

CC-RL

C++14 テクニカルプレビュー版

ユーザーズマニュアル

対象リビジョン

V1.12.00

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ご注意書き

1. 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。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、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等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。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

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、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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いますが、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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の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。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 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・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
13. 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禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

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が直接的、間接的に支配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。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。

目次

1.	概要	4
1.1	テクニカルプレビュー版について	4
1.2	著作権について	4
2.	オプション仕様	5
2.1	C++14言語規格のコンパイル時に使用可能なオプション一覧	5
2.1.1	コンパイル・オプション	5
2.1.2	アセンブル・オプション	7
2.1.3	リンク・オプション	7
3.	コンパイラ言語仕様	8
3.1	基本言語仕様	8
3.1.1	サポートしていないC++言語仕様	8
3.1.2	C++14処理系定義	8
3.1.3	データの内部表現と領域	10
3.2	拡張言語仕様	12
3.2.1	予約語	12
3.2.2	マクロ	12
3.2.3	#pragma指令	12
3.2.4	組み込み関数	12
4.	セクション仕様	13
4.1	予約セクション名	13
5.	ライブラリ仕様	14
5.1	概要	14
5.2	提供ライブラリ	14
5.3	ヘッダ・ファイル	14
6.	スタートアップ	18
6.1	スタートアップ・ルーチン	18
6.1.1	静的記憶領域を持つクラス型オブジェクトの初期化	18
7.	メッセージ	19
7.1	出力形式	19
7.1.1	出力形式1	19
7.1.2	出力形式2	19
7.2	メッセージ種別	19
8.	注意事項	20
8.1	ソース・デバッグ用の情報出力	20

1. 概要

このマニュアルは、RL78 ファミリー用 C コンパイラ・パッケージ CC-RL V1.12.00 に含まれている、C++14 言語規格でコンパイルする `-lang=cpp14` オプションのテクニカルプレビュー版を使用する場合の仕様および注意事項について説明します。

CC-RL ユーザーズマニュアルとあわせて参照してください。

1.1 テクニカルプレビュー版について

テクニカルプレビュー版は、お客様からのフィードバックにより、より良い機能にすることを目的としています。CC-RL V1.12.00 において `-lang=cpp14` オプションはテクニカルプレビュー版となります。使用の際には以下の内容をご確認ください。

- `-lang=cpp14` オプションは評価用であり動作は保証しませんので、製品開発には使用しないでください。
- `-lang=cpp14` オプションを使用した場合には、次のメッセージを出力します。
W0519999:CC-RL V1.12.00 の `-lang=cpp14` オプションはテクニカルプレビュー版になります。製品開発には使用できません。
- 次の URL より、本機能へのご意見・ご要望をお寄せください。
<https://forms.office.com/r/wSGqp6BKic>

1.2 著作権について

本ソフトウェアは次のソフトウェアを利用しています。

- LLVM および Clang は University of Illinois at Urbana-Champaign が著作権を有します。
- Protocol Buffers は Google Inc. が著作権を有します。

C++ 用のライブラリは次のソフトウェアを利用しています。詳細はコンパイラに同梱されているライセンスファイルを参照してください。

- `compiler_rt`
- `libc++`
- `libc++abi`
- `newlib`

その他のソフトウェア構成物はルネサスエレクトロニクス株式会社が著作権を有します。

2. オプション仕様

C++14 言語規格でコンパイルする場合、次のオプションを指定してください。

```
-lang=cpp14
```

【詳細説明】

C++14 言語規格 (ISO/IEC 14882:2014) に沿ってコンパイルします。

- 本オプション指定時に、C ソース・ファイルを入力に指定した場合にはコンパイル・エラーとなります。入出力ファイルに関しては、CC-RL ユーザーズマニュアル "2.2 入出力ファイル"を参照してください。
- 本オプション指定時に、使用可能な他のオプションについては次節を参照してください。

