ用Cコンパイラパッケージの使用上の注意事項を連絡します。

- 構造体のメンバを、sizeof演算子を含む式で初期化する場合の注意事項

### 1. 該当製品

- (1) R32C/100シリーズ用Cコンパイラパッケージ V.1.01 Release 00
- (2) M32Cシリーズ(注1)用Cコンパイラパッケージ (M3T-NC308WA)  
V.5.10 Release 1 ~ V.5.41 Release 01
- (3) M16Cシリーズ(注2)用Cコンパイラパッケージ (M3T-NC30WA)  
V.5.10 Release 1 ~ V.5.44 Release 00

**注1:** M32C/80、M16C/80、およびM16C/70シリーズの総称です。

**注2:** M16C/60、/30、/20、/10、/Tiny、およびR8C/Tinyシリーズの総称です

### 2. 内容

構造体のメンバを、sizeof演算子を含む式で初期化した場合に、コンパイラによるsystem error、またはwindowsエラーが発生する場合があります。

#### 2.1 発生条件

以下の条件をすべて満たす場合に発生します。

- (1) コンパイルオプション -Oconst(-OC)および-O3(-O)を使用している。  
もしくは、-O4、-O5、-OR\_MAXおよび-OS\_MAXのいずれかひとつを選択している。
- (2) const修飾された構造体を関数外で定義している。

- (3) (2)項の構造体のメンバをsizeof演算子を含む式で初期化している。  
(4) (2)項の構造体のメンバは、そのメンバを含む構造体の先頭メンバである。  
(5) (2)項の構造体のメンバの値を.(ドット)演算子で参照する式がある。

## 2.2 発生例

```
-----  
struct S{  
    int m;          /* 発生条件(4)項 */  
};  
  
const struct S t = {2+sizeof(int)}; /* 発生条件(2)および(3) 項*/  
  
int n;  
  
void func(void)  
{  
    n = t.m;       /* 発生条件(5) 項*/  
}  
-----
```

## 3. 回避策

構造体のメンバを定数で初期化してください。

```
-----  
struct S{  
    int m;  
};  
  
const struct S t = {4}; /* 定数で初期化 */  
  
int n;  
  
void func(void)  
{  
    n = t.m;  
}  
-----
```

## 4. 恒久対策

次バージョンで改修する予定です。

**[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.