

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

=====**必ずお読みください**=====

M32C/90, /80, M16C/80, /70 シリーズ用 C コンパイラパッケージ  
V.5.41 Release 00

リリースノート  
(第 2 版)

株式会社ルネサス ソリューションズ  
2006 年 9 月 16 日

概要

この度は M32C/90, M32C/80, M16C/80, M16C/70 シリーズ用 C コンパイラパッケージ V.5.41 Release 00 をご採用いただきありがとうございます。

本資料は C コンパイラパッケージの電子マニュアルの補足等について説明します。電子マニュアルの該当項目をご覧になる場合は、併せてこのリリースノートをご覧いただきますようお願い申し上げます。

1. 注意事項.....	3
1.1. Cコンパイラに関する注意事項.....	3
1.1.1. 整数定数同士の演算に関する注意事項.....	3
1.1.2. スtring命令、積和演算命令に関する注意事項.....	3
1.1.3. コンパイルオプション "-fdouble_32 (-fD32)" の注意事項.....	3
1.1.4. マクロ定義に関する注意事項.....	4
1.1.5. #if指令に関する注意事項.....	4
1.1.6. inline関数のネストに関する注意事項.....	5
1.1.7. コマンドオプション "-I" に関する注意事項.....	5
1.1.8. インクルードファイルの検索に関する注意事項.....	5
1.1.9. インラインアセンブル機能 (#pragma ASM ~ #pragma ENDASM、asm関数) に関する注意事項.....	5
1.1.10. _Bool型を使用しているプログラムのデバッグに関する注意事項.....	6
1.1.11. 前処理命令#defineに関する注意事項.....	6
1.2. 機種依存に関する注意事項.....	6
1.2.1. M16Cの割り込み制御レジスタに関する注意事項.....	6
1.2.2. SFR領域のアクセスに関する注意事項.....	7
1.3. Microsoft Windowsに関する注意事項.....	7
1.3.1. 動作環境に関する注意事項.....	7
1.3.2. ファイル名に関する注意事項.....	7
1.3.3. ウィルスチェックプログラムに関する注意事項.....	7
2. バージョンアップ内容.....	8
2.1. Cコンパイラ.....	8
2.1.1. 機能追加.....	8
2.1.2. 機能改定.....	8
2.2. アセンブラシステム.....	8
2.2.1. 機能追加.....	8
2.2.2. 機能改定.....	9
2.3. 統合開発環境.....	9

2.3.1. 更新 .....	9
2.4. コーティリティツール .....	9
2.4.1. 機能追加 .....	9
2.5. その他 .....	9
2.5.1. 機能追加 .....	9
3. ソフトウェアのバージョン一覧 .....	10
4. 標準関数ライブラリのMISRA Cルール適合に関して .....	11
4.1. ルール違反の要因 .....	11
4.2. ルール違反となった検査番号 .....	11
4.3. 評価環境 .....	11

## 1. 注意事項

製品をご使用になる際は、以下の内容に従ってくださるようお願いいたします。

### 1.1. C コンパイラに関する注意事項

#### 1.1.1. 整数定数同士の演算に関する注意事項

整数定数同士の演算結果が int 型のビット幅を超える値の場合、V.5.40 Release 00 以前とは異なるコードを生成しません。

- 発生例

```
void    func(void)
{
    unsigned long    l;

    l = 256 * 256;    // l=0            (V.5.41 Release 00)
                    // l=65536       (V.5.40 Release 00)
}
```

- 回避策

整数定数同士の演算結果が int 型のビット幅を超える場合は、整数定数に接尾語を付加してください。

```
void    func(void)
{
    unsigned long    l;

    l = 256UL * 256UL;
}
```

#### 1.1.2. スtring命令、積和演算命令に関する注意事項

V.5.41 Release 00 より、コンパイルオプション "-fuse\_strings (-fUS)" の選択がない場合は M32C/90, /80, M16C/80, /70 シリーズの String 命令を、コンパイルオプション "-fuse\_product\_sum (-fUPS)" の選択がない場合は M32C/90, /80, M16C/80, /70 シリーズの積和演算命令を用いたコードを生成しません。

