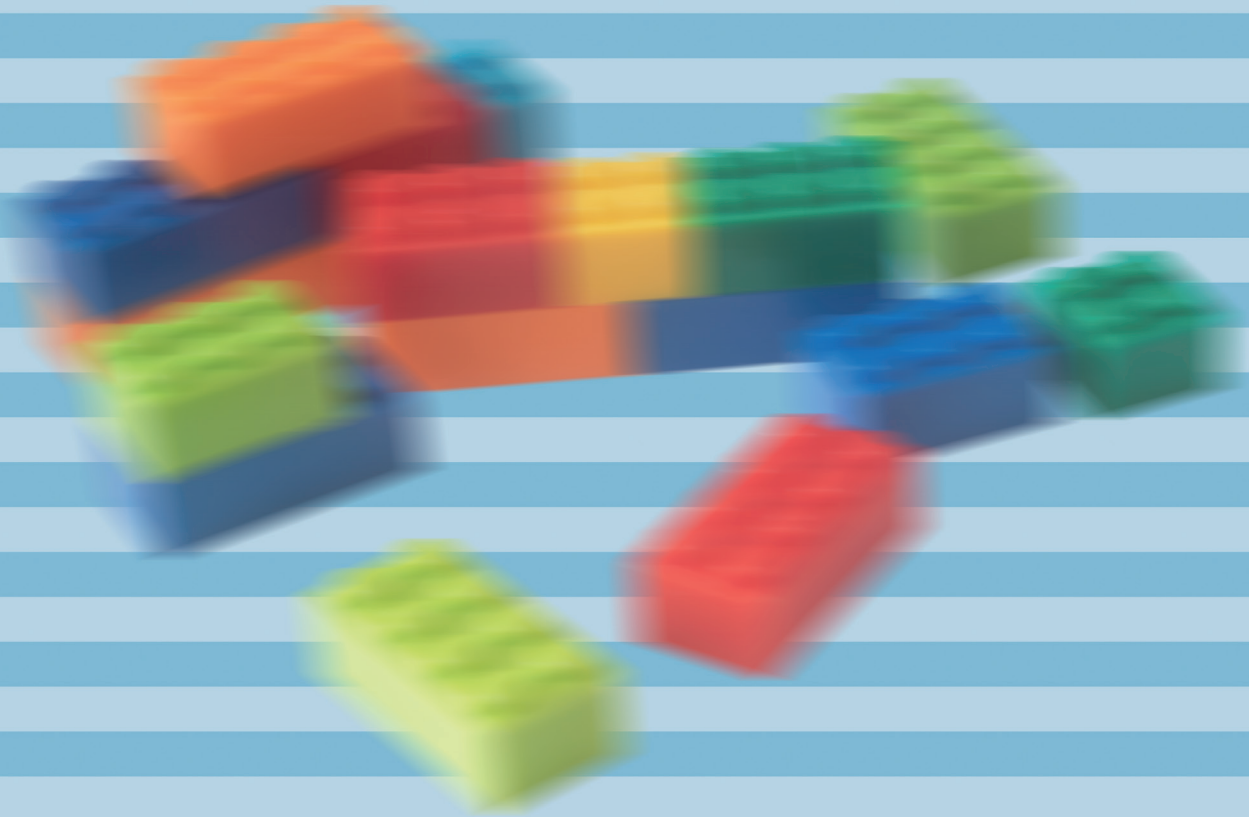
MISRA C 2012

組み込みプログラミングの高信頼性ガイド



MISRA-C 研究会　著

MISRA C:2012 解説書の発刊に寄せて

C プログラミング言語は、研究者用の言語として半世紀前に生まれ、いまだに多くの組込み系システム開発に用いられています。C 言語は、軽量で移植性に富み、機械語に近い処理から高度なサービスまで作れてしまう言語です。ですから OS はもちろん、アプリケーションに特化した専用言語の基底言語としても利用されています。21 世紀に入って一時は子供たちまでも C プログラムを書く時代がくるのかと思うほど解説書や教育の旋風が起きたものですが、そのブームは去り、どちらかというとシステム記述言語として社会の基盤を担う専門家の言語となっています。

