

## M3T-NC30WA ご使用上のお願い --変数と定数とのOR演算について--

M16C/60, M16C/30, M16C/Tiny, M16C/20, M16C/10, R8C/Tinyシリーズ用Cコンパイラ(アセンブラ・統合開発環境付き) M3T-NC30WAの使用上の注意事項を連絡します。

- 変数と定数とのOR演算を行った場合の注意事項

### 1. 該当製品

M3T-NC30WA V.2.00 Release 1 ~ V.5.20 Release 1

### 2. 内容

unsigned long型変数またはsigned long型変数と2のべき乗の定数とのOR演算を行うと、不正なコードを生成する場合があります。

### 3. 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) 変数Aと2のべき乗の定数とのOR演算を行っている。
- (2) 変数Aの型は、unsigned long型またはsigned long型である。
- (3) 変数Aの宣言には、volatile修飾子がある。
- (4) (1)の演算結果を代入する変数Bは、以下の条件を満たしている。
  - a. 自動変数である。
  - b. 変数Bの宣言には、volatile修飾子がない。
  - c. 変数Bの配置アドレスは、以下のいずれかである。
    - FBレジスタ-17番地以下
    - FBレジスタ+16番地以上

## 4. 発生例

```
-----volatile long gl;          /* 発生条  
件(2)、(3) */void func(void){      long bl[6];          /* 発生条件(4) */  bl[0] = gl |  
0x000001L; /* 発生条件(1)、(4) */  ...}-----  
-----
```

## 5. 回避策

以下の手順で回避してください。

- (1) signed long型またはunsigned long型の変数Cをもうける。
- (2) 変数AとOR演算を行う2のべき乗の定数を(1)の変数に代入する。
- (3) 変数Aと変数CのOR演算を行う。

```
-----volatile long gl;void  
func(void){      long bl[6];      long tmp;          /* 変数C */  tmp = 0x000001L;  
bl[0] = gl | tmp;      ...}-----
```

## 6. 恒久対策

次期バージョンアップの際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。