詳細は、次の RENESAS TECHNICAL UPDATE を参照ください。

- M16C/70 シリーズ、M16C/80 シリーズ、M32C/80 シリーズ、M32C/90 シリーズ String 命令、積和演算命令使用上の注意事項 (発行番号 : TN-16C-A157A/J)

M32C/90, /80, M16C/80, /70 シリーズの String 命令を用いたコードを生成する場合は、コンパイルオプション "-fuse\_strings (-fUS)"、積和演算命令を用いたコードを生成する場合は、コンパイルオプション "-fuse\_product\_sum (-fUPS)" を選択してください。

#### 1.1.3. コンパイルオプション "-fdouble\_32 (-fD32)" の注意事項

コンパイルオプション "-fdouble\_32 (-fD32)" を選択しコンパイルした場合、エラーメッセージ「nc308: Invalid option 'xxx'」を 2 件出力します。

このエラーメッセージは無視してください。

```
[ -fD32 を選択した場合のエラーメッセージ ]
nc308: Invalid option '-fD32'
nc308: Invalid option '-fD32'

[-fdouble_32 を選択した場合のエラーメッセージ]
nc308: Invalid option '-fdouble_32'
nc308: Invalid option '-fdouble_32'
```

### 1.1.4. マクロ定義に関する注意事項

マクロの定義内容にそのマクロ自体の名前を使用している場合、他の関数形式マクロの引数にそのマクロを指定すると、正しくマクロ置換されません。

- 発生例

```
int    a = 10;
#define a      a + a           // マクロ名 a
#define p( x,y ) x + y

void    func(void)
{
    int    i = p ( a , a );    // i = 80 になる。
                                // 正しくは i = 40
}
```

- 回避策

関数形式マクロの引数に渡すマクロは、その定義内容で使用しない名前でも定義してください。

```
int    a = 10;
#define b      a + a           // a とは異なるマクロ名に変更する
#define p( x,y ) x + y

void    func(void)
{
    int    i = p ( b , b );
}
```

### 1.1.5. #if 指令に関する注意事項

#if 指令の定数式がシフトで、そのシフトの左オペランドが負の値で、かつ右オペランドが unsigned 型の値である場合、シフト結果に対して正しく判定することができません。

- 発生例

```
void    func( void )
{
    char    a;

    #if (-1 << 1U) > 0        // 真と判断
        a = 1;                // (-1 << 1U) は -2 のため偽が正しい
    #else
        a = 2;
    #endif
}
```

- 回避策

シフトの左オペランドが負の値の場合は、そのシフトの右オペランドを signed 型の値にしてください。

```

void    func( void )
{
    char    a;

#ifdef  U_SUFFIXES // U 接尾語を使用しないことでシフトの右オペランドを signed 型にする
    a = 1;
#else
    a = 2;
#endif
}

```

### 1.1.6. inline 関数のネストに関する注意事項

仮引数を持つ inline 関数をネストすると、誤った実引数(実引数と異なる変数)を参照する場合があります。

- 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] inline 関数をネストしている。

[2] 呼び出し元 inline 関数 A と呼び出し先 inline 関数 B の仮引数名が同一である。

- 発生例

```

inline B(int aaa, char ccc)          /* 発生条件 (2) */
{
    .....
}

inline A(int c, int aaa, char *ccc)  /* 発生条件(2) */
{
    int    i;
    char   c;
    B(i,c);          /* 発生条件(1) */
}

```

- 回避策

次のいずれかの方法で回避してください。

[1] 呼び出し先の関数(発生例では、inline 関数 B)の引数名を変更する。

[2] inline 関数のネストをしない。

[3] コンパイルオプション "-Ofoward\_function\_to\_inline (-OFFTI)" を使用してコンパイルする。

### 1.1.7. コマンドオプション “-I” に関する注意事項

コマンドオプション “-I” で指定できるディレクトリの個数は最大 50 個です。

### 1.1.8. インクルードファイルの検索に関する注意事項

#include の記述において、ドライブ名付きで記述しコンパイル対象となるファイルが存在するディレクトリとは異なったディレクトリからコンパイルした場合、インクルードファイルを検索できない場合があります。

### 1.1.9. インラインアセンブル機能 (#pragma ASM ~ #pragma ENDASM、asm 関数) に関する注意事項

(1) #pragam ASM/ENDASM 内の記述に対して、アセンブル及びリンク時のエラーメッセージの行数、デバッグ情報の行情報等が正常に出力されない場合があります。

