

R32C/100シリーズ用Cコンパイラパッケージおよび M32Cシリーズ用Cコンパイラパッケージ ご使用上のお願い

M16CファミリR32C/100シリーズ用CコンパイラパッケージおよびM32Cシリーズ*用 Cコンパイラパッケージ(M3T-NC308WA)の使用上の注意事項を連絡します。

* M32Cシリーズは、M32C/80, M16C/80およびM16C/70シリーズの総称です。

- Cコンパイラの拡張機能 #pragma DMACを使用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シリーズ用Cコンパイラパッケージ (M3T-NC308WA)
V.5.00 Release 1 ~ V.5.41 Release 01

2. 内容

拡張機能 #pragma DMACで指示した変数を条件式で使用した場合、アセンブルエラーが発生することがあります。

3. 発生条件

以下のすべての条件を満たす場合に発生します。

- (1) 最適化オプション -O1, -O2, -O3, -O4, -O5, -OR, -OS, -OR_MAX および -OS_MAXのいずれか1つ以上を選択している。
- (2) #pragma DMACで指示した外部変数がある。
- (3) if文、while文またはfor文の条件判定式で、(2)項の外部変数と整数定数の論理積をとっている。
- (4) (3)の整数定数は、2のn乗で表される値(1, 2, 4, 8 . . .)である。

R32C/100シリーズ用Cコンパイラパッケージを使用する場合の発生例

unsigned char val;

```
unsigned long dmd0;
#pragma DMAC dmd0 DMD0      /* 発生条件(2) */

void func(void)
{
    if(dmd0 & 0x00000001L) { /* 発生条件(3)および(4) */
        val++;
    }
}
```

4. 回避策

#pragma DMACで指示した変数を一時変数に代入し、その一時変数を使用して判定してください。

回避例

```
unsigned char val;
unsigned long dmd0;
#pragma DMAC dmd0 DMD0

void func(void)
{
    unsigned long tmp;

    tmp = dmd0;          /* 一時変数に代入 */
    if(tmp & 0x00000001L) {
        val = 0x1;
    }
}
```

5. 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。