言語としては軽量で、学ぶべき構文も簡潔ではあるのですが、簡単であるがゆえにトラップも多いのが C 言語です。そのトラップは、80 年代から 90 年代においても多くのシステムプログラマ達を悩ませました。一人のプログラマが如何に優秀であってもコーディングできる量には限界があります。それを越えた量のコードを生産するには優秀なリーダの下にサブリーダをつけ、さらにジュニアを配置することが一つの解決策です。この方法の弱点は、リーダの優れた感性の伝達率よりも、ジュニアプログラマが言語仕様のトラップで引き起こす不具合発生率がはるかに高かったことです。とくに 90 年代に、洋の東西を問わず C プログラマ人口が爆発的に増えた自動車業界では、高信頼性と安全性要求から C プログラムの品質管理に多くの企業が悩みました。

その状態を改善すべく生まれたのが MISRA-C プログラミングガイドでした。MISRA チームは、英国での 3 年に渡る考察と議論を経て自動車の制御系に用いる C プログラミングの注意すべき点、起こりうる様々なトラップを整理し簡潔明快に記述し 1998 年に英国でリリースされました。

日本では、2000 年に入ってから自動車産業での MISRA-C 準拠が進み、今や日本ばかりでなく、日本企業が開発拠点を置く北米の多くのサイトでも MISRA-C が利用されています。IPA（独立行政法人情報処理推進機構）は、自動車以外の産業界用に少し準拠性を緩めたコーディングガイドを MISRA と提携した上で発行し版を重ねています。結果として医療器から事務機、家電に至るまで MSIRA-C 準拠によって日本の組込みプログラム開発の品質は常に自動チェックされています。

こうしてみると、MISRA-C ガイドラインは組込みシステム開発業界にとってコンパイラと双璧をなす品質保証システムと呼んでも過言ではないでしょう。現に多くの組込み C/C++ 開発環境（IDE）ではコンパイラと MISRA-C がペアで使えるようになっています。MISRA は英国の自動車産業から生まれたガイドラインですが、今や C 言語でのシステム記述用に無くてはならないものになっています。

ソフトウェアの実行は、速いプロセッサ、大きなメモリ空間があればどんどん向上します。しかし、ソフトウェア開発は高価なコンパイラや大きなディスプレイでプログラミングしても必ずしも品質向上につながりません。我々が過去四半世紀に品質管理上の知恵として学んだのは、コンピュータやコンパイラとともにプログラミングの作法を開発文化として共有し、開発者同士がお互いに意味の通じるプログラムを作ることでした。MISRA-C は基底である C 言語プログラムの開発文化を形成する要石（楔石、キーストーン）となったのです。

その一方で残念なのは、MISRA-C はガイドラインとしての簡潔性から細かい説明があまり書かれていないことです。ここは MISRA-C の弱点ともいえます。これを補うための解説が本書の使命であり、著者のみなさんの努力によって MISRA-C の改版とともに本書も改良がなされてきました。まさに日本の組込みソフトウェア技術者、システム技術者の継続的な努力と信念のたまものと言えるでしょう。

こうして、日本の多くの企業が MISRA-C を活用できる状況になったいま、その利用人口は本書の発行で一段と増すことと思われます。では、このままプログラマ人口が増え続ければ良いのでしょうか？多分、それはソフトウェア社会にとっての正しい社会進化とは言えないでしょう。なぜなら、いたずらに就業人口が増えるだけの経済社会は必ず破綻するからです。

