

M3T-CC32R V.4.30 Release 00 ご使用上のお願い

M32Rファミリ用CコンパイラパッケージM3T-CC32R V.4.30 Release 00の使用上の注意事項を連絡します。

- #pragma ADDRESSで構造体、配列または共用体を含むオブジェクトのアドレスを指定する場合の注意事項

1. 該当製品

M3T-CC32R V.4.30 Release 00

2. 内容

集成体(配列および構造体)または、集成体をメンバに持つ共用体に#pragma ADDRESS を指定した場合、オブジェクトの先頭アドレス以外に配置されたメンバまたは要素に、アクセス(読み出しおよび書き込み)または&でアドレスを取り出す処理を行うと、誤って集成体の先頭のメンバや要素のアドレスを用いたコードを生成します。

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) 以下のいずれかのオブジェクトを、関数の外で宣言している。
 - (a) 構造体
 - (b) 配列
 - (c) 構造体または配列をメンバに持つ共用体
- (2) (1)のオブジェクトに#pragma ADDRESS を指定している。
- (3) (1)のオブジェクトの、先頭アドレス以外に配置されたメンバまたは要素に対し、以下のいずれかを行っている
 - (a) アクセス(読み出しまたは書き込み)。
 - (b) &演算子による配置アドレスの取得

2.2 発生例

ソースファイル例1

sample1.c

```
#pragma ADDRESS data1 0x10000    /* 条件(2) */
struct stg1 {
    int a;
    int b;
} data1;                          /* 条件(1)(a) */

void func1(void)
{
    data1.b = 3;                  /* 条件(3)(a) */
}
```

ソースファイル例2

sample2.c

```
#pragma ADDRESS data2 0x20000    /* 条件(2) */
short data2[10];                /* 条件(1)(b) */
short *d2;

void func2(void)
{
    d2 = &data2[3];              /* 条件(3)(b) */
}
```

ソースファイル例3

sample3.c

```
union utg3 {
    double a[10];
    struct {
        int b;
        double c[5];
    } d;
} data3;                          /* 条件(1)(c) */

#pragma ADDRESS data3 0x30000    /* 条件(2) */

void func3(int i)
```

```
{
    data3.d.c[i] =          /* 条件(3)(a) */
    data3.a[4];           /* 条件(3)(a) */
}
```

3. 回避策

ポインタにキャストされたアドレスを用いてオブジェクトにアクセスするように変更ください。以下の例のようにマクロを用いると便利です。

sample1.c の変更例

```
#define data1 ( *(struct stg1*) 0x10000 ) /* マクロに変更 */
```

```
struct stg1 {
    int a;
    int b;
} /* data1 */ ;    /* data1の定義は削除し、
                  構造体タグstg1の宣言のみにする */
```

```
void func1(void)
{
    data1.b = 3;    /* マクロにより置換される */
}
```

sample2.c の変更例

```
#define data2 ( (short *) 0x20000) /* マクロに変更 */
```

```
/* short data2[10]; */    /* data2の定義は削除する */
short *d2;
```

```
void func2(void)
{
    d2 = &data2[3];    /* マクロにより置換される */
}
```

sample3.c の変更例

```
union utg3 {
    double a[10];
    struct {
```

```
int b;
double c[5];
} d;
} /* data3 */ ;      /* data3の定義は削除し、
                     共用体タグutg3の宣言のみにする */

#define data3 ( *(union utg3*) 0x30000 ) /* マクロに変更 */

void func3(int i)
{
    data3.d.c[i] =          /* マクロにより置換される */
    data3.a[4];           /* マクロにより置換される */
}
-----
```

4. 恒久対策

本問題は、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。