

統合開発環境 e² studio

ユーザーズマニュアル 入門ガイド

ルネサスマイクロプロセッサ RZファミリ

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。

ルネサスエレクトロニクス www.renesas.com

Rev.1.03 2023.06

ご注意書き

- 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。
- 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。
- 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

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。

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/

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカ ルアップデートを参照してください。

1. 静電気対策

CMOS製品の取り扱いの際は静電気防止を心がけてください。CMOS製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存 の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを 施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS製品を実装したボードについても同様の扱いをし てください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入によ り、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記 載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後 に切り替えてください。リセット時、外部発振子(または外部発振回路)を用いたクロックで動作を開始するシステムでは、クロックが十分安定した 後、リセットを解除してください。また、プログラムの途中で外部発振子(または外部発振回路)を用いたクロックに切り替える場合は、切り替え先 のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS製品の入力がノイズなどに起因して、V_π(Max.)から V_H(Min.)までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_π(Max.)からV_H (Min.)までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス(予約領域)のアクセス禁止

リザーブアドレス(予約領域)のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス(予約領域) があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュ メモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合がありま す。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

このマニュアルの使い方

このマニュアルは、アプリケーション・システムを開発する際の統合開発環境である e² studio について説 明します。

e² studio は RX ファミリ、RL78 ファミリ、RZ ファミリ用の統合開発環境で、ソフトウェア開発における、設計、実装、デバッグなどの各開発フェーズに必要なツールを単一のプラットフォームに統合しています。

統合することで、さまざまなツールを使い分けずにすべての開発を本製品のみで行うことができます。

対象者	このマニュアルは、e ² studio ョン・システムを開発するユー	を使用してソフトウェアやハードウェアのアプリケーシ ーザを対象としています。	
目的	このマニュアルは、ユーザが アのシステム開発を始める際	ターゲットデバイスを使用してハードウェアやソフトウェ ぽの e ² studio の機能を説明します。	
構成	このマニュアルは、大きく分り	ナて次の内容で構成しています。	
	第1章 概説 第2章 インストール 第3章 プロジェクトのインオ 第4章 ビルド 第5章 デバッグ 第6章 ヘルプ	ペートおよび作成	
読み方	このマニュアルを読むにあたっては、電気、論理回路、マイクロプロセッサに関する 一般知識が必要となります。		
凡例	データ表記の重み	: 左が上位桁、右が下位桁	
	アクティブ・ロウの表記	:XXX(端子、信号名称に上線)	
	注	:本文中につけた注の説明	
	注意	:気をつけて読んでいただきたい内容	
	備考	:本文中の補足説明	
	数の表記	:10 進数 XXXX	

目次

1. 概言	説	1
1.1	システム構成	1
1.2	システム要件	1
1.3	サポートするコンパイラ	2
1.4	コンパイラをアップデートするための追加ツール	2
1.5	サポートするエミュレータ	3
2. イン	ンストール	4
2.1	e ² studioインストーラを用いたインストール(64ビット版)	4
2.2	e ² studioインストーラを用いたインストール(32ビット版)	15
2.3	プラットフォームインストーラを用いたインストール	24
2.4	追加ソフトウェアのインストール	35
2.5	e² studioのアンインストール	
2.6	ツールチェーンへのe ² studioの登録	38
3. プ	ロジェクトのインポートおよび作成	41
3.1	既存プロジェクトのインポート	41
3.2	新規プロジェクトの作成	44
3.3	プロジェクトをバックアップする際の注意事項	91
4. ビノ	ルド	92
4.1	ビルドオプションの設定	92
4.2	プロジェクトのビルド	95
4.3	ビルド構成のレポート	97
5. ディ	バッグ	98
5.1	既存デバッグ構成の変更	99
5.2	新規デバッグ構成の作成	103
5.3	Launch Bar	105
5.4	基本的なデバッグ機能	106
5.5	デバッグ手順(RZ/A2Mの場合)	123
6. ~)	ルプ	124

1. 概説

ルネサス製マイクロプロセッサをサポートする e² studio は、Eclipse ベースの組み込み向け開発環境で す。e² studio は、ビルドフェーズ(エディタ、コンパイラ、リンカコントロールなど)から、拡張 GDB イ ンタフェースをサポートするデバッグフェーズをサポートするオープンソース Eclipse CDT (C/C++ 開発ツ ール)プロジェクトを基本に開発されたものです。

この章では、ハイエンド ARM ベースのルネサス製マイクロプロセッサである RZ ファミリ向けアプリケー ションを開発するための、e² studio のシステム構成と動作環境を説明します。

1.1 システム構成

一般的なシステム構成の例を以下に示します。



図 1-1 システム構成(RZ/A1H Renesas Starter Kit の例)

1.2 システム要件

本製品に対するシステムの必要条件を以下に示します。

PC ハードウェア環境:

- プロセッサ : 1GHz 以上(ハイパースレッディング及びマルチコア CPU をサポートする)
- メインメモリ : 4GB 以上
- ハードディスク容量 : 空き容量 2GB 以上
- ディスプレイ : 解像度 1,024 x 768 ピクセル以上; 65,536 色以上
- インタフェース : USB 2.0 (ハイスピードまたはフルスピード) ハイスピードが望ましい

動作環境:

- Windows® 10 (32 ビット版、64 ビット版)
- Windows® 8.1 (32 ビット版、64 ビット版)
- •

e² studio 64ビット版(2020-04以降)は64ビット版のOSが必須です。

e² studio ユーザーズマニュアル

Windows 64ビット版:

- システム : x64 ベースプロセッサ 推奨 2GHz 以上 (マルチコア CPU に対応)
 - Windows® 11
 - ・Windows® 10 (64 ビット版)
- メモリ容量 : 推奨 8GB 以上。最低 4GB 以上
- ハードディスク容量 : 空き容量 2GB 以上
- ディスプレイ : 1024×768 以上の解像度, 65536 色以上
- インタフェース : USB 2.0
- Microsoft Visual C++ 2010 SP1 ランタイムライブラリ *1
- Microsoft Visual C++ 2015-2022 ランタイムライブラリ *1

*1. e² studioインストール時に一緒にインストールされます。

1.3 サポートするコンパイラ

RZ ファミリデバイスでは、ARM アーキテクチャ用 GNU コンパイラを使用します。(各デバイスには、 GNU コンパイラの推奨バージョンがあります。詳細は、<u>https://www.renesas.com/rz</u>を参照してくださ い。) e² studio を[追加ソフトウェア]でインストールする際、コンパイラをインストールすることができます (インターネット接続が必要です。2.1 節を参照してください)。あるいは、コンパイラを以下の Web ペー ジからダウンロードして、個別にインストールすることもできます(ツールチェーンの登録が必要です)。

- GNU ARM Embedded toolchain (GCC 6; 6-2016-q4-major または上位バージョン): https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads
- GNU ARM Embedded toolchain (GCC^注 5; 5-2016-q3-update または下位バージョン): <u>https://launchpad.net/gcc-arm-embedded/+download</u>
- GNUARM-NONE toolchain: <u>https://gcc-renesas.com/rz/rz-download-toolchains/</u>

コンパイラのインストールに関しては 2.2 節、ツールチェーンを統合してコンパイラを e² studio に登録するには、2.6 節を参照してください。

注意: 'GCC'とは、'GNU Compiler Collection'を指します。(<u>https://gcc.gnu.org/</u>を参照してください。)

1.4 コンパイラをアップデートするための追加ツール

'Libgen Update for GNU ARM Embedded Toolchains' は、組み込みシステム向けに実装される標準 C ライ ブラリである、'newlib' 専用ライブラリのビルドに使用されるツールです。ライブラリの"prebuild"バージョ ン (ツールチェーンを作成するときの構成でビルドされるライブラリ)の代わりに、ユーザは、カスタムオ プションを使用し、Libgen Update によって独自の"projectbuild"ライブラリビルド構成を持つことができま す。

e² studio をインストールする際、Libgen Update for GNU ARM Embedded Toolchains を選択してインスト ールすることができます(インターネット接続が必要です)。あるいは、コンパイラを以下の Web ページか らダウンロードすることができます。

 Libgen Update for GNU ARM embedded toolchain (V1.2022.09 またはその上位バージョン) https://gcc-renesas.com/rz/rz-download-toolchains/

Libgen Update for GNU ARM Embedded Toolchains のインストールに関しては、2.4 節を参照してくださ

い。

1.5 サポートするエミュレータ

Segger J-Link Lite (RZ) および、その他エミュレータ 詳細は、RZ ファミリ製品の Web ページ (<u>https://www.renesas.com/rz</u>) を参照してください。



2. インストール

最新の e² studio 統合開発環境インストーラパッケージはルネサスウェブサイトから無償でダウンロードで きます。ルネサスエレクトロニクスは現在 e² studio の 32 ビット版と 64 ビット版両方をサポート致しま す。

詳細な情報は e² studio の製品情報ページ(<u>https://www.renesas.com/e2studio</u>) から確認できます。ソフト ウェアをダウンロードするには MyRenesas ページのルネサスアカウントにログインしてください。

この章では、e² studio のインストールおよびアンインストールについて説明します。

e² studio インストーラは e² studio の「インストール」および「アップグレード」、「修正」を提供致しま す。ただし、V5.x から V6.y もしくは V6.mから V7.n などのインストーラとメジャーバージョンが異なる e² studio には「アップグレード」や「修正」を適用しないでください。

「インストール」を選択して別なフォルダにインストールするか、旧版を一旦アンインストールしてください。

RZ/T2, RZ/N2, RZ/A3UL については、プラットフォームインストーラ(FSP with e² studio Installer)また は標準の e² studio インストーラのいずれかを用いてインストールできます。

- プラットフォームインストーラを用いたインストールを行う場合の要件は以下です。 ・Windows® 11、Windows® 10 (64 ビット版)
- インストールされる e² studio のバージョンは以下です。
 - ・e² studio 2023-01 または上位バージョン

RZ/G2L については、標準の e² studio インストーラを用いてインストールし、必要であれば後から FSP を 個別でインストールすることができます。

- 2.1 e² studio インストーラを用いたインストール(64 ビット版)
- e² studioインストーラをダブルクリックしてe² studioインストールウィザードページを開いてください。 インストーラパッケージはルネサスウェブサイトから無償でダウンロードできます。

注: e² studio が既に PC にインストールされた場合は図 2-1 の通りに変更、削除、インストールの選択肢が表示されます。



🛃 Renesas e² studio 2023-04 Setup	— 🗆 X
Renesas e² studio 2023-04 Setup	RENESAS
e2 studio version 23.4.0.R20230406-1341はすでにインストールされています。	
何をしたいですか?	
 変更 インストール済みの機能を追加または削除します。 場所: C:¥Renesas¥e2_studio 削除 既存のバージョンをアンインストールします。 	
場所:C:¥Renesas¥e2_studio インストール 別の場所にインストールします。	
<u>v202304061529</u> User: All Users < <u>B</u> ack <u>N</u> ext >	インストール Cancel

図 2-1 64 ビット版 e² studio のインストールウィザード



2. ようこそページ

インストール先を変更する場合は、[変更…]をクリックして指定してください。[Next]をクリックします。

- 注1: 複数のバージョンの e² studio インストールが必要な場合も、[変更…]をクリックしてインストール先を指 定してください。
- 注2: インストール先には英文字、数字、アンダーバーのみを含むフォルダパスを指定してください。

また、Windows のアカウント名の中に英文字、数字、アンダーバー以外の文字がある場合、e² studio の動 作に問題がある可能性がありますので、Windows のアカウント名にも英文字、数字、アンダーバーだけ使 用することをお勧めします。(詳細は e² studio V2020-04 以降のダウンロードページに掲載している注意 事項をご参照ください。)





3. デバイス・ファミリー

インストールするデバイス・ファミリーを選択できます(複数選択可)。RZファミリのソフトウェアを開発するために[RZデバイス・サポート]チェックボックスを選択してください。[Next]をクリックします。



図 2-3 64 ビット版 e² studio のインストール:デバイス・ファミリ



4. 拡張機能

インストールする追加コンポーネント(言語パック、SVN および Git サポート等)を選択してください。

[Next]をクリックします。

注意: 日本語の言語パックを指定しないとメニュー等が日本語表示になりません。

 Renesas e² studio 20 Renesas e² studio 20 インストールする拡張機能 	023-04 S 023-04 を選択し	etup Setup ます。		
ようこそ			Japanese Language Support	
→ 拡張機能		P	Chinese (Simplified) Language Support	
カスタマイズ 機能			Chinese (Traditional) Language Support	
追加 ソフトウェア			Git Integration Git SCM Support	
ライセンス ショートカット			Terminals ANSI/vt102 compatible Terminal support for Serial, ssh and Te	elnet
概要				
インストール中 結果				
	Selec	t All		
<u>v202304061529</u>		User: All Use	rs < <u>B</u> ack <u>N</u> ext >	インストール Cancel