2.1 C++14 言語規格のコンパイル時に使用可能なオプション一覧

2.1.1 コンパイル・オプション

-lang=cpp14 オプションと同時に指定して使用することができるオプションを示します。"×"のオプションは指定を無視するか、エラーメッセージを出力します。

表 1 -lang=cpp14 オプションと同時に指定することができるオプション

分類	オプション	同時指定	補足
バージョン表示指定	-V	✓	
ヘルプ指定	-help	✓	
出力ファイル指定	-o	✓	
	-obj_path	✓	
	-asm_path	✓	
	-prep_path	✓	
ソース・デバッグ制御	-g	✓	C++言語仕様によっては、ソース・デバッグ情報の出力に対応していないものがあります。注意事項を参照してください。
	-g_line	✓	
デバイス指定関係	-cpu	✓	
	-use_mda	✓	
処理中断指定	-P	✓	
	-S	✓	
	-c	✓	
プリプロセッサ制御	-D	✓	
	-U	✓	
	-I	✓	
	-preinclude	✓	
	-preprocess	×	

表 1 -lang=cpp14 オプションと同時に指定することができるオプション(2)

分類	オプション	同時指定	補足
メモリ・モデル	-memory_model={small medium}	✓	
	-far_rom	✓	プログラムによっては内部エラーが発生する場合があります。
最適化	-O{ size speed default lite nothing }	✓	
	-goptimize	✓	
個別最適化	-Oinline_level[=数値]	✓	
	-Oinline_size[=数値]	✓	
	-Opipeline[={on off}]	✓	
	-Ounroll[=数値]	✓	
	-Otail_call[={on off}]	✓	
	-Odelete_static_func[={on off}]	✓	
	-Omerge_files	×	
	-Ointermodule	×	
	-Owhile_program	×	
	-Oalias={ansi noansi}	×	
	-Osame_code={on off}	✓	
付加情報出力	-cref	×	
	-pass_source	✓	
エラー制御	-no_warning_num	✓	対象とするメッセージ番号は次の通りです。(W052xxxx は対象外です) W0510000 ~ W0519999, W0530000 ~ W0559999
	-change_message	✓	対象とするメッセージ番号は次の通りです。(W052xxxx は対象外です) W0510000 ~ W0519999, W0530000 ~ W0549999
	-error_file	✓	
生成コード変更	-dbl_size={4 8}	✓	
	-signed_char	✓	
	-signed_bitfield	×	オプションによる制御はできません。 signed も unsigned も付かない型のビット・フィールドは、符号付きとして扱います。(※-lang=c および c99 指定時は解釈が異なり、符号なしとして扱います)
	-switch	✓	
	-volatile	×	

表 1 -lang=cpp14 オプションと同時に指定することができるオプション(3)

分類	オプション	同時指定	補足
生成コード変更	-merge_string	×	
	-pack	✓	
	-stuff	×	
	-stack_protector	×	
	-stack_protector_all		
	-insert_nop_with_label	×	
	-control_flow_integrity	×	
拡張機能	-strict_std	×	
	-refs_without_declaration	×	
	-large_variable	×	
	-nest_comment	×	
	-character_set	×	オプションによる制御はできません。 入力ファイル中の文字列は UTF-8 として扱います。
MISRA 対応	-misra2004	×	
	-misra2012	×	
	-ignore_files_misra	×	
	-check_language_extension	×	
	-misra_intermodule	×	
パラメータ・ファイル指定	-subcommand	✓	
アセンブラ・リンク制御	-asmopt=arg	✓	
	-lnkopt=arg	✓	
	-asmcmd=filename	✓	
	-lnkcmd=filename	✓	
	-dev=filename	✓	
コンパイラ移行支援機能	-convert_cc={ca78k0r nc30 iar}	×	
	-unaligned_pointer_for_ca78k0r	×	