C 言語という特性を必要とする組込み製品やシステムソフトウェアの世界は、これから他者のソースコードを部品として組み込む時代に入ることでしょう。どのような会社や組織でも毎回 1 億行のソースコードをゼロから開発することはできないからです。そして、コードを部品として売る組織は、コードに値段を付けて直接売り買いするというよりも、コードに付帯するサービスを売買する可能性が高いでしょう。

その場合の買い手にとっての問題は何かというと膨大な C コードを自分たちの製品の部品として利用してよいかどうかの妥当性判断です。そして、売り手にとっての問題は、いかにして自社のコードが高品質であるかを示す社会的な品質ラベルを獲得することです。つまり、サービスに付帯するソフトウェア部品の品質を正しく評価できることが売り手、買い手双方の組織の繁栄を左右することになります。つまり、MISRA-C を使ってコード品質を保証できることが重要になるのです。

最近の自動車業界では、自動運転のソフトウェア開発が急務ですが、その開発ガイドライン ISO 26262 では、安全文化（Safety Culture）の重要性が指摘されています。この流れは今世紀初頭の開発プロセスに着目したころから明らかになっています。誕生して 100 年に満たないソフトウェア産業は未だに幼年期産業と呼べますが、その成長は著しく、今後は 20 世紀のハードウェア産業の進化過程と同様にソフトウェア品質を認証する力が重要という社会通念に発展することでしょう。そして国の政策としても大きな枠組みを作る方向へ進むであろうことは間違いありません。

我が日出る国ニッポンが、本書を通じて、プログラムを作ることだけから、取り入れるプログラムを監査し、しっかりとした枠組みの上に製品づくりをできるような国になることを願ってやみません。

2022 年 4 月

SESSAME 代表　飯塚悦功（東京大学名誉教授）

Foreword

MISRA C was originally published in April 1998 to assist the developers of automotive software in using the C language in a manner that is more suitable for safety-related applications. It was intended to fulfil the requirements for a language subset found in "Development Guidelines for Vehicle Based Software" produced by the original MISRA project and published in 1994. These guidelines, which set out the whole lifecycle requirements for developing safety-related software for road vehicles, required the use of a subset of a standardized structured programming language. More recently MISRA C is given as an example of one of the means of complying with requirements of ISO 26262 particularly for avoiding failures in the design and implementation of software units (see ISO 26262:2018 Part 6 Table 6).

Since its initial publication the uptake of MISRA C has far exceeded our expectations and it is now being used in embedded systems programming throughout the world and in most sectors, even in sectors where quality rather than safety is the main consideration. The applications where MISRA C is being used include the automotive, military, aerospace, rail transportation, medical devices and consumer electronics sectors.

In October 2004 we published MISRA-C:2004 (MISRA C2), and subsequently in March 2013 we published MISRA C:2012 (MISRA C3). This latest revision to MISRA C focused on clarity in compliance, for example making the following distinctions in guideline classifications:

The document now contains "guidelines" which are further divided into "directives" and "rules". Rules are constraints on language usage that are usually able to be checked by a static analysis tool. Directives cover wider topics such as restrictions on the software development environment.
A further "mandatory" classification of guidelines was introduced (in addition to "required" and "advisory"). It is not permitted to deviate from a "mandatory" guideline.
The scope of the document now covers C99 language features, and each guideline indicates which language variant it is applicable to (C90, C99, or both).
For rules, further classifications apply to aid the use of static analysis tools. A "decidable" rule is in theory always detectable by a static analysis tool. It is also indicated whether a rule can be checked at the software unit level ("single translation unit"), or needs checking during later software integration ("system"). This latter point is reflected in ISO 26262:2018 where static analysis is now also a requirement during software integration and verification.

Two further significant developments have taken place since publication of MISRA C:2012:

We have published more robust and extended guidance on managing compliance with MISRA C and handling deviations; this is found in the document MISRA Compliance:2016. We are very grateful for the support of the Japanese industry in helping to define this approach.
We have analysed MISRA C against guidelines for secure programming as found in ISO/TS 17961:2013 and CERT C. As a result, by introducing a small number of additional directives and rules, we have been able to extend the scope of MISRA C to cover both safe and secure programming in the same guidelines. MISRA C:2012 was reissued in February 2019 to incorporate these changes.