図 2-4 64 ビット版 e² studio のインストール: 拡張機能



5. コンポーネント

[デバイス・ファミリー] で選択した必要なコンポーネントは、すべて自動的に選択されます。 選択したコンポーネントを確認し、[Next]をクリックします。



図 2-5 64 ビット版 e² studio のインストール: カスタマイズ機能



6. 追加ソフトウェア

GCC ARM Embedded (GNU ARM Embedded toolchain と同様)、LibGen for GCC ARM Embedded な どの追加ソフトウェアが選択されていることを確認してください。(自動的に選択された追加ソフトウ ェアの一覧は、選択したデバイス・ファミリーによって決定します。)選択したソフトウェアのインス トーラは、 e^2 studio インストーラに呼び出されます。(詳細は、2.4 節を参照してください。)一覧に含 まれない、GCC ARM Embedded の他のバージョンを使用したい場合は、別途インストールしてくださ い(1.3 節を参照)。[Next]をクリックします。

/ストールする追加ソフト	ウェアを選択します。	
ようこそ	Renesas Toolchains & Utilities (0) GCC Toolchains & Utilities (1)	
デバイス・ファミリ 拡張機能 カスタマイズ 機能 追加 ソフトウェア	□ GNU ARM Embedded 9.3.1 2020q2 9.3.1.2020q2 GNU ARM Embedded 9.3.1 2020q2 ダウンロード・サイズ:111.2 MB □ GNU ARM Embedded 9.2.1 2019q4 9.2.1.2019q4 GNU ARM Embedded 9.2.1 2019q4 ダウンロード・サイズ:94.0 MB	
/T ビンズ 概要 インストール中 結果	 GNU ARM Embedded 6 2017q2 6.0.0.2017q3 GNU ARM Embedded 6 2017q2 ダウンロード・サイズ: 82.6 MB LibGen for GNU ARM Embedded 1.2022.09 LibGen for GNU ARM Embedded (requires GNU ARM Embedded toolchain) ダウンロード・サイズ: 51.7 MB 	
	GCC ARM A-Profile	*
	GCC ARM A-Profile (AArch64 bare-metal) 10.3 2021.07 10.3.0.2021-07 GCC ARM A-Profile (AArch64 bare-metal) 10.3 2021.07 ダウンロード・サイズ:97.8 MB	
	97.8 MB	のダウンロードが必要

図 2-6 64 ビット版 e² studio のインストール:追加ソフトウェア



7. ライセンス

ライセンス契約を読んで同意した後、[Next]をクリックします。 ライセンス契約に同意しない場合、インストールは続行できません。

enesas e² studio 2	2023-04 Setup	RENES
ようこそ	次のソフトウェア契約を読んで同意して	<ださい。
デバイス・ファミリ 拡張機能 カスタマイズ 機能 追加 ソフトウェア ライセンス ショートカット 概要 インストール中 結果	Renesas e2 studio OpenJDK License Agreement ARM DS-5 Toolchain Integration IAR Plugin Manager	License Terms and Conditions for RENESAS e2 studio This Renesas e2 studio license agreement ("Agreement") is between the entity on whose behalf you are entering into this Agreement ("Client") and Renesas Electronics Corporation, a Japanese company with its registered office at 3-2-24, Toyosu, Koto-ku, Tokyo 135-0061, Japan ("Renesas"). YOU SHOULD READ THIS AGREEMENT CAREFULLY, AS IT CONSTITUTES A BINDING CONTRACT BETWEEN CLIENT AND RENESAS. The Renesas IDE Software (defined below) is intended for commercial use by a company or corporation only and is not designed, developed or produced for any private use or purpose. If you are an individual, or you intend to install the Renesas IDE Software on behalf of an individual, or the Renesas IDE Software is expected to be used for a private purpose directly or indirectly, you should click "No" on the installer. Otherwise, by clicking the "I accept" button or other button or mechanism
	> ソフトウェア契約の条件に同意しま	designed to advanded a correspondent to the terms of an electronic conv. of this J.

図 2-7 64 ビット版 e² studio のインストール: ライセンス



8. ショートカット

スタートメニューに表示するショートカット名を選択し、[Next]をクリックします。

🗟 Renesas e² studio 20	023-04 Setup			· · · · · · · · · · · · · · · · · · ·	— 🗆 X
Renesas e² studio 20	023-04 Setup			RE	NESAS
ようこぞ デバイス・ファミリ 拡張機能 カスタマイズ 機能 追加 ソフトウェア ライセンス ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	重要なプログラムおよびファイルへの	ショートカットは、次の場所に作 enesas Electronics e2 studio	成されます:		€ デフォルトを復元(I
<u>v202304061529</u>	User: All Users	< <u>B</u> ack	<u>N</u> ext >	インストール	Cancel

図 2-8 64 ビット版 e² studio のインストール: ショートカット



9. 概要

[インストール]をクリックして、e² studio をインストールしてください。

enesas e² studio 2023	-04 Setup			REN	JESV
ようこそ デバイス・ファミリ 拡張機能 カスタマイズ 機能 追加 ソフトウェア ライセンス ショートカット ・ 概要 インストール中 結果	tyストールの準備 インストールの準備 インストールするソフトウ ・Renesas e2 stud ・Java Runtime v1 ・IAR Plugin Mana ・Renesas e2 stud ・Renesas RZ Fam ・Renesas RZ Fam ・Renesas FSP Sm ・Renesas FSP Sm ・Japanese Language ・Babel Language ・CC for Depresent	II 7 : lio v23.4.0.R20230406-1341 17.0.0 ager v1.1.0.202007251457 lio Common Components v2 lio Common Components v2 lio Tools v23.4.0.R20230406 nily Support v23.4.0.R20230406 nily Support v23.4.0.R20230406 nily Support v23.4.0.R20230406 nily Support v23.4.0.R20230406 nily Support v23.4.0.R20230406 nily Support v23.4.0.R20230406 nart Configurator Core v9.1.0 nart Configurator ARM v9.1.0 age Support v4.5.100 chain Integration v1.0.4.v202 Pack for eclipse in Japanese Pack for tools.cdt in Japanese Pack for tools.cdt in Japanese Pack for tools.cdt in Japanese	23.4.0.R20230406-13 or ARM Devices v23.4 -1341 -06-1341 ramework) v23.4.0.R2 .v20230323-1156 0.v20230319-1838 20929-0934 (91.08%) v4.22.0.v20 (85.22%) v4.22.0.v20 (85.22%) v4.22.0.v20 (91.08.v20221121-094 0.00200300 1010	41 .0.R20230406-1341 0230406-1341 211218020001 211218020001 20211218020001 3	
<u>v202304061529</u>	User: All Users	< <u>B</u> ack	<u>N</u> ext >	<u>イ</u> ンストール	Cancel

10. インストール中...

インストールの実行中は、追加ソフトウェアで選択した項目に応じて、ソフトウェアをインストールするためのダイアログボックスが開きますので、画面の指示に従って操作を行ってください。詳細は、2.4節を参照してください。



11. 結果

インストールの結果が表示されます。エラーメッセージが無いか確認してください。 [OK]をクリックすると、インストールが終了します。

🗟 Renesas e² studio 20	23-04 Setup			— 🗆 X
Renesas e² studio 20	23-04 Setup			RENESAS
ようこそ デバイス・ファミリ 拡張機能 カスタマイズ 機能 追加 ソフトウェア ライセンス ショートカット 概要 インストール中	e2 studioのインストールが完了しました。 OKをクリックして閉じてください。 □ Launch e2 studio? ☑ View Release Notes? ☑ View What's New? 役立つリンク: GCC ARM A-Profile (AArch64 bare-r	netal): C:¥Renesas¥e2	studio¥toolchains¥gcc	<u>: arm aarch64¥10 2021 07</u>
<u>v202304061529</u>	User: All Users	< <u>B</u> ack	<u>N</u> ext >	OK Cancel

図 2-10 64 ビット版 e² studio のインストール:結果



2.2 e^2 studio インストーラを用いたインストール (32 ビット版)

 e² studioインストーラをダブルクリックしてe² studioインストールウィザードページを開いてください。 [次へ(N)] をクリックします。

注1: e² studio が既に PC にインストールされた場合は図 2-11 の通りに変更、削除、インストールの選択肢が表示されます。複数のバージョンの e² studio インストールが必要な場合も、[インストール]をクリックしてインストール先を指定してください。

e ^² studio 7.8.0.R2020032	1-2326 セットアップ	RENESAS
e2 studio は既にインストールさ	ಗಿರಗುಕ್ಕತೆ	
	何を実施しますか?	
	アップグレード パージョン 7.8.0.R20200321-2326 へのアップ 場所: C:¥Renesas¥770	<i>オ</i> グレード
	修正 存在するインストールの機能に対する追加, 前 場所: C:¥Renesas¥780	11)除
	削除 存在するバージョンをアンインストール 場所: Ci¥Renesas¥780	
	インストール 異なる場所へのインストール	

図 2-11 32 ビット版 e2 studio のインストールウィザード

2. ようこそページ

[次へ(N)] をクリックします。

🔜 e² studio セットアップ			—	×
e ^² studio 7.8.0.R20200321-2326 セットアップ		REN	ESΛ	S
はうこそ	ようこそ e² studio インストール・ウィザードへ。 このウィザードではインスト	ール手順を案内します。		
○ インストール・フォルダー	Nextを押して次のページへ進みます。			
○ デバイス・ファミリー	Back を押して前のページで変更を加えます。			
○ 追加コンポーネント	Cancel を押すといつでもインストールをキャンセルできます。			
○ コンポーネント				
○ 追加ソフトウェア				
○ ライセンス				
○ ショートカット				
○ 概要				
〇 インストール中				
○ 結果				
			<u>プロキシーを</u>	<u>設定</u>
<u>v202003220126</u>	< 戻る(<u>B</u>)	次へ(<u>N</u>) > インストール	(① キャンセ	JV -

図 2-12 32 ビット版 e² studio のインストール:ようこそ

RENESAS

3. インストール・フォルダー

デフォルトのインストール先は "C:¥Renesas¥e2_studio" に設定されています。変更する場合は、イン ストールするフォルダをテキストボックスに直接入力するか [参照...] ボタンをクリックして指定してく ださい。[次へ(N)] をクリックします。

- 注1: 複数のバージョンの e² studio インストールが必要な場合も、[インストール]をクリックしてインストール 先を指定してください。
- 注2: インストール先には英文字、数字、アンダーバーのみを含むフォルダパスを指定してください。 また、Windowsのアカウント名の中に英文字、数字、アンダーバー以外の文字がある場合、e² studioの動 作に問題がある可能性がありますので、Windowsのアカウント名にも英文字、数字、アンダーバーだけ使 用することをお勧めします。(詳細は e² studio V2020-04 以降のダウンロードページに掲載している注意 事項をご参照ください。)

studio 7.8.0.R20	200321-2326 (9779)	RENESAS
● ようこそ	どのフォルダにインストールしますか?	
ニシ インストール・フォルタ	Ø- C:¥Renesas¥e2_studio	
○ デバイス・ファミリー		😣 デフォルトを復元 (R) 🗋 参照…(s)
○ 追加コンポーネント		
○ コンポーネント		
○ 追加ソフトウェア		
○ ライセンス		
○ ショートカット		
○ 概要		
〇 インストール中		
○ 結果		
<u>v202003220126</u>		< 戻る(B) 次へ(N) > インストール (I) キャンセル



4. デバイス・ファミリー

インストールするデバイス・ファミリーを選択できます(複数選択可)。RZファミリのソフトウェアを開発するために[RZデバイス・サポート]チェックボックスを選択してください。



図 2-14 デバイス・ファミリー(32 ビット版 e² studio):RZ デバイス・サポート



注意: [Support for RZ/G Devices]は、RZ/G Linux アプリケーションプロジェクトを生成、デバッグするためのオプ ションです。(e² studio 32 ビット版のみ選択可能)RZ/G デバイス上で Linux 用アプリケーションを開発す る場合は、こちらも選択してください。



図 2-15 デバイス・ファミリー(32 ビット版 e² studio): Support for RZ/G Devices



5. 追加コンポーネント

インストールする追加コンポーネント(言語パック、SVN および Git サポート、RTOS サポート)を選 択してください。

FreeRTOS、OpenRTOS などの RTOS を使用している場合は、[RTOS]を選択してください。 [次へ] をクリックします。

注意: 日本語の言語パックを指定しないとメニュー等が日本語表示になりません。

🔜 e² studio セットアップ		— 🗆 X
e ² studio 7.8.0.R202003	321-2326	RENESAS
 ● ようこそ ● インストール・フォルダー 		日本語言語パック 日本語サポート
 ● デバイス・ファミリー □ 追加コンポーネント 	• 📮	中国語(簡体字)言語パック 簡体字サポート
 ○ コンボーネント ○ 追加ソフトウェア ○ ライセンス 		中国語 (繁体字) 言語パック 繁体字サポート
○ ショートカット ○ 概要	• 🖵	コラボレーション (SVN, Git サポート)
○ インストール中 ○ 結果		RTOS FreeRTOS & OpenRTOS デバッグ・サポート
		Terminals シリアル, ssh. telnet接続の ANSI/vt102 互換ターミナル・サポート
	Select All Deselect A	.11
<u>v202003220126</u>		< 戻る(<u>B</u>) 次へ(<u>N</u>) > インストール ([) キャンセル