2.1.2 アセンブル・オプション

全てのアセンブル・オプションを使用可能です。

2.1.3 リンク・オプション

次のオプションは、-lang=cpp14 オプションと同時に使用できません。その他のオプションは使用可能です。

表 2 -lang=cpp14 オプションと同時に指定することができるオプション

分類	オプション	同時指定	補足
出力制御	-VFINFO	×	

3. コンパイラ言語仕様

3.1 基本言語仕様

3.1.1 サポートしていない C++ 言語仕様

以下の仕様はサポートしていません。

- 例外処理
- 実行時型情報
- スレッド
- アトミック計算

3.1.2 C++14 処理系定義

処理系定義について説明します。

表 3 処理系定義

section No.	Item	Description
1.3.6	diagnostic message	"7. メッセージ"を参照してください。
1.4	required libraries for freestanding implementation	"5. ライブラリ仕様"を参照してください。
1.7	bits in a byte	8 bits です。
1.9	interactive device	対話型装置の構成については、特に規定しません。
1.10	number of threads in a program under a freestanding implementation	マルチスレッドはサポートしていません。
2.2	mapping physical source file characters to basic source character set	UTF-8 としてそのままマッピングします。
2.2	physical source file characters	UTF-8 です。
2.2	converting characters from source character set to execution character set	UTF-8 のまま同じです。
2.2	whether source of translation units must be available to locate template definitions	ソースは不要です。
2.9	mapping header name to header or external source file	記述通りに解釈してファイル名に対応付けます。
2.14.3	value of multicharacter literal	実行時文字集合の、下位 4 バイトの値を持ちます。
2.14.3	value of wide-character literal containing multiple characters	実行時文字集合の、最後の 1 文字の値を持ちます。
2.14.3	value of wide-character literal with single c-char that is not in execution wide-character set	その文字の値を持ちます。
2.14.3	encoding of universal character name not in execution character set	その文字の値を持ちます。
2.14.3	semantics of non-standard escape sequences	¥e と ¥E が有効です。値は共に 0x1b です。
2.14.3	value of character literal outside range of corresponding type	エラーとなります。
2.14.5	concatenation of some types of string literals	エラーとなります。

表 3 処理系定義(2)

section No.	Item	Description
3.6.1	defining main in freestanding environment	規定しません。
3.6.1	parameters to main	規定しません。
3.6.1	start-up and termination in freestanding environment	規定しません。スタートアップの実装に依存します。
3.6.1	linkage of main	C リンケージになります。
3.6.2	dynamic initialization of static objects before main	スタートアップの実装に依存します。
3.6.2	dynamic initialization of thread-local objects before entry	スレッドはサポートしません。
3.9.1	extended signed integer types	サポートしていません。
3.9.1	representation of char	1 バイト
3.9.1	signedness of char	Unsigned char 型です。ただし、オプションで切り替えることができます。
3.9.1	value representation of floating-point types	IEEE754 に準拠します。
3.9.2	value representation of pointer types	"3.1.3 データの内部表現と領域"を参照してください。
3.11	alignment	"3.1.3 データの内部表現と領域"を参照してください。
4.13	rank of extended signed integer type	サポートしていません。
5.3.3	sizeof applied to fundamental types other than char, signed char, and unsigned char	"3.1.3 データの内部表現と領域"を参照してください。
5.3.4	support for over-aligned types	サポートしていません。
5.8	result of right shift of negative value	算術シフトです。
7.2	underlying type for enumeration	"3.1.3 データの内部表現と領域"を参照してください。
7.4	meaning of asm declaration	サポートしていません。
8.4.1	string resulting from <code>__func__</code>	関数名を返します。
16.2	nesting limit for <code>#include</code> directives	メモリに依存します。
16.6	<code>#pragma</code>	<code>#pragma</code> 指令 を参照
16.8	text of <code>__DATE__</code> when date of translation is not available	常に得ることができます。
16.8	text of <code>__TIME__</code> when time of translation is not available	常に得ることができます。
16.8	definition and meaning of <code>__STDC__</code>	1 で定義します。
16.8	definition and meaning of <code>__STDC_VERSION__</code>	定義しません。
17.6.5.12	exceptions thrown by standard library functions that do not have an exception specification	例外処理はサポートしていません。
18.2	type of <code>size_t</code>	unsigned int 型です。
18.5	exit status	規定しません。

