

M3T-NC308WAご使用上のお願い --const修飾された配列型の参照について--

M32C/90, M32C/80, M16C/80および M16C/70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。

- const修飾された配列型の参照に関する注意事項

1. 該当製品

M3T-NC308WA V.5.00 Release 1 ~ V.5.20 Release 02

2. 内容

typedefで宣言された配列型変数をconst修飾して参照すると、リンク時にワーニングを出力し、生成された実行形式ファイルは正しく動作しません。

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) typedefを用いて配列型変数を型定義している。
- (2) (1)の配列型変数は、const、nearおよびfar修飾のいずれの修飾もされていない。
- (3) (1)の配列型変数は、関数外で宣言されている。またはstatic記憶クラスを用いて関数内で宣言されている。
- (4) (3)の配列型変数はconst修飾されているが、nearおよびfar修飾はされていない。
- (5) (3)の配列型変数の要素を、配列名を用いて直接参照している。
- (6) コンパイルオプション -fconst_not_ROM(-fCNR), -ffar_RAM(-fFRAM)および -fnear_ROM(-fNRROM) のいずれも使用していない。

2.2 発生例

```

-----
typedef char  arr_t[2];          /* 発生条件(1)および(2) */
const arr_t  arr = { 0x11, 0x22 }; /* 発生条件(3)および(4) */
char c;

void func(void)
{
    c = arr[1];                 /* 発生条件(5) */
}
-----

```

発生例のリンク時の出力メッセージ

```

-----
a.c 7 Warning (ln308): a.r30 : 16-bits unsigned value is out of
    range 0 -- 65535. address='fe0124'
-----

```

3. 回避策

以下のいずれかの方法で回避してください。

- (1) const修飾する配列型変数を、typedef宣言しない。

例

```

-----
const char  arr = { 0x11, 0x22 };
-----

```

- (2) const修飾する配列型変数を、明示的にfar修飾して宣言する。

例

```

-----
const arr_t far arr = { 0x11, 0x22 };
-----

```

- (3) 配列型変数のtypedef宣言と、const修飾を同時に行う。

例

```

-----
typedef const char  c_arr_t[2];
c_arr_t            arr = { 0x11, 0x22 };
-----

```

4. 恒久対策

次期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.