図 2-16 追加コンポーネント(32 ビット版 e² studio): RTOS の選択



6. コンポーネント

[デバイス・ファミリー] で選択した必要なコンポーネントは、すべて自動的に選択されます。 選択したコンポーネントを確認し、[次へ] をクリックします。



図 2-17 コンポーネント (32 ビット版 e² studio)



7. 追加ソフトウェア

GCC ARM Embedded (GNU ARM Embedded toolchain と同様)、LibGen for GCC ARM Embedded な どの追加ソフトウェアが選択されていることを確認してください。(自動的に選択された追加ソフトウ ェアの一覧は、選択したデバイス・ファミリーによって決定します。)選択したソフトウェアのインス トーラは、 e^2 studio インストーラに呼び出されます。(詳細は、2.3 節を参照してください。)一覧に含 まれない、GCC ARM Embedded の他のバージョンを使用したい場合は、別途インストールしてくださ い(1.3 節を参照)。[次へ] をクリックします。

4377	Toolchains & Utilities (3)	
インストール・フォルダー		
デバイス・ファミリー	🙀 GCC ARM Embedded	* ^
	GCC ARM Embedded 4.9 2015q3 4.9 2015q3	
這加コンハーイント	ダウンロード・サイズ: 63.7 MB	
コンポーネント		
追加ソフトウェア	GCC ARM Embedded 6 2017q2 6 2017q2	
ライセンス	ダウンロード・サイズ: 82.6 MB	
) ショートカット	LibGen for GCC ARM Embedded 1.1	
一個型	LibGen for GCC ARM Embedded (requires GCC ARM Embedded toolchain)	
	977U-F+#17:7.6MB	
インストール中…	🙀 Linaro GCC	*
19.7K	Linaro GCC 4.8-2014.04 4.8.0.201404	
	Linary GCC 4.8-2014.04	
	2770-1-91X: 030 MB	
		~

図 2-18 追加ソフトウェア (32 ビット版 e² studio)



8. ライセンス

ライセンス契約を読んで同意した後、[次へ]をクリックします。 ライセンス契約に同意しない場合、インストールは続行できません。

		•(ENES/1
● ようこそ	次のソフトウェア使用許諾を読み承諾してください	N Contraction of the second se
◉ インストール・フォルダー	WHIS StateViewer	STATEVIEWER LICENSE AGREEMENT
•) デバイス・ファミリー	Eclipse Foundation Software User Agr	ATTENTION: PLEASE READ THIS DOCUMENT CAREFULLY BEFORE
	IAR Plugin Manager	THE SOFTWARE. THE INDIVIDUAL OR ENTITY INSTALLING THIS
シュロ加コンハーイント		SOFTWARE (THE "LISER") AGREES TO BE BOUIND BY THE TERMS OF THIS
● コンポーネント		LICENSE.
◉ 追加ソフトウェア		IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, DO NOT
> ライセンス		OR USE THE SOFTWARE.
- 		This License Agreement (the 'Agreement') is entered into at the date of installation of the Software (the 'Effective Date') by
) 41 m		and between the USER and WITTENSTEIN aerospace & simulation
ノ (城安 		Itd, a corporation registered under the laws of England and trading
ン インストール中		
) 結果	└──私はソフトウェア使用許諾を承諾します	
	1	

図 2-19 ライセンス (32 ビット版 e² studio)

9. ショートカット

スタートメニューに表示するショートカット名を選択し、[次へ] をクリックします。

e ⁻ studio 7.8.0.R202003	RENESAS
 ようこそ 	次の場所に重要なプログラムやファイルにショートカットを作ります:
◉ インストール・フォルダー	マスタートメニュー: Reports Electronics altudia
◉ デバイス・ファミリー	
◉ 追加コンポーネント	
◎ コンポーネント	
◉ 追加ソフトウェア	
 ライセンス 	
🖒 ショートカット	
○ 概要	
○ インストール中	
○ 結果	
v202003220126	< 戻る(B) 次へ(N) > インストール (I) キャンセル

RENESAS

10. 概要

[インストール]をクリックして、e² studio をインストールしてください。

² studio 7.8.0.R202003	21-2326 セットアップ RENES	
● ようこそ	続けるには情報を確認し Install を押してください。	
◉ インストール・フォルダー	インストールされる必要なソフトウェア: None	^
◉ デバイス・ファミリー	インコレール きち スコン・デー・シント	
● 追加コンポーネント	1 ノストールとイレタコノハーイント: Renesas e2 studio (7.8.0.R20200321-2326) Renesas e2 studio (7.8.0.R20200321-2326)	
ションポーネント	IAR Plugin Manager (1.1.0.201906261345)	
〕 追加ソフトウェア	GCC for Renesas RZ Build Support (7.8.0.v20200313-1056) Renesas RZ Debug Support (7.8.0.v20191127-2112)	
シ ライセンス	Renesas e2 studio Common Build (6.10.0.v20200313-1019) Renesas e2 studio Common Build for GCC Toolchains (6.8.2.v20200312-1245)	
◉ ショートカット	Renesas e2 studio Memory Usage View (6.11.0.v20200224-1628) Renesas e2 studio Stack Analysis View (6.9.0.v20200305-0646)	
◇概要	Renesas e2 studio Common Debug (6.10.1.v20200312-1118)	
) インストール中	Renesas e2 studio AKM Common Debug (6.11.0.v20200312-0639) Renesas Debug Views (6.11.0.v20200312-0842)	
○結果	Renesas Smart Help (6.11.0.v20200206-0641) GCC for Renesas RZ Support Files (1.1.7.v20200311-1145)	,
		キャンクリ

図 2-21 概要(32 ビット版 e² studio)

11. インストール中...

インストールの実行中は、追加ソフトウェアで選択した項目に応じて、ソフトウェアをインストールす るためのダイアログボックスが開きますので、画面の指示に従って操作を行ってください。詳細は、2.3 節を参照してください。



12. 結果

[OK]をクリックすると、インストールが終了します。

🗟 e² studio セットアップ			-		×
e ^² studio 7.8.0.R202003	21-2326	R	ENE	SΛ	S
 。ようこそ ・インストール・フォルダー ・デバイス・ファミリー ・追加コンポーネント ・コンポーネント ・ョンポーネント ・追加ソフトウェア ・ライセンス ・ショートカット ・概要 ・インストール中 「〉結果 	e2 studio のインストールが完了しました。 OK をクリックして終了してくださ □ e2 studio を起動する ☑ リリースノートを参照する	ίι _α			
<u>v202003220126</u>	< 戻る(<u>B</u>)	次へ(<u>N</u>) >	ОК	キャンセ	!IV

図 2-22 結果 (32 ビット版 e² studio)

2.3 プラットフォームインストーラを用いたインストール

プラットフォームインストーラは、e² studio ツール、FSP パッケージ、GCC ツールチェーン、その他の インストールに必要なツールを含んでいます。以下の手順で、プラットフォームインストーラをダウンロー ドしてインストールしてください。

 (1) GitHub のルネサス製 RZ ファミリ・マイクロコントローラ用 Flexible Software Package (FSP) の ページにアクセスしてください。
 RZ/T2 用 : <u>https://github.com/renesas/rzt-fsp/releases</u>

RZ/N2 用 : https://github.com/renesas/rzn-fsp/releases

RZ/A3UL 用: https://github.com/renesas/rza-fsp/releases

以降、ここでは RZ/T2 用プラットフォームインストーラを使用してご説明します。

(2) プラットフォームインストーラ("setup_rztfsp<version>_e2s_<version>.exe")を選択し、ダウンロードの直接リンクをクリックしてください。

🖯 https://g	https://github.com/renesas/rzt-fsp/releases		£≡	Ē
	 Known issues checks Getting Started document in each product page RZ/T2M RZ/T2L MD5 Checksums 	<u>).</u>		
	 RZT_FSP_Packs_v1.2.0.zip 61333533d50cab851a93018dcfb53375 RZT_FSP_Packs_v1.2.0.exe 8a828c9c36693a963bcd7d3cb84e338 fsp_documentation_v1.2.0.zip a755b772c5f1c9b75894e3b3a7d2 setup_rztfsp_v1_2_0_e2s_v2023-01.exe 8e3de15d072c4913363ec setup_rztfsp_v1_2_0_rzsc_v2023-01.exe f44dda578759685874074 	5 b 3a0e :3917d1c 4199448a	156e6 a31a7	
	▼Assets 7			
	<pre>\$\Phisp_documentation_v1.2.0.zip</pre>	3.47 MB	Ma	ar 3
	PRZT_FSP_Packs_v1.2.0.exe	28.7 MB	Ma	ar 3
	ORZT_FSP_Packs_v1.2.0.zip	25.3 MB	Ma	ar 3
	𝔅 setup_rztfsp_v1_2_0_e2s_v2023-01.exe	1.51 GB	last mo	nth
	Setup_rztfsp_v1_2_0_rzsc_v2023-01.exe	572 MB	last mo	nth

図 2-23 インストール – FSP パッケージのダウンロード

(3) インストールファイルを実行してください。



(4) インストールするコンポーネントをカスタマイズしたい場合は、[インストール・タイプ] ページで "Custom Install" を選択し、[Next] ボタンを押してください。

初めて e² studio を使うユーザは、"Quick Install"オプションで簡単な手順を選択することをお勧めし ます。"Quick Install"オプションでは、デフォルトで e² studio、FSP、GCC ARM Embedded Toolchain をインストールします。"Quick Install"を選択した場合は、手順(6)から(8)の画面は表示されません。

Renesas RZ/T Flexible Software Package (FSP) v1.2.0 with e ² studio 2023-01 Setup	— 🗆 X
Renesas RZ/T Flexible Software Package (FSP) v1.2.0 with e ² studio 2023-01 Setup	RENESAS
インストール・タイプ	
Select Install Type:	
Quick Install Default installation of e ² studio, FSP & GCC ARM Embedded	
Custom Install Custom installation of e ² studio, FSP & GCC ARM Embedded	
<u>v202303060051</u> User: All Users < <u>B</u> ack <u>N</u> ext > <u>1</u> ンストール	Cancel
図 2-24 インストール - インストール・タイプの選択	

(5) [ようこそ] ページ

デフォルトのフォルダを使用するか、あるいは [変更…]をクリックしてフォルダを変更できます。



[Next] で次に進みます。



図 2-25 インストール - [ようこそ] ページ

e ² studio ユーザ-	ーズマニュアル	入門ガィ
(6) [拡張機能]	ページ	
必要な機能	Eを選択し、[Nex	ɪt] ボタンを押してください。
"Quick Inst	all"を選択した [」]	場合は、このページは表示されません。
🔀 Renesas RZ/T Flex	xible Software Packa	age (FSP) v1.2.0 with e ² studio 2023-01 Setup
Renesas RZ/T Flex	xible Software Pa	ackage (FSP) v1.2.0 with e ² studio 2023-01 Setup
インストールする拡張機	幾能を選択します。	
LOZZ		Innonoco I anguago Sunnort
		Japanese Language Support
● 拡張機能		Chinese (Simplified) Language Support
カスタマイズ		
1750 RE		Chinese (Traditional) Language Support
リフトウェア		
ライセンス		Git SCM Support
ショートカット		Terminals
概要		ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet
インストール中		
結果		
	Select All	
v202303060051	User: All Users	< <u>B</u> ack <u>N</u> ext > <u>イ</u> ンストール Cancel
	_	

図 2-26 インストール – 拡張機能

(7) [カスタマイズ機能] ページ

インストールするコンポーネントを選択し、[Next] ボタンで次に進みます。 "Quick Install"を選択した場合は、このページは表示されません。



Renesas RZ/T Flexib	le Software Package (FSP) v1.2.0 with e ² studio 2023-01 Setup – – – ×
ようこそ 拡張機能	インストールするコンポーネントを選択します。
→ カスタマイズ 機能	OpenJDK & OpenJFX ✓ Renesas FSP Smart Configurator Core (9.0.0.v20230106-1254) Common components for Renesas FSP Smart Configurator
追加 ソフトウェア ライヤンス	 ✓ GCC for Renesas RZ Build Support (23.1.0.v20221212-1830) GCC for Renesas RZ Build Support ✓ Renesas RZ Common (23.1.0.v20221115-0828)
ショートカット 概要	Renesas RZ Common Renesas RZ Debug Support (23.1.0.v20221115-0828) Renesas RZ Debug Support
インストール中 結果	すべて選択(S) すべて解除(D)
<u>v202303060051</u>	Size of install: 767.9 MB User: All Users < <u>B</u> ack <u>N</u> ext > <u>インストール</u> Cancel