3.1.3 データの内部表現と領域

この項では、CC-RL が扱うデータのそれぞれの型における、内部表現と値域について説明します。

(1) 基本型

表 4 基本型

Data Type	Size (byte)	Alignment (byte)	Signed/Unsigned	Data range Minimum Value	Maximum Value	Note
char	1	1	Unsigned	0	+255	-signed_char オプションを指定した場合、signed char 型と同じ値の範囲を持ちます。
signed char	1	1	Signed	-128	+127	
unsigned char	1	1	Unsigned	0	+255	
short	2	2	Signed	-32768	+32767	
signed short	2	2	Signed	-32768	+32767	
unsigned short	2	2	Unsigned	0	+65535	
int	2	2	Signed	-32768	+32767	
signed int	2	2	Signed	-32768	+32767	
unsigned int	2	2	Unsigned	0	+65535	
long	4	2	Signed	-2147483648	+2147483647	
signed long	4	2	Signed	-2147483648	+2147483647	
unsigned long	4	2	Unsigned	0	+4294967295	
long long	8	2	Signed	-9223372036854775808	+9223372036854775807	
signed long long	8	2	Signed	-9223372036854775808	+9223372036854775807	
unsigned long long	8	2	Unsigned	0	+18446744073709551615	
wchar_t	2	2	Unsigned	0	+65535	
char16_t	2	2	Unsigned	0	+65535	
char32_t	4	2	Unsigned	0	+4294967295	
bool	1	1	Unsigned	-	-	ゼロビットのみが意味を持ちます。1~7 ビット目は不定値となります。
float	4	2	Signed	1.17549435E-38F	3.40282347E+38F	

表 4 基本型(2)

Data Type	Size (byte)	Alignment (byte)	Signed/Unsigned	Data range Minimum Value	Maximum Value	Note
double (-double_size=4)	4	2	Signed	1.17549435E-38F	3.40282347E+38F	
double (-double_size=8)	8	2	Signed	2.2250738585072014E-308	1.7976931348623158E+308	
long double (-double_size=4)	4	2	Signed	1.17549435E-38F	3.40282347E+38F	double 型とサイズは同じだが型は異なる。
long double (-double_size=8)	8	2	Signed	2.2250738585072014E-308	1.7976931348623158E+308	double 型とサイズは同じだが型は異なる。

(2) 複合型

- ポインタ型、配列型

表 5 ポインタ型、配列型

Data Type		Size(byte)	Alignment(byte)
ポインタ	near ポインタ	2	2
	far ポインタ	4	2
左辺値参照	near 参照	2	2
右辺値参照	far 参照	4	2
データメンバへのポインタ		2	2
メンバ関数へのポインタ		4	2
配列		配列要素のサイズ×要素の数	配列要素の整列条件

- 列挙型

表 6 列挙型

列挙子の限界値	列挙子の上限値	内部表現の型	備考
-128	127	signed char	-
0	255	unsigned char	全列挙子が 0 - 127 の範囲に収まる場合はここに含む
-32768	32767	signed short	-
0	65535	unsigned short	全列挙子が 0-65535 の範囲に収まる場合はここに含む
-2147483647	2147483647	signed long	-
0	4294967295	unsigned long	全列挙子が 0-4294967295 の範囲に収まる場合はここに含む
-9223372036854775808	9223372036854775807	signed long long	-
上記以外		signed long long	警告を出力する。

3.2 拡張言語仕様

3.2.1 予約語

CC-RL で追加されている予約語については、CC-RL ユーザーズマニュアル "4.2.1 予約語"を参照してください。

3.2.2 マクロ

-lang オプションのパラメータによって定義が異なるマクロを示します。その他のマクロについては CC-RL ユーザーズマニュアル "4.2.2 マクロ"をあわせて参照してください。

値は全て 10 進定数です。

表 7 マクロ

マクロ名	-lang=cpp14 指定時の定義	-lang=c,c99 指定時の定義
__cplusplus	201402L	定義しません
__clang__	1	定義しません
__STDC_HOSTED__	0	0 (-lang=c99 オプションを指定した場合に定義)
__STDC__	1	1 (-strict_std オプションを指定した場合に定義)
__STDC_VERSION__	定義しません	199409L(-lang=c かつ-strict_std オプションを指定した場合に定義) 199901L (-lang=c99 オプションを指定した場合に定義)
__STDC_IEC_559__	定義しません	1 (-lang=c99 オプションを指定した場合に定義)