(2) コンパイラは、レジスタや変数の有効範囲について、プログラムフローを解析して処理を行っているため、インラインアセンブル機能 (#pragma ASM ~ #pragma ENDASM または asm 関数) でフローに影響を与えるようなブランチ (条件ブランチ含む) を記述しないようにしてください。

- (3) インラインアセンブル機能を使用してレジスタの値を変更する記述をする場合、有効範囲中でレジスタの値を変更した情報を得ることができません。必ずレジスタを退避・復帰してください。

#### 1.1.10. `_Bool` 型を使用しているプログラムのデバッグに関する注意事項

`_Bool` 型を使用したプログラムをデバッグする場合、デバッガが `_Bool` 型に対応しているかご確認ください。`_Bool` 型に対応していないデバッガをご使用になる場合は、コンパイル時にデバッグオプション `-gBTC(-gbool_to_char)` をご使用ください。

#### 1.1.11. 前処理命令 `#define` に関する注意事項

マクロ `ULONG_MAX` と同一値になるマクロを定義する場合は、必ず接尾語 `UL` を付けてください。

### 1.2. 機種依存に関する注意事項

#### 1.2.1. M16C の割り込み制御レジスタに関する注意事項

最適化オプション `-O5` を指定すると、ビット操作命令 (`BTSTC`、`BTSTS`) を生成する可能性があります。`BTSTC`、`BTSTS` 命令は、M16C の割り込み制御レジスタを書きかえる命令として使用できません。本オプションを使用する場合は、必ず生成されたコードに問題がない事をご確認ください。

- 発生例

以下のプログラムに対して最適化オプション `-O5` を指定した場合、最適化により `BTSTC` 命令を生成します。このため、割り込み要求ビットの判定が正しく行われず、意図しない動作をおこします。

```
#pragma ADDRESS TA0IC 006ch // M16C/80 タイマ A0 割り込み制御レジスタ
struct {
    char    ILVL : 3;
    char    IR   : 1; // 割り込み要求ビット
    char    dmy  : 4;
} TA0IC;

void WaitUntillRisON( void )
{
    while( TA0IC.IR == 0 ) // 1 になるまで待つ
    {
        ;
    } // 1 になったら 0 に戻す
}
```

- 回避策

- [1] 該当する最適化オプションに加えてオプション `-Ono_asmopt (-ONA)` を指定することにより、`BTSTC`、`BTSTS` 命令を生成する最適化を抑制してください。
- [2] 次のように `asm` 関数を挿入することにより、最適化を抑制してください。



```

#pragma ADDRESS TA0IC 006ch // M16C/80 タイマ A0 割り込み制御レジスタ
struct {
    char    ILVL : 3;
    char    IR   : 1; // 割り込み要求ビット
    char    dmy  : 4;
} TA0IC;

void WaitUntillRisON( void )
{
    while( TA0IC.IR == 0 ) // 1 になるまで待つ
    {
        asm( ); // 1 になったら 0 に戻す
    }
}

```

- 注意

コンパイルオプション "-Ono\_asmopt (-ONA) "、または asm 関数の使用による対策後は、必ず BTSTC、BTSTS 命令が生成されていない事を確認してください。

## 1.2.2. SFR 領域のアクセスに関する注意事項

SFR 領域のレジスタをアクセスする場合には特定の命令を使用しなければならないことがあります。この特定の命令は機種毎に異なりますので、詳しくは各機種のユーザーズマニュアルなどを参照してください。この注意事項に関わる命令は、asm 関数等のインラインアセンブル機能を使用して、プログラム中に命令を直接記述してください。

## 1.3. Microsoft<sup>1</sup> Windowsに関する注意事項

### 1.3.1. 動作環境に関する注意事項

- (1) C コンパイラパッケージは、Windows 98、Windows NT 4.0 以降の環境で動作します。Windows 3.1 および Windows NT 3.5x 以前のバージョンでは動作しません。
- (2) 日本語 Windows NT 環境でコマンドプロンプトのサイズが「80 x 25」以外に設定されている場合、製品を起動するとコマンドプロンプトのサイズが頻繁に切り替わります。コマンドプロンプトのサイズは「80 x 25」に設定してください。