図 2-27 インストール - カスタマイズ機能

(8) [追加ソフトウェア] ページ

インストールするツールチェインを選択し、[Next] ボタンを押してください。"Quick Install"を選択 した場合は、このページは表示されません。





(9) ソフトウェア契約に同意いただけましたらチェックを入れてください。[Next] で次に進みます。





図 2-29 インストール - ソフトウェア契約

(10) [ショートカット] ページ

スタートメニューに登録するショートカット名を入力します。[Next] で次に進みます。

注: すでに別のバージョンの e² studio がインストールされている場合は、それと区別が付くような名前に書き換えることをお勧めします。




インストール – ショートカット 図 2-30

(11) [概要] ページの内容を確認してください。[インストール] ボタンを押すとインストールが始まりま す。





図 2-31 インストール – 概要

(12) [OK] ボタンを押してインストールを終了します。



🔜 Renesas RZ/T Flex	ible Software Package (FSP) v1.2.0 with e ² studio 2023-01 Setup $\hfill\square$
Renesas RZ/T Flex	tible Software Package (FSP) v1.2.0 with e ² studio 2023-01 Setup
ようこそ	Renesas RZ/T Flexible Software Package (FSP) v1.2.0 with e ² studio 2023-01のインストールが完了しました。
拡張機能	OKをクリックして閉じてください。
カスタマイズ 機能 追加 ソフトウェア ライセンス	 □ Launch e2 studio? ✓ View Release Notes? ✓ View What's New? ✓ View Renesas FSP User Manual?
ショートカット	
概要 インストール中 → 結果	後立つリンク: Renesas FSP: C:¥Renesas¥rzt¥e2studio v2023-01 fsp v1.2.0 Renesas FSP User Manual: C:¥Renesas¥rzt¥e2studio v2023-01 fsp v1.2.0 GCC ARM Embedded: C:¥Renesas¥rzt¥e2studio v2023-01 fsp v1.2.0¥toolchains¥gcc arm¥9 2019q4 GCC ARM Embedded: C:¥Renesas¥rzt¥e2studio v2023-01 fsp v1.2.0¥toolchains¥gcc arm¥9 2020q2 GCC ARM A-Profile (AArch64 bare-metal): C:¥Renesas¥rzt¥e2studio v2023-01 fsp v1.2.0¥toolchains¥ gcc arm aarch64¥10.3-2021.07
<u>v202303060051</u>	User: All Users < Back Next > OK Cancel

図 2-32 インストール - インストールの完了



2.4 追加ソフトウェアのインストール

e² studio インストーラは、[追加ソフトウェア]で選択された GNU Compilers、Libgen Update for GNU ARM Embedded Toolchains などの他のインストーラを呼び出します。

2.4.1 GNU ARM Embedded toolchain

インストール中、e² studio インストーラの[追加ソフトウェア]で GNU ARM Embedded Toolchain が選択されている場合、このインストールウィザードが表示されます。

- 1. [次へ]をクリックします。
- 2. ライセンス契約を読んだ後、[同意する]をクリックします。
- 3. [インストール先フォルダ]にパスを指定してください。デフォルトのフォルダを推奨します。
- 4. インストールが終わると、Windows環境変数のパスや、レジストリ情報を追加するオプションを選択することができます。



図 2-33 GNU ARM Embedded Toolchain のインストール



2.4.2 Libgen Update for GNU ARM Embedded Toolchains

インストール中、e² studio インストールの[追加ソフトウェア]で Libgen Update for GNU ARM Embedded Toolchain が選択されている場合、このインストールウィザードが表示されます。GNU ARM Embedded Toolchain を更新しない場合は、[Close] (—)をクリックして、Libgen Update for GNU ARM Embedded Toolchains を終了します。

- 1. [Click here to select your GNU ARM Embedded Toolchain installation folder]をクリックします。
- 2. 更新する[GNU ARM Embedded Toolchain]のフォルダを選択します。

	Libgen Update for GNU AR Version 1.2022.09	M Embedded Toolchai	ins		
	Copyright c 2017-2022 CyberT	hor Studios Ltd. All Rights	Reserved.		
	Click here to visit our support p	latform at llvm-gcc-renes	as.com		
	Click here to visit our website		- A	EYB	ERTHO
			~		STUDIO
This Libgen upda recommend th	te installer will be making changes to at you only apply this update to GNU incc	your existing GNU ARM Embe ARM Embedded release 5-20 ompatibilities.	edded Toolcha 016-q1-updat	ain installati e or later, t	on(s). We o avoid
	Click here to download the GN	U ARM Embedded Toolchain i	installer(s)		
Г. т .				.: 1	
[100	Ichain folder not selected. Plea	se select the toolchain installation	folder to co	ontinue.]	
		CM Embedded Toolchain Installatio	on roidei		
	Install Libgen Update for	GNU ARM Embedded T	oolchains		
Autor	natically <u>I</u> nstall Libgen Update for	all detected GNU ARM Em	nbedded Too	lchains	
Autor	natically Install Libgen Update for	all detected GNU ARM Em	nbedded Too	olchains	
Autor	natically Install Libgen Update for	all detected GNU ARM Em	nbedded Too	olchains	
Autor	natically Install Libgen Update for	all detected GNU ARM Em	nbedded Too	olchains	
Autor	Natically Install Libgen Update for	all detected GNU ARM Em	nbedded Too	olchains —	
Autor	IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09	all detected GNU ARM Em M Embedded Toolchair フォルダーの参照	nbedded Too	olchains 	
Autor	IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照	nbedded Too		
Autor ibgen Update for GN	IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber Click here to visit our support of	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照 Current folder:	nbedded Too	olchains X	
Autor ibgen Update for GN	IU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber Click here to visit our support p Click here to visit our website	all detected GNU ARM Em M Embedded Toolchair フォルダーの参照 Current folder: CY=¥GNU Arm Embedded Toolcha	nbedded Too ns	olchains — — — Hate	
Autor ibgen Update for GN	NU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber <u>Click here to visit our support p</u> <u>Click here to visit our website</u>	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照 Current folder: Ct*GNU Arm Embedded Toolcha	nbedded Too ns	olchains — — date	R THOR STUDIOS
Autor ibgen Update for GN	NU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber Click here to visit our support p Click here to visit our website	all detected GNU ARM Em M Embedded Toolchain 7オルダーの参照 Current folder: C:¥¥GNU Arm Embedded Toolcha	nbedded Too ns	date	THOR STUDIOS
Autor ibgen Update for GN	IU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber Click here to visit our support p Click here to visit our support p Click here to visit our website	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照 Current folder: C:¥¥GNU Arm Embedded Toolcha Canon Common Files	nbedded Too ns	date	THOR STUDIOS (s). We
Autor ibgen Update for GN	IU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber <u>Click here to visit our support p</u> <u>Click here to visit our website</u> ate installer will be making changes to hat you only apply this update to GNU	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照 Current folder: C:¥¥GNU Arm Embedded Toolcha Canon Common Files ControlCenter	nbedded Too ns	date	C X THOR STUDIOS (s). We avoid
Autor ibgen Update for GN Official States This Libgen update recommend the	IU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber <u>Click here to visit our support p</u> <u>Click here to visit our website</u> ate installer will be making changes to hat you only apply this update to GNL inc	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照 Current folder: C:¥¥GNU Arm Embedded Toolcha Canon Common Files ControlCenter ControlCenter Control Center	nbedded Too ns	date	C X THOR STUDIOS (s). We avoid
Autor ibgen Update for GN	IU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber <u>Click here to visit our support p</u> <u>Click here to visit our website</u> ate installer will be making changes to hat you only apply this update to GNL inc <u>Click here to download the G</u>	all detected GNU ARM Em M Embedded Toolchain 7オルダーの参照 Current folder: C:¥¥GNU Arm Embedded Toolcha Canon Canon Common Files ControlCenter ControlCenter GNU Arm Embed	nbedded Too ns sin¥9 2020-q2-upo	date	C X THOR STUDIOS (s). We avoid
Autor ibgen Update for GN	IU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber Click here to visit our support p Click here to visit our support p Click here to visit our website ate installer will be making changes to hat you only apply this update to GNU inc Click here to download the G	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照 Current folder: C:¥¥GNU Arm Embedded Toolcha Canon Canon Common Files ControlCenter GNU Arm Embe dotnet FrontLine GNU Arm Embe 2 3 1 9 2020-r2-11	nbedded Too ns iin¥9 2020-q2-upo edded Toolchain	date	CS). We avoid
Autor ibgen Update for GN This Libgen upda recommend th	IU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber Click here to visit our support p Click here to visit our support p Click here to visit our website ate installer will be making changes to hat you only apply this update to GNL inc Click here to download the GP Olchain folder not selected. Plea	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照 Current folder: C:¥¥GNU Arm Embedded Toolcha Current folder: C:¥¥GNU Arm Embedded Toolcha Control Center Control Center Control Center GNU Arm Embe dotnet FrontLine GNU Arm Embe 2 > 9 2020-q2-u S Google	nbedded Too ns ain¥9 2020-q2-upo edded Toolchain update	date	C THOR STUDIOS (s). We avoid
Autor ibgen Update for GN This Libgen upda recommend th [Too	IU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber Click here to visit our support p Click here to visit our support p Click here to visit our website ate installer will be making changes to hat you only apply this update to GNU inc Click here to download the GP olchain folder not selected. Plea Click here to select your GNU A	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照 Current folder: C:¥¥GNU Arm Embedded Toolcha Canon Canon Common Files ControlCenter GNU Arm Embed dotnet FrontLine GNU Arm Embed 2 > 9 2020-q2-u S Google HTML Help Wo	nbedded Too ns iin¥9 2020-q2-upo edded Toolchain update	date	(s). We avoid
Autor ibgen Update for GN This Libgen upda recommend th [Too	IU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber Click here to visit our support p Click here to visit our support p Click here to visit our website ate installer will be making changes to hat you only apply this update to GNL inc Click here to download the GP olchain folder not selected. Plea Click here to select your GNU A	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照 Current folder: C:¥¥GNU Arm Embedded Toolcha C:¥¥GNU Arm Embedded Toolcha Canon Canon Common Files ControlCenter ControlCenter GNU Arm Embe dotnet FrontLine GNU Arm Embe 2 9 2020-q2-u Google HTML Help Wol S IIS	nbedded Tool ns iin¥9 2020-q2-upo edded Toolchain update rkshop	date	(s). We avoid
Autor ibgen Update for GN This Libgen upda recommend th [Too	NU ARM Embedded Toolchains ILibgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyber Click here to visit our support p Click here to visit our website ate installer will be making changes to hat you only apply this update to GNU inc Click here to download the GP Olchain folder not selected. Plea Click here to select your GNU A Install Libgen Update fo	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照 Current folder: C4¥GNU Arm Embedded Toolcha Canon Canon Common Files ControlCenter ControlCenter ControlCenter GNU Arm Embe ControlCenter GNU Arm E	nbedded Too ns ain¥9 2020-q2-upd edded Toolchain update rkshop	date	(s). We avoid
Autor ibgen Update for GN This Libgen upda recommend th [Too	IU ARM Embedded Toolchains IU ARM Embedded Toolchains Libgen Update for GNU AR Version 1.2022.09 Copyright c 2017-2022 Cyberi Click here to visit our support p Click here to visit our support p Click here to visit our website ate installer will be making changes to hat you only apply this update to GNU inc Click here to download the GP Olchain folder not selected. Plea Click here to select your GNU A Install Libgen Update fo	all detected GNU ARM Em M Embedded Toolchain フォルダーの参照 Current folder: C:¥¥GNU Arm Embedded Toolcha Current folder: C:¥¥GNU Arm Embedded Toolcha Canon Canon Common Files ControlCenter ControlCenter ControlCenter GNU Arm Embed ControlLine GNU Arm Embed ControlLine Current folder: Current folder:	nbedded Too ns iin¥9 2020-q2-upo edded Toolchain ipdate irkshop	date	(s). We avoid

図 2-34 Libgen Update for GNU ARM Embedded: ツールチェーンフォルダの選択



3. [Install Libgen Update for GNU ARM Embedded Toolchains]をクリックします。



図 2-35 Libgen Update for GNU ARM Embedded: ツールチェーンの更新

他のツールチェーンを更新する場合は、[Libgen Update for GNU ARM Embedded]をダウンロードして実行 してください(1.4 節を参照)。

2.5 e^2 studio $\mathcal{O}\mathcal{P}\mathcal{V}\mathcal{A}\mathcal{V}\mathcal{A}\mathcal{H}\mathcal{P}\mathcal{V}$

e² studio のアンインストールは、Windows OS での通常のプログラムアンインストール手順で行えます。 1. [スタート] → [設定] → [アプリ(アプリと機能)]を選択します。

- 2. アプリのリストから"Renesas e² studio" を選択し、[アンインストール] ボタンをクリックします。
- [アンインストール] ダイアログボックスの [アンインストール] ボタンをクリックして削除を確認してく ださい。

アンインストールの最後に、e² studio はインストール先から削除され、ショートカットメニューも削除さ れます。

2.6 ツールチェーンへの e² studio の登録

インターネット接続時に e² studio をインストールする際、GCC ARM Embedded などのツールチェーンを e² studio インストーラで自動的にインストールすることができますが、インターネット接続のない環境で e² studio をインストールする際、ウェブサイトからダウンロードしたインストーラファイルを使用してツール チェーンを別途インストールし、e² studio に登録する必要があります。(RZ 用ツールチェーンのダウンロー ドサイトについては、1.3 節を参照してください。)

e² studio は、起動時に、インストールしたツールチェーンをスキャンします。ツールチェーンのインスト ール後に e² studio を実行する場合、ツールチェーンを選択するための Toolchain Integration ウィンドウが表 示されます。