3.2.3 #pragma 指令

CC-RL ユーザーズマニュアル "4.2.4 #pragma 指令"に記載の#pragma 指令は使用できません。

3.2.4 組み込み関数

CC-RL ユーザーズマニュアル "4.2.7 組み込み関数"に記載の組み込み関数を使用することができます。詳細は CC-RL ユーザーズマニュアルを参照してください。

4. セクション仕様

C++14 言語規格でコンパイルする場合に予約されているセクション名と再配置属性を示します。その他のセクション仕様に関しては、CC-RL ユーザーズマニュアルを参照してください。

4.1 予約セクション名

表 8 予約セクション名

Default Section Name	Relocation Attribute	Description
.init_array	CONSTF	グローバルコンストラクタ用セクション
.callt0	CALLT0	callt 関数呼び出しのテーブル用セクション
.text	TEXT	コード部用セクション (near 領域配置)
.textf	TEXTF	コード部用セクション (far 領域配置)
.textf_unit64kp	TEXTF_UNIT64KP	コード部用セクション (セクションを先頭が偶数番地になるように、64KB-1 境界にまたがらないように配置)
.const	CONST	ROM データ (near 領域配置) (ミラー領域内)
.constf	CONSTF	ROM データ (far 領域配置)
.data	DATA	near 初期化データ用セクション (初期値あり)
.dataf	DATAF	far 初期化データ用セクション (初期値あり)
.sdata	SDATA	初期化データ用セクション (初期値あり, saddr 配置変数)
.bss	BSS	データ領域用セクション (初期値なし, near 領域配置)
.bssf	BSSF	データ領域用セクション (初期値なし, far 領域配置)
.sbss	SBSS	データ領域用セクション (初期値なし, saddr 配置変数)
.option_byte	OPT_BYTE	ユーザ・オプション・バイト, およびオンチップ・デバッグ指定専用セクション
.security_id	SECUR_ID	セキュリティ ID 指定専用セクション
.flash_security_id	FLASH_SECUR_ID	フラッシュ・プログラマ・セキュリティ ID 指定専用セクション
.vect< ベクタテーブル・アドレス>	AT	割り込みベクタテーブル -split_vect オプションを指定したとき, .vect< ベクタテーブル・アドレス> でセクションを生成します。ベクタテーブル・アドレスは 16 進数表記になります。

5. ライブラリ仕様

5.1 概要

CC-RL C++14 テクニカルプレビュー版では、C++ソース・プログラムのコンパイル用に次のソフトウェアを用いたライブラリを提供しています。各ソフトウェアの詳細はコンパイラ・パッケージに同梱されているライブラリ・ソースを参照してください。

- compiler_rt
- libc++
- libc++abi
- newlib

5.2 提供ライブラリ

-cpu オプションによる CPU コアの種別 S1、S2、S3 のそれぞれに対して、次の 6 種類のライブラリを提供しています。-lang=cpp14 用のライブラリであり、-lang=c または c99 指定時の使用はサポートしていません。

表 9 提供ライブラリ名

ライブラリ名	概要
rl78_libc.lib	C99 準拠標準ライブラリ
rl78_libm.lib	数学関数ライブラリ
rl78_libgloss.lib	低水準ライブラリ
rl78_libcxx.lib	C++標準ライブラリ
rl78_libcxxabi.lib	C++ランタイムライブラリ
rl78_compiler-rt.lib	コンパイラ用のランタイムライブラリ