### 1.3.2. ファイル名に関する注意事項

ソースプログラムファイルの名前や作業を行うディレクトリ名は、次の注意事項に従ってください。

- (1) 漢字を含むディレクトリ名、ファイル名は使用できません。
- (2) ファイル名に使用するピリオド(.)は一つのみ使用可能です。
- (3) ネットワークパス名は使用できません。ドライブ名に割り当ててご使用ください。
- (4) 「ショートカット」は使用できません。
- (5) "..."表記を用いて2つ以上のディレクトリを指定することはできません。
- (6) パス指定を含めたファイル名の長さが128文字以上になるものは使用できません。

### 1.3.3. ウィルスチェックプログラムに関する注意事項

ウィルスチェックプログラムが常駐した状態でC コンパイラパッケージを起動すると正常に起動しない場合があります。その場合は、ウィルスチェックプログラムの常駐を解除してからC コンパイラパッケージを起動しなおしてください。

<sup>1</sup> Microsoft、WindowsおよびWindows NTは、米国Microsoft Corporation の米国およびその他の国における商標または登録商標です。

## 2. バージョンアップ内容

### 2.1. C コンパイラ

#### 2.1.1. 機能追加

##### 2.1.1.1. コンパイルオプション “-fuse\_strings (-fUS)” の追加

ストリングス命令を使用したコードを生成するためのコンパイルオプション “-fuse\_strings (-fUS)” を追加しました。このオプションを選択する場合は、次の RENESAS TECHNICAL UPDATE を参照ください。

- M16C/70 シリーズ、M16C/80 シリーズ、M32C/80 シリーズ、M32C/90 シリーズ ストリング命令、積和演算命令使用上の注意事項 (発行番号 : TN-16C-A157A/J)

##### 2.1.1.2. コンパイルオプション “-fuse\_product\_sum (-fUPS)” の追加

積和演算命令を使用したコードを生成するためのコンパイルオプション “-fuse\_product\_sum (-fUPS)” を追加しました。

このオプションを選択する場合は、次の RENESAS TECHNICAL UPDATE を参照ください。

- M16C/70 シリーズ、M16C/80 シリーズ、M32C/80 シリーズ、M32C/90 シリーズ ストリング命令、積和演算命令使用上の注意事項 (発行番号 : TN-16C-A157A/J)

#### 2.1.2. 機能改定

##### 2.1.2.1. アセンブラマクロ関数、strcmp 関数の改定

アセンブラマクロ関数、および strcmp 関数に対して、次の RENESAS TECHNICAL UPDATE に対する対策コードを適用しました。

- M16C/70 シリーズ、M16C/80 シリーズ、M32C/80 シリーズ、M32C/90 シリーズ ストリング命令、積和演算命令使用上の注意事項 (発行番号 : TN-16C-A157A/J)

##### 2.1.2.2. コンパイル速度の改善

最適化オプション “-O3” ~ “-O5”、“-OR”、“-OS”、“-OR\_MAX”、“-OS\_MAX” 選択時のコンパイル時間を高速化しました。

##### 2.1.2.3. 問題点改修

ツールニュースでお知らせの次の問題点を改修しました。

- (1) MISRA C ルールチェッカ SQMlint を使用する場合の注意事項
- (2) for 文のループ内で配列型変数をアクセスする場合の注意事項
- (3) 拡張機能 #pragma SPECIAL の使用に関する注意事項

## 2.2. アセンブラシステム

### 2.2.1. 機能追加

#### 2.2.1.1. リンクオプション “-NOMCU” の追加

プロジェクトのターゲットマイコンとは異なるマイコンシリーズのリロケータブルオブジェクトファイルのリンクを可能にするためのリンクオプション “-NOMCU” を追加しました。

なお、このリンクオプションはリンク規則を変更します。このため、リンクはマイコンのシリーズによって使用できない命令が存在してもエラーを検出しません。このリンクオプションを選択する場合は注意してください。

### 2.2.1.2. アセンブラ指示命令 ".ID" による ID ファイル生成機能の追加

既存のロードモジュールコンバータのオプション ".ID" でも、アセンブラ指示命令 ".ID" でも ID ファイルを出力できます。

これにともない、アセンブラ指示命令 ".ID"、".PROTECT"、ロードモジュールコンバータのオプション ".ID"、"-protect1"および"-protectx"を組み合わせて使用する場合の仕様を変更しました。