ツールチェーンの登録	
ツールチェーン・タイプ	インストール・パス
V GNU ARM Embedded	
GNU ARM Embedded - 4.9.3.20150529	C:\Program Files (x86)\GNU Tools ARM Embedded\4.9 2015q3\
GNU ARM Embedded - 5.4.1.20160919	C:\Program Files (x86)\GNU Tools ARM Embedded\5.4 2016q3\
GNU ARM Embedded - 6.3.1.20170620	C:\Program Files (x86)\GNU Tools ARM Embedded\6 2017-q2-update\
GNU ARM Embedded - 7.2.1.20170904	C:\Program Files (x86)\GNU Tools ARM Embedded\7 2017-q4-major\
GNU ARM Embedded - 7.3.1.20180622	C:\Program Files (x86)\GNU Tools ARM Embedded\7 2018-q2-update\
GNU ARM Embedded - 8.3.1.20190703	C:\Program Files (x86)\GNU Tools ARM Embedded\8 2019-q3-update\
GNU ARM Embedded - 9.2.1.20191025	C:\Program Files (x86)\GNU Tools ARM Embedded\9 2019-q4-major\
✓ ✓ KPIT GNUARM-NONE-EABI Toolchain	
KPIT GNUARM-NONE-EABI Toolchain - v14.02	2 C:\Program Files (x86)\KPIT\GNUARM-NONEv14.02-EABI\arm-none-eabi\arm-none-eabi\
KPIT GNUARM-NONE-EABI Toolchain - v16.01	C:\Program Files (x86)\KPIT\GNUARM-NONEv16.01-EABI\arm-none-eabi\arm-none-eabi\
Select all Deselect all マ 起動時に "ツールチェーンの登録'を有効にする	

図 2-36 ツールチェーンの登録

インストール済みのツールチェーンを確認するには、 [ヘルプ]メニューから[Renesas ツールチェーンの追加]をクリックして、[Renesas ツールチェーン管理] (下図)を開きます。e² studio に登録したいツールチェーンを確認してください。

入門ガイド



図 2-37 Renesas ツールチェーンの追加



使用したいツールチェーンがリストにない場合は、[追加] ボタンをクリックしてインストールされている 場所を指定してください。

	Renesas ツールチェーン管理		<p 4<="" th="" ▼=""><th>$\rightarrow -$</th></p>	$\rightarrow -$
✓ Renesas	✓ 記動時にインストールされたツールチェーンを探す			
Breakpoints		インストール・パス		
> FSP	✓ ✓ GNU ARM Embedded			
Module Download	9.3.1.20200408	C:¥Program Files (x86)¥GNU Arm En	nbedded Toolchain¥9 2020-q2-update	¥
My Renesas	KPIT GNUARM-NONE-EABI Toolchain			
Reality AI Authenticat	xPack GNU ARM Embedded	N		
Renesas QE	GCC ARM A-Profile (AArch64 bare-meta	I) C:¥Renesas¥e2 studio¥toolchains¥o	icc arm aarch64¥10 2021 07¥occ-arm-	10 3-2
Smart Demo	V 10.5.1.20210021	e.thenesuster_statiottoolenanisty		10.5 1
> Smart Manual				
Support Folders				
Tracealyzer		Download スキャン	追加 除去	
< >	<			
? 📐 🗹 🔘			適用して閉じる キャンセノ	V
				-
📴 新規ツ	ールチェーンの追加		×	
登録され	ていない新規ツールチェーンの統合			
[GNU] 例	<pre><toolchain_path>¥GNURXv12.02-ELF¥rx-elf</toolchain_path></pre>			
[Renesas	。] 例 <toolchain_path>¥RX¥1_2_1</toolchain_path>			
ロケーション:			参照	
	<u>.</u>			
٢				
?		OK	キャンセル	
?		OK	キャンセル	
 アオルダーの選択 		OK	<u>キャンセル</u> ×	
 ⑦ ③ 7オルダーの選択 	ر م Decomp Files (دول) کے (۱۹۹۲ میں Embedded ک	OK	<u>キャンセル</u> ×	
 ⑦ ⑦ 7オルダーの選択 ← → ヾ ↑ 	« Program Files (x86) > GNU Tools Arm Embedded >	ОК 9 2019-q4-major v 0	キャンセル × 9 2019-q4-majorの検索	
 ⑦ 0 0<td>■ « Program Files (x86) → GNU Tools Arm Embedded → ハフォルダー</td><td>ОК 9 2019-q4-major v в</td><td>キャンセル > 9 2019-q4-majorの検索</td><td></td>	■ « Program Files (x86) → GNU Tools Arm Embedded → ハフォルダー	ОК 9 2019-q4-major v в	キャンセル > 9 2019-q4-majorの検索	
② フォルダーの選択 ← → × ↑ 整理 × 新しい シ PC	▶ « Program Files (x86) > GNU Tools Arm Embedded > パフォルダー へ 名前 へ	OK 9 2019-q4-major くび 更新日時 種類	キャンセル タ 2019-q4-majorの検索 単正 ・ ② サイズ	
 ⑦ ⑦ フォルダーの選択 ← → ◆ ↑ 整理 ● 新しい ● PC ③ 3D オブジェ: 	■ « Program Files (x86) > GNU Tools Arm Embedded > ルフォルダー へ 名前 クト	OK 9 2019-q4-major v じ 更新日時 種類 2023/05/17 10:38 771	キャンセル × 9 2019-q4-majorの検索 単1ズ レフオルダー	
 ⑦ Ø ⑨ ⑦ ∅ ⑨ ∅ ∅<td>■ « Program Files (x86) » GNU Tools Arm Embedded » ハフォルダー クト 名前 クト し は し は し し</td><td>OK 9 2019-q4-major くび 更新日時 種類 2023/05/17 10:38 ファイ1 2023/05/17 10:38 ファイ1 2023/05/17 10:38 ファイ1</td><td>キャンセル ♀ 9 2019-q4-majorの検索 Ⅲ ● ② サイズ ↓ フオルダー ↓ フオルダー ↓ フオルダー</td><td></td>	■ « Program Files (x86) » GNU Tools Arm Embedded » ハフォルダー クト 名前 クト し は し は し し	OK 9 2019-q4-major くび 更新日時 種類 2023/05/17 10:38 ファイ1 2023/05/17 10:38 ファイ1 2023/05/17 10:38 ファイ1	キャンセル ♀ 9 2019-q4-majorの検索 Ⅲ ● ② サイズ ↓ フオルダー ↓ フオルダー ↓ フオルダー	
 ⑦ 0 0<td>■ « Program Files (x86) > GNU Tools Arm Embedded > ハフオルダー クト 2ト 2ト 2ト 2ト 10 10 10 10 10 10 10 10 10 10</td><td>OK 9 2019-q4-major く ひ 2 東新日時 種類 2023/05/17 10:38 77f1 2023/05/17 10:38 77f1 2023/05/17 10:38 77f1 2023/05/17 10:38 77f1</td><td>キャンセル 9 2019-q4-majorの検索 単1ズ レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー</td><td></td>	■ « Program Files (x86) > GNU Tools Arm Embedded > ハフオルダー クト 2ト 2ト 2ト 2ト 10 10 10 10 10 10 10 10 10 10	OK 9 2019-q4-major く ひ 2 東新日時 種類 2023/05/17 10:38 77f1 2023/05/17 10:38 77f1 2023/05/17 10:38 77f1 2023/05/17 10:38 77f1	キャンセル 9 2019-q4-majorの検索 単1ズ レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー	
 ⑦ 0 0<td>■ « Program Files (x86) » GNU Tools Arm Embedded » ルフオルダー クト 名前 ・ こ 日 ・ 日 ・ 日 ・ の ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・</td><td>OK 9 2019-q4-major く ひ ノ 更新日時 種類 2023/05/17 10:38 ファイ) 2023/05/17 10:38 ファイ) 2023/05/17 10:38 ファイ) 2023/05/17 10:38 ファイ) 2023/05/17 10:38 ファイ)</td><td>キャンセル × ○ 9 2019-q4-majorの検索 Ⅲ ・ ② サイズ レフオルダー レフオルダー レフオルダー レフオルダー</td><td></td>	■ « Program Files (x86) » GNU Tools Arm Embedded » ルフオルダー クト 名前 ・ こ 日 ・ 日 ・ 日 ・ の ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	OK 9 2019-q4-major く ひ ノ 更新日時 種類 2023/05/17 10:38 ファイ) 2023/05/17 10:38 ファイ) 2023/05/17 10:38 ファイ) 2023/05/17 10:38 ファイ) 2023/05/17 10:38 ファイ)	キャンセル × ○ 9 2019-q4-majorの検索 Ⅲ ・ ② サイズ レフオルダー レフオルダー レフオルダー レフオルダー	
 ⑦ 0 0<td>■ « Program Files (x86) » GNU Tools Arm Embedded » ルフオルダー ~ 名前 ~ 名前 ~ 日前 ■ arm-none-eabi ■ bin ■ lib ■ share</td><td>9 2019-q4-major く む ノ 9 2023/05/17 10:38 7941 2023/05/17 10:38 7941 2023/05/17 10:38 7941 2023/05/17 10:38 7941 2023/05/17 10:39 7741 2023/05/17 10:39 7741</td><td>キャンセル × 9 2019-q4-majorの検索 単 ド ・ ク 9 2019-q4-majorの検索 レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー</td><td></td>	■ « Program Files (x86) » GNU Tools Arm Embedded » ルフオルダー ~ 名前 ~ 名前 ~ 日前 ■ arm-none-eabi ■ bin ■ lib ■ share	9 2019-q4-major く む ノ 9 2023/05/17 10:38 7941 2023/05/17 10:38 7941 2023/05/17 10:38 7941 2023/05/17 10:38 7941 2023/05/17 10:39 7741 2023/05/17 10:39 7741	キャンセル × 9 2019-q4-majorの検索 単 ド ・ ク 9 2019-q4-majorの検索 レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー レフオルダー	
 ⑦ 0 0<td>■ « Program Files (x86) > GNU Tools Arm Embedded > ハフオルダー 2ト 名前 ■ arm-none-eabi ■ bin ■ lib ■ share</td><td>9 2019-q4-major く ひ 人 9 2019-q4-major く ひ 人 夏新日時 運新日時 種類 2023/05/17 10:38 7741 2023/05/17 10:39 7741 2023/05/17 10:39 7741</td><td>キャンセル</td><td></td>	■ « Program Files (x86) > GNU Tools Arm Embedded > ハフオルダー 2ト 名前 ■ arm-none-eabi ■ bin ■ lib ■ share	9 2019-q4-major く ひ 人 9 2019-q4-major く ひ 人 夏新日時 運新日時 種類 2023/05/17 10:38 7741 2023/05/17 10:39 7741 2023/05/17 10:39 7741	キャンセル	
 ⑦ ⑦ ⑦ ⑦ ⑦ ※ ※ ※ ※ <li< td=""><td>● « Program Files (x86) > GNU Tools Arm Embedded > ハフォルダー 2ト 名前 ● arm-none-eabi ● bin ■ lib ■ share 7メルダー: 9 2019-q4-major</td><td>9 2019-q4-major V び ノ 更新日時 種類 2023/05/17 10:38 ファイ1 2023/05/17 10:38 ファイ1 2023/05/17 10:38 ファイ1 2023/05/17 10:38 ファイ1 2023/05/17 10:39 ファイ1</td><td>キャンセル × ○ 9 2019-q4-majorの検索 Ⅲ ・ ② サイズ レフオルダー レフオルダー レフオルダー メフオルダー</td><td></td></li<>	● « Program Files (x86) > GNU Tools Arm Embedded > ハフォルダー 2ト 名前 ● arm-none-eabi ● bin ■ lib ■ share 7メルダー: 9 2019-q4-major	9 2019-q4-major V び ノ 更新日時 種類 2023/05/17 10:38 ファイ1 2023/05/17 10:38 ファイ1 2023/05/17 10:38 ファイ1 2023/05/17 10:38 ファイ1 2023/05/17 10:39 ファイ1	キャンセル × ○ 9 2019-q4-majorの検索 Ⅲ ・ ② サイズ レフオルダー レフオルダー レフオルダー メフオルダー	

図 2-36 ツールチェーン管理:新規ツールチェーンの登録

RENESAS

e² studio ユーザーズマニュアル

3. プロジェクトのインポートおよび作成

この章では、既存プロジェクト(ソフトウェアパッケージのサンプルコードなど)をインポートする方 法、および、新規プロジェクトを作成する方法について説明します。

注意 : e² studio インストールフォルダ名、プロジェクト名とそのフォルダ、ソースファイル名に英文字、数字、ア ンダーバーのみを使用してください。

3.1 既存プロジェクトのインポート

ルネサスでは、RZ デバイス開発用のソフトウェアパッケージとして、便利なサンプルコードをサポートしています。このサンプルコードは、e² studio に簡単にインポートすることができます。また、プロジェクト をインポートすることにより、古いプロジェクトも使用することができます。