※ S2 コア用のライブラリは、乗除積和演算器 (-use_mda=mda) を使用しています。

※ 全てのライブラリは、単精度浮動小数点数 (-dbl_size=4) を使用しています。

5.3 ヘッダ・ファイル

ヘッダ・ファイルを示します。

表 10 ヘッダ・ファイル

分類	ヘッダ・ファイル	説明
C99 準拠標準ライブラリ	<assert.h>	診断機能
	<complex.h>	複素数計算
	<ctype.h>	文字操作
	<errno.h>	エラー
	<float.h>	浮動小数点型の特性
	<inttypes.h>	整数型の書式変換
	<iso646.h>	代替つづり
	<limits.h>	整数型の大きさ
	<locale.h>	文化圏固有操作

表 10 ヘッダ・ファイル(2)

分類	ヘッダ・ファイル	説明
C99 準拠標準ライブラリ	<math.h>	数学
	<setjmp.h>	非局所分岐
	<signal.h>	シグナル操作
	<stdarg.h>	可変個数の実引数
	<stdbool.h>	論理型及び論理値
	<stddef.h>	共通の定義
	<stdint.h>	整数型
	<stdio.h>	入出力
	<stdlib.h>	一般ユーティリティ
	<string.h>	文字列操作
	<tgmath.h>	型総称数学関数
	<time.h>	日付及び時間
	<wchar.h>	多バイト文字及びワイド文字拡張ユーティリティ
	<wctype.h>	ワイド文字種類及びワイド文字大文字小文字変換ユーティリティ
C++標準ライブラリ	<algorithm>	アルゴリズム
	<array>	配列
	<bitset>	ビットの固定サイズシーケンス
	<chrono>	時間ユーティリティ
	<codecvt>	コード変換ファセット
	<complex>	複素数
	<condition_variable>	条件変数
	<deque>	両端キュー
	<exception>	例外ハンドリング
	<forward_list>	単方向リスト
	<fstream>	ファイルストリーム
	<functional>	関数オブジェクト
	<future>	Future パターン
	<initializer_list>	初期化子リスト
	<iomanip>	フォーマットとマニピュレータ
	<ios>	iostream 基底クラス
	<iosfwd>	先行宣言
	<iostream>	標準 iostream オブジェクト
	<istream>	入力ストリーム
	<iterator>	イテレータの定義
	<limits>	算術型の実装プロパティ
	<list>	双方向リスト
	<locale>	ローカライゼーション (地域化)
	<map>	連想配列
	<memory>	メモリ

表 10 ヘッダ・ファイル(3)

分類	ヘッダ・ファイル	説明
C++標準ライブラリ	<mutex>	ミューテックス
	<new>	動的メモリ管理
	<numeric>	一般的な数値操作
	<ostream>	出カストリーム
	<queue>	FIFO キュー
	<random>	乱数生成
	<ratio>	コンパイル時有理数
	<regex>	正規表現
	<scoped_allocator>	スコープ付きアロケータ
	<set>	集合
	<sstream>	文字列ストリーム
	<stack>	LIFO スタック
	<stdexcept>	例外クラス
	<streambuf>	ストリームバッファ
	<string>	文字列クラスと、文字特性
	<system_error>	システムエラーサポート
	<tuple>	タプル
	<type_traits>	型特性
	<typeindex>	型のインデックス
	<unordered_map>	非順序連想配列
	<unordered_set>	非順序集合
	<utility>	ユーティリティコンポーネント
	<valarray>	数値の配列
<vector>	ベクタ配列	
C 言語互換ライブラリ	<cassert>	assert.h の互換
	<ccomplex>	complex.h の互換
	<cctype>	ctype.h の互換
	<cerrno>	errno.h の互換
	<cfloat>	float.h の互換
	< cinttypes>	inttypes.h の互換
	<ciso646>	iso646.h の互換
	<climits>	limits.h の互換
	<locale>	locale.h の互換
	<cmath>	math.h の互換
	<csetjmp>	setjmp.h の互換
	<csignal>	signal.h の互換
	<cstdarg>	stdarg.h の互換
	<cstdbool>	stdbool.h の互換
	<cstddef>	stddef.h の互換
	<cstdint>	stdint.h の互換
	<cstdio>	stdio.h の互換

表 10 ヘッダ・ファイル(4)