## 2.2.2. 機能改定

### 2.2.2.1. エラー処理追加

アセンブラオプション "-M82" または "-M90" を設定した場合、"MUL.W" 命令および "MULU.W" 命令の dest にこれらの命令では使用できないアドレッシングモードを記述した場合にアセンブルエラーになるように改定しました。詳細は、次の RENESAS TECHNICAL UPDATE を参照ください。

- 「M32C/80 シリーズ、M32C/90 シリーズ MUL.W 命令、MULU.W 命令使用上の注意事項（発行番号：TN-16C-A156A/J）」

## 2.3. 統合開発環境

### 2.3.1. 更新

統合開発環境 High-performance Embedded Workshop を V. 4.00.03 から V.4.01.01 にバージョンアップしました。

## 2.4. ユーティリティツール

### 2.4.1. 機能追加

#### 2.4.1.1. Call Walker 用.sni ファイル作成ツール gensni

アブソリュートモジュールファイル(.x30)から Call Walker の入力ファイル(.sni)を作成するためのツール gensni を追加しました。

#### 2.4.1.2. Map Section Information ウィンドウ用.map ファイル作成ツール genmap

アブソリュートモジュールファイル(.x30)から High-performance Embedded Workshop の Map Section Information ウィンドウの入力ファイル(.map)を作成するためのツール genmap を追加しました。

## 2.5. その他

### 2.5.1. 機能追加

#### 2.5.1.1. Call Walker 対応

アプリケーションプログラム中で使用するスタックサイズを算出するためのユーティリティツール Call Walker をサポートしました。

#### 2.5.1.2. High-performance Embedded Workshop の Map Section Information ウィンドウ対応

アブソリュートモジュールファイルのマップ情報を表示する High-performance Embedded Workshop の Map Section Information ウィンドウをサポートしました。

### 3. ソフトウェアのバージョン一覧

C コンパイラパッケージ V.5.41 Release 00 に含まれているソフトウェアの各バージョンは以下のとおりです。

- nc308 V.1.08.05.000
- cpp308 V.4.08.01.000
- ccom308 V.5.04.14.000
- aopt308 V. 1.04.05.000
- as308 V. 4.03.00.000
- mac308 V. 2.22.00.000
- pre30 V.1.10.12
- asp308 V. 4.03.00.000
- ln308 V. 4.04.00.000
- lb308 V. 2.02.01.000
- lmc308 V. 3.00.02.000
- xrf308 VV. 2.02.00.000
- abs308 V. 1.03.00.000
- stk V.1.00.05.000
- utl308 V.1.00.11.000
- Stk Viewer V.1.00.01
- MapViewer V.3.01.02
- genmap V.1.00.00.000
- gensni V.1.00.00.002

## 4. 標準関数ライブラリの MISRA C ルール適合に関して

M32C/90, M32C/80, M16C/80, M16C/70 シリーズ用 C コンパイラパッケージの標準関数ライブラリの C ソースコードは、MISRA C ルールに対して 52 のルール違反が認められますが、これらの違反は動作に支障がありません。

### 4.1. ルール違反の要因

C コンパイラパッケージの標準関数ライブラリの C ソースコードにおいて、ルール違反となった主な要因は次の通りです。

- (1) C コンパイラの仕様 (near/far 修飾、asm()関数、#pragma)
- (2) ANSI 規格に基づく関数の宣言
- (3) 条件文における評価順序をカッコ()により明示的に記述していない
- (4) 暗黙の型変換

### 4.2. ルール違反となった検査番号

ルール違反になった検査番号は次のとおりです。

1	12	13	14	18	21	22	28	34	35
36	37	38	39	43	44	45	46	48	49
50	54	55	56	57	58	59	60	61	62
65	69	70	71	72	76	77	82	83	85
99	101	103	104	105	110	111	115	118	119
121	124								

### 4.3. 評価環境

コンパイラ	M32C/80, M16C/80, M16C/70 シリーズ用 C コンパイラパッケージ V.5.20 Release 1
コンパイルオプション	-O -c -gnone -finfo -fNII -misra_all -r \$*.csv
MISRA C チェッカ	SQMLint V.1.00 Release 1A