- 1. ソフトウェアパッケージをダウンロードします。
- [プロジェクト・エクスプローラー]を右クリックするか、[ファイル]メニューから [インポート]をクリ ックして、[インポート] ダイアログボックスを開きます。
- ツールチェーンへのe² studioの登録



図 3-1 インポートメニュー



- 3. [一般] から [既存プロジェクトをワークスペースへ] を選択し、[次へ] をクリックします。
- プロジェクトのフォルダ、あるいはアーカイブ・ファイルを選択して [参照] をクリックし、ワークスペースでプロジェクトのフォルダを選択します。

3 インポート	■ インポート –	
選択 アーカイブ・ファイルまたはディレクトリーから新規プロジェク	プロジェクトのインボート 既存の Eclipse プロジェクトを検索するディレクトリーを選択します。	
インポート・ウィザードの選択(<u>S</u>): フィルタ入力	○ ルート・ディレクトリーの選択①: C:\Users\duyph\Downloads\Workspace\Tutoria > ● アーカイブ・ファイルの選択(<u>A</u>): C:\Users\duyph\Downloads\an_r01an2511eg02 >	参照(<u>R</u>) 参照(<u>R</u>)
 ◇ 一般 ○ CMSIS Pack ◇ GNUARM-NONE/RZ(DS-5) project convers ◇ Renease & Import Existing C/C++ Project in ◇ Renesas CA78KOR (CS+) プロジェクト ◇ Renesas CC-RX/CC-RL (CS+) プロジェクト ◇ Renesas GitHub Amazon FreeRTOS プロジェクト ◇ アーカイブ・ファイル ◇ ファイル・システム ◇ フォルダーまたはアーカイブ由来のプロジェクト ◇ 設定 > ◇ C/C++ 	プロジェクト(P):	すべて選択(<u>s</u>)
⑦ < 戻る(B) 次へ	⑦ < 戻る(B) 次へ(N) > 終了(E)	キャンセル

図 3-2 プロジェクトファイルに含むフォルダあるいはアーカイブ・ファイルの選択

5. 古いe² studio (V6.0以前) で生成したプロジェクトは、更新する必要はありません。
 'プロジェクトの更新が必要です'のポップアップメッセージをクリックします。(次ページへ続きます。)

📴 GSG - e² studio	- 🗆 X
ファイル(E) 編集(E) ソース(<u>S</u>) リン	ファクタリング(T) ナビゲート(<u>N)</u> 検索(<u>A)</u> プロジェクト(<u>P</u>) Renesas <u>V</u> iews 実行(<u>R</u>) ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>)
🔦 💠 🔳 🎋 デバッグ(B)	✓ I RZ_A1H_Tutorial_RSK R ×
:::: ▼ ::: ▼ ::: ▼ ::: > // ▼	▼ □ □ □ □ ↓ ↓ ▼ □ ▼ □ ▼ □ ▼ □ ▼ □ ▼ □ ▼
Ъ プロジェクト・エクスプ 🛛 🗖 🗖	
E S V 8 > E RZ_A1H_Tutorial_RSK	
	②問題 ■コンソ ※ ●スマー ■メモリ >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
	RA クリック
	プロジェクトの更新が必要です ワークスペースにあるプロジェクトはビルドする前に更新する必要があります。 クリックしてこれらのプロジェクトを更新してください。
	—————————————————————————————————————



RENESAS

入門ガイド

プロジェクトの右クリックメニューで、[古い e2 studio プロジェクトを更新...]を実行します。



図 3-4 プロジェクトの自動更新

起動構成を変換する必要があります。(図 3-5)。

8		_		\times	
古い e2 studio プロジェク 更新するプロジェクトの選択	7トの更新 ?				
RZ_A1H_Tutorial_RSK					
0	終了(E)		キャンセル	,	

図 3-5 [変換が必要です] メッセージ



3.2 新規プロジェクトの作成

新規プロジェクトを作成する場合、FSP パッケージ有無、またデバイスグループごとにプロジェクト作成 方法が異なります。

それぞれの場合について、プロジェクトを作成する方法を説明します。

3.2.1 RZ/A2M のプロジェクトの作成(FSP パッケージ無し)

RZ/A2M Evaluation Board Kit を使用する場合の例を説明します。RZ/A2M で作成されるプロジェクトのタ イプは、"Loader Project"または"Application Project"を選択することが可能です。RZ/A2M では、この2つの 両方のプロジェクトを作成し、ビルド、デバッグ(ダウンロード)する必要があります。

"Loader Project"はローダプログラム作成用のプロジェクトであり、"Application Project"はユーザアプリケ ーションプログラム作成用のプロジェクトです。"Loader Project"は、RZ/A2M Evaluation Board Kit に搭載さ れているシリアルフラッシュメモリに高速にアクセスできるように設定した後、"Application Project"に分岐 します。(ローダプログラムがダウンロードされていない場合は、アプリケーションプログラムを実行する ことができません。)

お客様のシステムで使用する場合には、ご使用のシリアルフラッシュメモリに合わせて、ローダプログラ ムの内容を変更してください。

Windows の [スタート] メニューから e² studio を起動し、ワークスペースディレクトリを指定します。こ のワークスペースに作成したプロジェクトが追加されます。

新規にプロジェクトを作成するには、以下の手順を実行してください。

[ファイル] → [新規] → [C/C++ Project] の順にクリックすると、新規プロジェクト作成ウィザードが起動します(下図)。

דר	<mark>イル(F)</mark> 編集(E) ソース(S) リファクタリング(T)	ナビゲート(N)	検索(A)	プロジェクト(P)	Renesas Vie	ws 実行(R)	ウィンドウ(W)	ヘルプ(H
	新規(N)	Alt+シフト+N	1 >	Renesas C/C++	Project			>
	ファイルを開く(.)		c ♣	Makefile Project	t with Existin	g Code		
۵,	ファイル・システムからプロジェクトを開く		C	C/C++ Project		C または C+・	+ プロジェクトを	新規作成
	最近のファイル		> 🗾	ブロジェクト(R)…				
		Ctul M		CIC++ プロミジェノ	カトに変換 にり	ヘート ネーチャ	_た;白カn)	

図 3-6 新規プロジェクト作成ウィザード



2. 新規プロジェクトのテンプレートを選択します。[次へ(N)>] をクリックすると次の画面に進みます。

 新規 C/C++ プロ 新規 C/C++ プロ 	ジェクト – ロ X ジェクトのテンプレート
All CMake Make	GNU for Renesas RZ C/C++ Executable Project A C/C++ Executable Project for Renesas RZ using the GNU ARM Embedded toolchain.
Renesas Debug	GNU for Renesas RZ C/C++ Library Project A C/C++ Library Project for Renesas RZ using the GNU ARM Embedded toolchain. Renesas RZ/A C/C++ FSP Project Create an executable or static library C/C++ FSP project for Renesas RZ/A.
	Renesas RZ/G C/C++ FSP Project Create an executable or static library C/C++ FSP proiect for Renesas RZ/G.
?	< 戻る(B) 次へ(N) > 終了(E) キャンセル

図 3-7 プロジェクトテンプレートの選択

3. 最初にLoaderプロジェクトを作成します。プロジェクト名"Loader"を入力し、[次へ(N)>] で進めます。

lew GCC for Ren	nesas RZ Executable Project	-
New GCC for Rene	sas RZ Executable Project	
プロジェクト名(巴) し	oader	
ロケーション(1)	C·¥I (sers¥2022-034-user¥e2 studio¥GSG¥I oader	参昭(R)
	Create Directory for Project	<i>≥ 1</i> 177 (<u>11</u>)····
ファイル・システムを選		
ワーキング・セット		
ワーキング・セット	~にプロジェクトを追加(I)	新規(<u>W</u>)
ワーキング・セット(0)):	✓ 選択(<u>E</u>)

図 3-8 プロジェクト名の指定

-

以下の設定を選択します。

- 言語:"C"
- ツールチェーン: "GNU ARM Embedded"
- ツールチェーン・バージョン: "9.3.1.20200408"
- ターゲット・デバイス: "R7S921053"
- プロジェクトタイプ: "Loader Project"

[Hardware Debug 構成を生成] で、"J-Link ARM"を選択し、[次へ(N)>] で進めます。

New GCC for Rene	sas RZ Executable Proi	ect		\rightarrow
Full toolchain sup selected one.	oort is only available in GCC	ARM Embedded versio	ons lower than the	
Toolchain Settings				
言語:	● C ○ C++			
ツールチェーン:	GNU ARM Embedded	~		
ツールチェーン・バージョン	ע: 9.3.1.20200408	~		
		<u>ツールチェーンの管理</u>		
RTOS:	None	~		
RTOS Version:		~		
Device Settings			Configurations	
Target Board:	ustom	~	✓ Hardware Debu	g 構成を生成
			J-Link ARM	\sim
ターゲット・デバイス: R	7S921056		Rolosco 構成を生	- Б Ѷ
		<u>デバイスのアンロック</u>	Telease 147, 2 -	- 126
エンディアン: L	ttle	~		
プロジェクト・タイプ: L	oader Project	~		
-				

図 3-9 Select toolchain, device & debug settings: RZ/A2M Evaluation Board Kit の場合

4. "Loader Project"はSmart Configuratorに対応していないため、そのまま[次へ(N)>] ボタンで次の画面に進みます。

lew GCC for	Renesas	R7 Executa	ble Projec	+			
Select Coding	Assistant	Settings	isie i rojee				
Use Smart	Configura -ド生成 ^{&}	tor 🌢					
The e2 studio p drivers) for MC based on settir programming i	peripheral U periphen ngs entered interfaces	code generate al functions (d via a graphi (APIs) and are	or and smart clocks, timer cal user inter not limited	: configurator s, serial interf face (GUI). Fu to initializatio	automatically ge aces, A/D conver nctions are provi n of peripheral fu	enerates pro ters, DMA co ded as appl unctions.	grams (device ontrollers, etc.) ication
	UART CSI DMA	Timet Port Clock	Automatic generation of periphs function sottings	App de Dovico drivor	ication under velopment Midaleware RTCS	Software	relier
(?)		< 戻る([3)	次へ(N) >	終了(F)		キャンセル

図 3-10 コーディング・アシスタントツールの選択



e² studio ユーザーズマニュアル

5. 初期設定ではUSB、RTCは使用しない設定としています。"Loader Project"ではそのまま[次へ(N)>] ボタンで次の画面に進みます。

					\times
New GCC for Renesa	s RZ Executable	Project			4
Select Peripheral Usage	Options				4
Select Peripheral Usage	Options				
Use USB0 periphera	I				
Use USB1 periphera	I				
Use RTC0 periphera	L				
Use RTC1 periphera	L				
?	< 戻る(<u>B</u>)	次へ(<u>N</u>) >	終了(<u>F</u>)	キャンセル	,

図 3-11 Peripheral 追加オプションの選択

6. CPU追加オプションはデフォルトの設定を変更せず、[次へ(N)>]ボタンを押してください。

8		$ \Box$ \times
New GCC for Renesas RZ Executable	Project	
Select Additional CPU Options		
Select Additional CPU Options		
Instruction Set	Arm	~
InterWorking		
Target FPU	neon	~
Floating Point ABI	Hard	~
Warn if stack size exceeds the limit, in byte	es 100	
(?) < 戻る(<u>B</u>)	次へ(N) > 終了(E)	キャンセル

図 3-12 CPU 追加オプションの選択

7. 'Project-Built'を選択し、 [次へ(N)>]ボタンを押してください。

2				
New GCC for Renes	sas RZ Executable Pro 設定の選択	ecutable Project ・ リを使用する。		Ż
ライブラリー・ソースの選抜	尺:			
Optimized		 Newlib 		
コードサイズを最適化さ	されたライブラリを使用する。			
<u>ライブラリータイプの選</u> 択	2:			
Project-Built		O Pre-Built		
ユーザーが指定したオス	プションでライブラリーをビルドす	3.		
?	< 戻る(<u>B</u>)	次へ(<u>N</u>) >	終了(<u>F</u>)	キャンセル
0				•••

図 3-13 ライブラリ・ジェネレータ設定の選択

8. プロジェクトの要約が表示されます。最後に[終了(F)]ボタンを押すとプロジェクトが作成されます。



New GCC for Ren	esas RZ Executable	Project		\diamond
Summary of projec	t "Loader"	,		4
Device: R7S921056 Toolchain Version: 9 Files added: generate/syste generate/comp generate/drive generate/syste generate/linker	.3.1.20200408 m/ piler/ rs/ m/ :_script.ld			~

図 3-14 新規プロジェクト作成ウィザード (要約)



9. 「Loader」という新しいCプロジェクトが作成されます。



図 3-15 作成された新規 C プロジェクト

10. 次にApplicationプロジェクトを作成します。 [ファイル] → [新規] → [C/C++ Project] の順にクリックすると、新規プロジェクト作成ウィザードが起動します(下図)。