分類	ヘッダ・ファイル	説明
C 言語互換ライブラリ	<cstdlib>	stdlib.h の互換
	<cstring>	string.h の互換
	<ctgmath>	tgmath.h の互換
	<ctime>	time.h の互換
	<wchar>	wcchar.h の互換
	<wctype>	wctype.h の互換

6. スタートアップ

6.1 スタートアップ・ルーチン

main 関数を実行する前に、CC-RL ユーザーズマニュアル "8.2 スタートアップ・ルーチン"に掲載されている処理の他に、下記を実施してください。

- 静的記憶領域を持つクラス型オブジェクトの初期化

6.1.1 静的記憶領域を持つクラス型オブジェクトの初期化

静的記憶領域を持つクラス型オブジェクトのコンストラクタを呼び出してください。

コンストラクタのアドレスは .init_array セクションに配置されています。これを呼び出すには、スタートアップ・ルーチン内に次のように記述します。

```
MOVW    BC,#LOWW(SIZEOF(.init_array))
BR      $.L2_INIT
.L1_INIT:
DECW    BC
DECW    BC
MOV     ES,#HIGHW(STARTOF(.init_array))
MOVW    AX,ES:LOWW(STARTOF(.init_array))[BC]
MOV     CS,#0x00
PUSH    BC
CALL    AX
POP     BC
.L2_INIT:
CLRW    AX
CMPW    AX,BC
BNZ     $.L1_INIT
```

7. メッセージ

7.1 出力形式

-lang=cpp14 オプションを指定した場合に出力するメッセージには、2種類の出力形式があります。

7.1.1 出力形式 1

CC-RL ユーザーズマニュアル "10. メッセージ"で説明しているメッセージ番号を含む出力形式です。詳細は CC-RL ユーザーズマニュアル を参照してください。

出力するメッセージは、以下のメッセージ番号が対象です。

0510000 番台、0530000 番台、0540000 番台、0550000 番台、0560000 番台

(1) ファイル名と行番号を含む場合

```
ファイル名 (行番号) : メッセージ種別 05 メッセージ番号 : メッセージ
```

(2) ファイル名を含まない場合

```
メッセージ種別 05 メッセージ番号 : メッセージ
```

7.1.2 出力形式 2

メッセージの出力形式は次の通りです。

(1) ファイル名、行番号、カラム番号を含む場合

```
ファイル名 : 行番号 : カラム番号 : メッセージ種別 : メッセージ
```

(2) ファイル名を含まない場合

```
メッセージ種別 : メッセージ
```

7.2 メッセージ種別

メッセージ種別は次の通りです。

表 11 メッセージ種別一覧表

メッセージ種別		説明
出力形式 1	出力形式 2	
C	-	内部エラー : 処理を中止します。 オブジェクト・コードは生成しません。
E	error	エラー : 一定数以上発生した場合、処理を中止します。 オブジェクト・コードは生成しません。
F	fatal	致命的エラー : 処理を中止します。 オブジェクト・コードは生成しません。
M	remark	インフォメーション : 処理を続行します。 オブジェクト・コードを生成します。
W	warning	ワーニング : 処理を続行します。 オブジェクト・コードを生成します。(ユーザが意図したものと異なる可能性があります。)
-	note	他のメッセージに付随する情報です。

8. 注意事項

8.1 ソース・デバッグ用の情報出力

次の言語仕様に関連するソース・デバッグ用の情報出力には対応していません。

- 無名共用体
- 名前空間
- 派生クラス
 - 仮想基底クラス
 - 仮想関数
 - 純粋仮想関数
- テンプレート

改訂記録	CC-RL C++14 テクニカルプレビュー版 ユーザーズマニュアル
------	------------------------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Jan.20.23	—	初版発行

CC-RL

C++14テクニカルプレビュー版ユーザーズマニュアル

発行年月日 2023年1月20日 Rev.1.00

発行 ルネサス エレクトロニクス株式会社
〒135-0061 東京都江東区豊洲3-2-24 (豊洲フォレシア)

CC-RL
C++14 テクニカルプレビュー版
ユーザーズマニュアル