We appreciate the continued support that is being given to MISRA C in Japan and to this latest publication by the MISRA C Study Group. We hope that this new book will continue to contribute to the acceptance and wide usage of MISRA C.

Looking to the future, we will shortly publish a number of new MISRA documents including the MISRA Guidelines for Automotive Safety Arguments, enhancements to MISRA C to include support for C11 and C18, and an update to MISRA C++. A significant milestone in 2019 was that MISRA took over responsibility for the AUTOSAR C++ guidelines which will be incorporated into the new version of MISRA C++. We therefore look forward to continued co-operation to develop global standards for the development of software in embedded applications.

David Ward (Dr)
MISRA Project Manager
On behalf of the MISRA Steering Committee and the MISRA C Working Group

CONTENTS

1 はじめに

近年，様々な分野において電子技術の応用が著しく普及し，電子制御が多くのシステムに取り入れられている．多くの電子制御では，組込みソフトウェアと呼ばれる，装置を制御するためのソフトウェアが重要な要素となっており，このソフトウェアに対する要求がより高度で複雑なものへと推移するにつれ，ソフトウェアの品質および信頼性が極めて重要なものになっている．

このような状況の中，ヨーロッパの自動車業界では，1994 年に MISRA[1] が "Development guideline for vehicle based software" を発表した[2]．さらに MISRA は，1998 年に "Guidelines for the use of the C language in vehicle based software"(MISRA C:1998) を発表．これらのドキュメントは，MISRA ドキュメントと呼ばれている．

特に後者は MISRA C として，ヨーロッパだけではなく日本や北米の自動車業界におけるプログラミングガイドラインのベースとなった．2004 年 10 月にはタイトルを "Guidelines for the use of the C language in critical systems" と変えた第 2 版 (MISRA C:2004) が出版され，自動車業界以外にも広く普及した．さらに 2013 年 3 月には第 2 版の内容を改善し C99 対応など時流に合わせた第 3 版 (MISRA C:2012) が出版された .

日本では，MISRA ドキュメントを翻訳した「自動車用ソフトウェアの開発ガイドライン (TP-01001)」および「自動車用 C 言語利用のガイドライン (TP-01002)」が, 自動車技術会から発行された．さらに MISRA-C:2004 を翻訳した「JASOTP 自動車用 C 言語利用のガイドライン (TP-01002:2005)」も出版されているが，MISRA C:2012 の翻訳版は 2019 年 3 月現在発行されていない．

2002 年 4 月に発足し, SESSAME[3] の WG3 として活動してきた MISRA-C 研究会は，MISRA C:1998 のルールを解説した「組込み開発者におくる MISRA-C」を 2004 年 5 月に，MISRA C:2004 のルールを解説した「組込み開発者におくる MISRA-C:2004」を 2006 年 10 月に日本規格協会から発刊した．MISRA C:2012 の発表後はその検討に着手し，長い時間をかけて議論を重ねて，ようやくここに MISRA C:2012 対応の解説書を出版する運びとなった．

本書は，MISRA の了解のもと作成されたもので，日本国内における MISRA C 統一解釈を目指した内容となっている．MISRA C は，車載用ソフトウェアを対象としたプログラミングのガイドラインという位置づけにはあるが，その内容は車載用ソフトウェアに限らず，組込みソフトウェア開発者にとって有益な内容である事から，組込みソフトウェアの開発に取り組む多くの方々の参考になるものと思われる．

なお，本書の中で ISO/IEC 9899 からの引用として日本語の文が記載されているが，これについては，日本規格協会から発行されていた「プログラミング言語 C (JIS X 3010-1993)」、「プログラ