ファイ	<mark>イル(F)</mark> 編集(E) ソース(S) リファクタリング(T)	ナビゲート(N)	検索(A)	プロジェクト(P)	Renesas Vie	ws 実行(R)) ウィンドウ(W)	ヘルプ(H)
	新規(N)	Alt+シフト+N	>	Renesas C/C++	- Project			>	
	ファイルを開く(.)		C.	Makefile Projec	t with Existin	g Code		_	
۵,	ファイル・システムからプロジェクトを開く		C [°]	C/C++ Project		C または C+	+ プロジェクトを	新規作風	戊 戊
	最近のファイル		> 🗾	プロジェクト(R)…					
		Ctol 114		CIC++ プロミジェノ	7トに 恋 協 (で)	(エエ ネーチャ	_を治したい		i

図 3-16 新規プロジェクト作成ウィザード



11. 新規プロジェクトのテンプレートを選択します。[次へ(N)>] をクリックすると次の画面に進みます。

All CMake Make Renesas Debug Renesas RZ	GNU for Renesas RZ C/C++ Executable Project A C/C++ Executable Project for Renesas RZ using the GNU ARM Embedded toolchain.	^
	GNU for Renesas RZ C/C++ Library Project A C/C++ Library Project for Renesas RZ using the GNU ARM Embedded toolchain. Renesas RZ/A C/C++ FSP Project	
	Renesas RZ/G C/C++ FSP project for Renesas RZ/A Create an executable or static library C/C++ FSP project Create an executable or static library C/C++ FSP proiect for Renesas RZ/G	

図 3-17 プロジェクトテンプレートの選択

12. プロジェクト名"Application"を入力し、[次へ(N)>] で進めます。



New GCC for Renesas	RZ Executable Project	
プロジェクト名(P): Appl	cation	
☑ デフォルト・ロケーション	/の使用(<u>D</u>)	
ロケーション(<u>L</u>):	C:¥Users¥2022-034-user¥e2_studio¥GSG¥Application	参照(<u>R</u>)
	Create Directory for Project	
ファイル・システムを選択(<u>(</u>): デフォルト 〜	
ワーキング・セット		
	プロジェクトを追加(II)	新規(<u>W</u>)
ワーキング・セット(<u>O</u>):		選択(<u>E</u>)

図 3-18 プロジェクト名の指定



13. 以下の設定を選択します。

- 言語:"C" or "C++"
- ツールチェーン: "GCC ARM Embedded"
- ツールチェーン・バージョン: "6.3.1.20170620"
- ターゲット・デバイス: "R7S921053"
- プロジェクトタイプ: "Application Project"

[Hardware Debug 構成を生成] で、"J-Link ARM"を選択し、[次へ(N)>] で進めます。

oolchain Settings			
言語:			
ツールチェーン:	GNU ARM Embedded	~	
ツールチェーン・バージョン	9.3.1.20200408	~	
		ツールチェーンの管理	
RTOS:	None	~	
RTOS Version:		~	
Device Settings			Configurations
Target Board:	ustom	~	
larger board.			✓ Hardware Debug 備成を生成
ターゲット・デバイス: R	75921056		
		デバイスのアンロック	 Release 構成を生成
エンディアン: Li	ttle	~	
プロジェクト・タイプ: A	pplication Project	~	

図 3-19 Select toolchain, device & debug settings: RZ/A2M Evaluation Board Kit の場合



14. 必要に応じて、コーディング・アシスタント機能を使用することができます。

注: [スマート・コンフィグレータ]はコード生成とコンフィグレータを1画面で統一的に扱えるようにしたイン タフェースを提供します。クロック設定、割り込み設定、ピン設定が全て含まれます。

"Use Smart Configurator"を選択して、[次へ(N)>] ボタンで次の画面に進みます。

C	— 🗆 X
New GCC for Renes	ant Settings
✓ Use Smart Config □ Use 周辺コード生成	urator
The e2 studio periphe drivers) for MCU perip based on settings ent programming interfac	ral code generator and smart configurator automatically generates programs (device pheral functions (clocks, timers, serial interfaces, A/D converters, DMA controllers, etc.) ered via a graphical user interface (GUI). Functions are provided as application ces (APIs) and are not limited to initialization of peripheral functions.
	Application Code Software Components Middleware & Drivers Device Drivers MCU Hardware
(0)	

図 3-20 コーディング・アシスタントツールの選択

15. 初期設定ではUSB、RTCは使用しない設定としています。USBおよびRTCをご使用の場合には、使用する周辺モジュール/チャネルを選択してください。

8					\times
New GCC for Renesas	RZ Executable	Project			\$
Select Peripheral Usage	Options				4
Select Peripheral Usage	Options				
Use USB0 peripheral					
Use USB1 peripheral					
Use RTC0 peripheral					
Use RTC1 peripheral					
?	< 戻る(<u>B</u>)	次へ(<u>N</u>) >	終了(<u>F</u>)	キャンセル	
0	/				

図 3-21 Peripheral 追加オプションの選択



16. CPU追加オプションはデフォルトの設定を変更せず、[次へ(N)>]ボタンを押してください。

New GCC for Renesas RZ Executable P	roject	-
Select Additional CPU Options		
Select Additional CPU Options		
Instruction Set	Arm	~
InterWorking		
Target FPU	neon	~
Floating Point ABI	Hard	~
Warn if stack size exceeds the limit, in bytes	100	
(?) < 戻る(<u>B</u>)	次へ(N) > 終了(E)	キャンセル

図 3-22 CPU 追加オプションの選択

17. ライブラリータイプの選択において、 'Project-Built'を選択してください。選択後、[次へ(N)>]ボタンを押 してください。



8					\times
New GCC for Renes	as RZ Executable Pro	ject			\$
ライブラリジェネレーター	設定の選択				4
ライブラリー・ソースの選択	ም:				
Optimized		Newlib			
コードサイズを最適化さ	されたライブラリを使用する。				
─ライブラリータイプの選択	:				
Project-Built		O Pre-Built			
ユーザーが指定したオス	^プ ションでライブラリーをビルドす	ె.			
?	< 戻る(<u>B</u>)	次へ(<u>N</u>) >	終了(<u>F</u>)	キャンセル	

図 3-23 ライブラリ・ジェネレータ設定の選択



18. プロジェクトの要約が表示されます。最後に[終了(F)]ボタンを押すとプロジェクトが作成されます。

					\times
New GCC for Renesas	RZ Executable	Project			4
Summary of project "Ap	oplication"				2
Device: R7S921056					~
Toolchain Version: 9.3.1.2	0200408				
Files added:					
generate/system/					
generate/compiler/	hin m (
generate/configurat	tion/				
generate/os abstrac	ction/				
generate/sc_drivers,	/				
generate/system/					
generate/linker_scri	ot.ld				
					\sim
0		V7 A (NI) >	44	+	
	< 矢の(B)		1 ft	キャノセノ	

図 3-24 新規プロジェクト作成ウィザード (要約)

19. "パースペクティブを開く"を選択します。

📴 関連付けられたパースペクティブを	開きますか?	×
27-ト・コンフィグレータ	ペースペクティブを開きますか?	
 常にこの設定を使用する(<u>R</u>) 		
	パースペクティブを開く(<u>O)</u>	いいえ(<u>N</u>)

図 3-25 パースペクティブ選択画面



20. 「Application」という新しいCプロジェクトが作成されます。



図 3-26 作成された新規 C プロジェクト

このプロジェクトは、"Application_main.c" というアプリケーションファイルと、自動的に生成されるファ イルー式から構成されています。[プロジェクト・エクスプローラー] パネルではすべてのプロジェクトおよ びソースファイルを Windows エクスプローラーと同様のフォルダ階層として表示します。



3.2.2 RZ/A1 のプロジェクト作成(FSP パッケージ無し)

RZ/A1H Renesas Starter Kit を使用する場合の例を説明します。

作成されたプロジェクトは、RZ/A1の大容量内蔵 RAM で動作します。フラッシュメモリで動作させる場合には、お客様にてフラッシュメモリの仕様やメモリ配置に合わせてプロジェクトの内容を変更してください。

Windows の [スタート] メニューから e² studio を起動し、ワークスペースディレクトリを指定します。こ のワークスペースに作成したプロジェクトが追加されます。新規にプロジェクトを作成するには、以下の手 順を実行してください。

[ファイル] → [新規] → [C/C++ Project] の順にクリックすると、新規プロジェクト作成ウィザードが起動します(下図)。

ファ・	イル(F) 編集(E) ソース(S) リファクタリング(T)	ナビゲート(N) 村	_{食索(A}) プロジェクト(P)	Renesas Vie	ews 実行(R)	ウィンドウ(W)	ヘルプ(H)
	新規(N)	Alt+シフト+N >		Renesas C/C++	Project			>
	ファイルを開く(.)		C.	Makefile Project	t with Existin	ig Code		
۵,	ファイル・システムからプロジェクトを開く		¢	C/C++ Project		C または C++	・プロジェクトを	新規作成
	最近のファイル	>	· 🗾	プロジェクト(R)				
	月月15 2 7 2 1 2 1	Ctul . MI	-	$C(C_{++} \exists \Pi \wr)_{\mp} h$	トニ 恋 扬 (C)	ヘート ネーチャー	た。迫加い	

図 3-27 新規プロジェクト作成ウィザード

2. 新規プロジェクトのテンプレートを選択します。[次へ(N)>] をクリックすると次の画面に進みます。

All CMake	GNU for Renesas RZ C/C++ Executable Project A C/C++ Executable Project for Renesas RZ using the GNU
Make Renesas Debug	ARM Embedded toolchain.
Renesas RZ	A C/C++ Library Project for Renesas RZ using the GNU ARM Embedded toolchain.
	Renesas RZ/A C/C++ FSP Project
	<i>Create an executable or static library C/C++ FSP project for Renesas RZ/A.</i>
	Renesas RZ/G C/C++ FSP Project

図 3-28 プロジェクトテンプレートの選択



3. プロジェクト名を入力し、[次へ(N)>] で進めます。

New GCC for Rene	sas RZ Executable Project	
プロジェクト名(P): Tu	Itorial	
✓ デフォルト・ロケージ	ジョンの使用(<u>D)</u>	
ロケーション(<u>L</u>):	C:¥workspace¥Tutorial	参照(<u>R</u>)
	Create Directory for Project	
ファイル・システムを選	択(Y): <mark>デフォルト</mark> ~	
ワーキング・セット		
□ ワーキング・セット	・にプロジェクトを追加(II)	新規(<u>W</u>)
ワーキング・セット(〇)		~ 選択(<u>E</u>)

図 3-29 プロジェクト名の指定

- 言語:"C"
- ツールチェーン: "GNU ARM Embedded"
- ツールチェーン・バージョン: "9.3.1.20200408"
- ターゲット・デバイス: "R7S721001"
 [Hardware Debug 構成を生成] で、"J-Link ARM"を選択し、[次へ(N)>] で進めます。



Full toolchain sup selected one.	pport is only available in GCC	ARM Embedded versio	ons lower than the
Гoolchain Settings च क.		1	
ョ m. ツールチェーン:	GNU ARM Embedded	~	
ツールチェーン・バージョ	ン: 9.3.1.20200408	~	
	1	<u>ツールチェーンの管理</u>	
RTOS:	None	~	
RTOS Version:		~	
Device Settings			Configurations
Target Board:	Custom	~	✓ Hardware Debug 構成を生成
			J-Link ARM ~
ターゲット・ナバイス:	R7S721001	デバイフのマンロック	☐ Release 構成を生成
エンディアン:	ittle		
プロジェクト・タイプ:	デフォルト	~	

図 3-30 Select toolchain, device & debug settings: RZ/A1H Renesas Starter Kit の場合



- 必要に応じて、コード生成サポート機能を選択してください。
 注意:
 - コード生成プラグイン(Peripheral Code Generator)は GUI 操作でデバイスドライバや周辺機能のコードの生成機能をサポートします。周辺機能を操作する関数を API として提供する他、様々な機能を提供します。
 - スマート・コンフィグレータ(Smart Configurator)はコード生成とミドルウェアの設定機能を一つの GUI にま とめたツールです。スマート・コンフィグレータはクロック、割り込みとピンを設定するためのビューも提供 します。
- 5. スマート・コンフィグレータ、コード生成プラグインは一部のデバイスはサポートされていない場合が あります。[次へ(N)>] ボタンで次の画面に進みます。

New GCC for F	enesas RZ Executable Project		-
Select Coding A	ssistant Settings		
Use Smart C	onfigurator 🌢		
Use 周辺コー	≶生成 ≜		
The e2 studio pe drivers) for MCU based on setting	ripheral code generator and smart configurator automatically genera peripheral functions (clocks, timers, serial interfaces, A/D converters, s entered via a graphical user interface (GUI). Functions are provided	ites prog DMA co as applie	grams (device ontrollers, etc.) cation
programming in	terfaces (APIs) and are not limited to initialization of peripheral function	ons.	
E E	Automatic generation Application under		
1	ND Port Settings	f.ware	
=		icrocentre	olier
-			
?	< 戻る(<u>B</u>) 次へ(<u>N</u>) > 終了(E)		キャンセル

