

M3T-NC308WA, M3T-NC30WA ご使用上のお願い

Cコンパイラ(アセンブラ・統合化開発環境付き) M3T-NC308WA、M3T-NC30WA の使用上の注意事項を連絡します。

- #pragma ADDRESS宣言を使用した場合の注意事項
- 代入元の変数に#pragma ADDRESS宣言またはvolatile修飾子を使用した場合の注意事項

1. #pragma ADDRESS宣言を使用した場合の注意事項

1.1 該当製品

M32C/80, M16C/80, M16C/70シリーズ用

M3T-NC308WA V.1.00 Release 1 ~ V.5.00 Release 1

M16C/60, M16C/30, M16C/20, M16C/10シリーズ用

M3T-NC30WA V.1.00 Release 1 ~ V.5.00 Release 2

1.2 内容

#pragma ADDRESS宣言した変数のvolatileが有効にならない場合があります。

1.3 発生条件

以下の条件の条件をすべて満たす場合に発生します。

- (1) 変数を定義している。
- (2) (1)の定義のあとで、(1)の変数を#pragma ADDRESS宣言している。

1.4 発生例

```
-----  
int    gi;        /* 発生条件(1) */  
#pragma ADDRESS gi 500H /* 発生条件(2) */  
-----
```

1.5 回避策

#pragma ADDRESS宣言を変数定義よりも先に宣言してください。

```
-----  
#pragma ADDRESS gi 500H  
int gi;  
-----
```

1.6 恒久対策

本内容は、次期バージョンアップで改修する予定です。

ページの先頭へ

[M3T-NC308WA, M3T-NC30WA ご使用上のお願い](#)
[MAECT-M3T-NC308WA_2-030316D](#)

2. 代入元の変数に#pragma ADDRESS宣言またはvolatile修飾子を使用した場合の注意事項

2.1 該当製品

M32C/80, M16C/80, M16C/70シリーズ用

M3T-NC308WA V.1.00 Release 1 ~ V.5.00 Release 1

2.2 内容

代入文の代入先の変数のサイズが8ビットで、代入元の変数(※)のサイズが16ビットの場合、代入元の変数が8ビットのサイズで参照される場合があります。16ビットで読み出す必要があるI/Oレジスタ等の場合は、8ビットで読み出されてしまうのでご注意ください。

(※) volatile修飾された変数または#pragma ADDRESS宣言された変数に限ります。

2.3 発生条件

以下の条件の条件をすべて満たす場合に発生します。

- (1) 代入文がある。
- (2) (1)の代入文の代入先の変数のサイズは8ビットである。
- (3) (1)の代入文の代入元の変数のサイズは16ビットである。
- (4) (3)の代入元の変数は、volatile修飾または#pragma ADDRESS宣言された変数である。

2.4 発生例

```
-----  
#pragma ADDRESS XOR 2C0H  
char gc; /* 発生条件(2) */  
int XOR; /* 発生条件(3) */  
-----
```

```
void func(void)
{
    gc = XOR;          /* 発生条件(1) */
}
```

2.5 回避策

代入元の変数をサイズが16ビットのvolatile修飾していないテンポラリ変数に代入し、テンポラリ変数を代入先の変数に代入してください。

```
volatile int gi;
char    gc;
void func(void)
{
    int tmp;
    tmp = gi;    /* 代入元の変数をテンポラリ変数に代入 */
    gc = tmp;    /* テンポラリ変数を代入先に代入 */
}
```

2.6 恒久対策

本内容は、次期バージョンアップ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。