図 3-31 コーディング・アシスタントツールの選択

6. CPU追加オプションはデフォルトの設定を変更せず、[次へ(N)>]ボタンを押してください。

		— [
New GCC for Renesas RZ Executa	ble Project		-
Select Additional CPU Options			
Select Additional CPU Options			
Instruction Set	Arm		~
InterWorking			
Target FPU	vfpv3-d16		\sim
Floating Point ABI	Hard		~
Warn if stack size exceeds the limit, in	bytes 100		
? < 戻る(B)) 次へ(N) > 終	了(E) キャ	ンセル

図 3-32 CPU 追加オプションの選択

7. 'Project-Built'を選択し、 [次へ(N)>]ボタンを押してください。

8	—	
New GCC for Renesas RZ Executable Project ライブラリジェネレーター 設定の選択		Ď
ライブラリー・ソースの選択:		
○ Optimized コードサイズを最適化されたライブラリを使用する。	Newlib	
ライブラリータイプの選択:		
 Project-Built ユーザーが指定したオプションでライブラリーをビルドする。 	O Pre-Bulit	
? < 戻る(B) 次/	(N) > 終了(E) キャ	ァンセル

図3-33 ライブラリ・ジェネレータ設定の選択

選択したコンパイラが'Libgen Update for GNU ARM Embedded Toolchains'によって更新されていない場

合、ライブラリ・ジェネレータに関する警告メッセージが次のウィザードに表示されます。これは、プロジェクトビルドエラーの原因となる場合があります。2.2.2節を参照し、'Libgen Update for GNU ARM Embedded Toolchains'を実行してください。

New GCC for Renesas RZ E	xecutable Project		
Library Generator is not insta	lled for the selected toolchain!		
ライブラリー・ソースの選択:			
Optimized	Newlib		
コードサイズを最適化されたライズ	ブラリを使用する。		
ライブラリータイプの選択:			
O Project-Built	Pre-Built		
ユーザーが指定したオプションで	ライブラリーをビルドする。		
?	< 戻る(<u>B</u>) 次へ(<u>N</u>) >	終了(E)	キャンセル

図 3-33 ライブラリ・ジェネレータの設定:警告メッセージ



8. プロジェクトの要約が表示されます。最後に[終了(F)]ボタンを押すとプロジェクトが作成されます。

	—	\times		
New GCC for Renesas RZ Executable Project				
Summary of project "Tutorial"		1		
Device: R7S721001		^		
Toolchain Version: 9.3.1.20200408				
Files added:				
generate/start.S				
generate/inthandler.c				
generate/interrupt_handlers.h				
generate/typedefine.h				
generate/vects.c				
generate/westup.c				
generate/invsetup.c				
generate/linker script.ld				
generate/iodefines/				
,		~		
《戻る(B) 次へ(N) >	終了(E) キャンセ	2110		

図 3-34 新規プロジェクト作成ウィザード (要約)


9. 「Tutorial」という新しいCプロジェクトが作成されます。



図 3-35 作成された新規 C プロジェクト

このプロジェクトは、"Tutorial.c" というアプリケーションファイルと、自動的に生成されるファイルー式 から構成されています。[プロジェクト・エクスプローラー] パネルではすべてのプロジェクトおよびソース ファイルを Windows エクスプローラーと同様のフォルダ階層として表示します。



3.2.3 RZ/G2L, G2LC, G2UL, V2L のプロジェクトの作成(FSP パッケージ有り) RZ/G2L Evaluation Board Kit を使用する場合の例を説明します。

最初に、FSP パッケージをインストールします。

1. GitHubのルネサス製RZ/G, RZ/Vファミリ・マイクロコントローラ用Flexible Software Package (FSP) のページにアクセスしてください。

RZ/G用 : <u>https://github.com/renesas/rzg-fsp/releases</u>

RZ/V用 : <u>https://github.com/renesas/rzn-fsp/releases</u>

以降、ここではRZ/G用プラットフォームインストーラを使用してご説明します。

2. FSPインストーラ("RZG_FSP_Packs_ <version>.exe")を選択し、ダウンロードの直接リンクをクリッ クしてください。

/ID5 Checksums		
 RZG_FSP_Packs_v1.2.0.zip aa2919418ff100981d122a5c RZG_FSP_Packs_v1.2.0.exe b91709d57724a7a6309b55c fsp_documentation_v1.2.0.zip 2e5971970fdd253176b 	ee1f77d78 40e9306748 1dd9f7ff1f631	
Assets 5		
Assets 5	996 KB	Nov 29,
Assets s Øfsp_documentation_v1.2.0.zip ØRZG_FSP_Packs_v1.2.0.exe	996 KB 33.6 MB	Nov 29, Nov 29,
Assets 5 Øfsp_documentation_v1.2.0.zip ØRZG_FSP_Packs_v1.2.0.exe ØRZG_FSP_Packs_v1.2.0.zip	996 KB 33.6 MB 32.7 MB	Nov 29, Nov 29, Nov 29,
Assets 5 Image: Second	996 KB 33.6 MB 32.7 MB	Nov 29, Nov 29, Nov 29, Nov 29, Nov 29,
Assets 5	996 KB 33.6 MB 32.7 MB	Nov 2 Nov 2 Nov 2 Nov 2 Nov 2 Nov 2

図3-36 FSPパッケージのダウンロード

- 3. e² studioが起動している場合、終了してください。
- 4. FSPインストーラ("RZG_FSP_Packs_ <version>.exe")を実行してください。



5. [Next] ボタンを押してインストールを開始してください。



図3-37 FSPパッケージのインストール

6. ソフトウェア契約に同意いただけましたら[I Agree]ボタンを押してください。次に進みます。

W Renesas RZG FSP v1.2.0 Se	etup		X
	RZG FSP v1 .2.0		
RENESAS	Please review the license terms be FSP v1.2.0.	efore installing Renesa	as RZG
Press Page Down to see the	rest of the agreement.		
Copyright [2020-2022] Rene Reserved.	esas Electronics Corporation and/or	r its affiliates. All Righ	its 🔺
and/or its affiliates and may be used with products of Re other uses are authorized. Renesas products are sold p Purchasers are solely respon	only nesas Electronics Corp. and its affi ursuant to Renesas terms and con isible for	iliates ("Renesas"). No	D
express or implied, to any	esas products and Renesas assum	nes no liability. No lice	nse, v
If you accept all terms of the	agreement, dick I Agree.		
· · · · · · · · · · · · · · · · · · ·			Consol
	< <u>B</u> ack	I Agree	Cancel

図3-38 FSPパッケージのインストール



7. e² studioがインストールされているフォルダを選択してください。ここでは C:¥Renesas¥e2_studio¥ に インストールされているものとします。

🐻 Renesas RZG FSP v1.2.0	Setup	Location	—		×
RENESAS	Choose the fold	ler in which to ins	tall Renesas RZG	GFSP v1.2	2.0.
The installation path must (C:¥Renesas¥e2_studio). P	point to the root of lease make sure e.	f the e2 studio in: 2 studio is closed	stallation (e.g. before installatio	n.	
Browse to folder where e	2 studio is installed		Bro	wse]
Space required: 37.0 MB Space available: 659.9 GB					
		< <u>B</u> ack	Install	Can	ncel

図3-40 FSPパッケージのインストール

8. [Finish] ボタンを押してインストールを終了します。

W Renesas RZG FSP v1.2.0 Setu	р — Ц ×
	Completing Renesas RZG FSP v1.2.0 Setup
	Renesas RZG FSP v1.2.0 has been installed on your computer.
	Click Finish to close Setup.
	Open up documentation for this release
	Click here to visit the FSP GitHub repository
	< Back Finish Cancel

図3-41 FSPパッケージのインストール

- Windows の [スタート] メニューから e² studio を起動し、ワークスペースディレクトリを指定します。
 このワークスペースに作成したプロジェクトが追加されます。
 新規にプロジェクトを作成するには、以下の手順を実行してください。
- 10. [ファイル] → [新規] → [C/C++ Project] の順にクリックすると、新規プロジェクト作成ウィザードが起動します(下図)。

ファ	<mark>イル(F)</mark> 編集(E) ソース(S) リファクタリング(T)	ナビゲート(N) 村	食索(A) プロジェクト(P)	Renesas Vie	ws 実行(R)	ウィンドウ(W)	ヘルプ(H)
	新規(N)	Alt+シフト+N >		Renesas C/C++	Project			>
	ファイルを開く(.)		C.	Makefile Projec	t with Existin	g Code		
۵,	ファイル・システムからプロジェクトを開く		C	C/C++ Project		C または C++	- プロジェクトを	新規作成
	最近のファイル	>		プロジェクト(R)				
		Chill MA		<i>CIC</i> ++ プロミジェク	フトに変換 (Ҁ)(ヘェュ ネーチャー	-た:白加い	

11. 新規プロジェクトのRZ/G用テンプレートを選択します。[次へ(N)>] をクリックすると次の画面に進みます。

All CMake Make	GNU for Renesas RZ C/C++ Executable Project A C/C++ Executable Project for Renesas RZ using the GNU ARM Embedded toolchain.
Renesas Debug Renesas RZ	GNU for Renesas RZ C/C++ Library Project A C/C++ Library Project for Renesas RZ using the GNU ARM Embedded toolchain.
	Renesas RZ/A C/C++ FSP Project Create an executable or static library C/C++ FSP project for Renesas RZ/A.
	Renesas RZ/G C/C++ FSP Project Create an executable or static library C/C++ FSP project for Renesas RZ/G.
	Renesas RZ/N C/C++ FSP Project Create an executable or static library C/C++ FSP project for Renesas RZ/N.
٢	

図 3-40 プロジェクトテンプレートの選択



図 3-39 新規プロジェクト作成ウィザード

12. プロジェクト名を入力し、[次へ(N)>] で進めます。

Renesas RZ/G C/C++ FSP Pr	oject			
enesas RZ/G C/C++ FSP P Project Name and Location	Project			Ď
Project name				
Blinky 「デフォルト・ロケーションの使用	用(<u>D</u>)			
ロケーション(<u>L</u>): C:¥workspace	¥Blinky			参照(<u>R</u>)
<u>'ou can download more Rene</u>	<u>sas packs here</u>			

図 3-41 プロジェクト名の指定

13.

- Board : "RZ/G2L Evaluation Kit (SMARC)"
- ・ ツールチェーン: "GNU ARM Embedded"
- ツールチェーン・バージョン: "9.2.1.20191025"
 を選択し、[次へ(N)>] で進めます。



ヘビルコト	እ	門	ガ	イ	ド
-------	---	---	---	---	---

_					
📴 Renesas RZ	/G C/C++ FSP Project				\Box \times
Renesas RZ/	G C/C++ FSP Project				Ď
Device Selection	on				
FSP Version:	1.2.0	×	Board Description		
Board:	RZ/G2L Evaluation Kit (SMARC)	~	Device Details		
Device:	R9A07G044L23GBG_CM33		TrustZone	No	$\hat{}$
<i>C</i>			Pins	456	
Toolchains			Debugger		
GNU ARM En	nbedded		J-Link ARM		~
9.3.1.2020040)8				
?	[< 戻る(<u>B</u>)	次へ(<u>N</u>) >	終了(E)	キャンセル

図 3-42 Device and Tool Selection: RZ/G2L Evaluation Kit の場合



[Build Artifact]に 'Executable' を、[RTOS Selection]に' No RTOS' を選択してください。
 また、[Sub-core start state]には必ず' Secure' を選択してください。そうでない場合、現バージョンではプロジェクトが正しくビルドできません。
 [次へ(N)>] ボタンで次の画面に進みます。

😰 Renesas RZ/G C/C++ FSP Project		
Renesas RZ/G C/C++ FSP Project Build Artifact, RTOS Selection and Sub-Core Selection		Ď
Build Artifact Selection Executable Project builds to an executable file Static Library Project builds to a static library file Executable Using an RZ/G Static Library Project builds to an executable file Project uses an existing RZ/G static library project Sub-core start state Secure Start sub-core in secure state Non-secure Start sub-core in non-secure state 	RTOS Selection No RTOS	~
? < 戻る	5(B) 次へ(N) > 終了(E	キャンセル

図 3-46 Build Artifact, RTOS Selection and Sub-Core Selection



15. [Blinky]テンプレートを選択し、[終了(F)]ボタンを押してください。

Renesas RZ/	G C/C++ FSP Project					
enesas <mark>RZ</mark> /G	C/C++ FSP Project					
Project Templa	te Selection					
Project Templa	te Selection					
•	Bare Metal - Blinky					
	Bare metal FSP project that in	cludes BSP and will I	olink LEDs if available	e. This project will initia	lize cloc	ks, pins,
	stacks, and the C runtime envi [Renesas.RZG.1.2.0.pack]	ronment.				
	environment. [Renesas.RZG.1.2.0.pack]					
Code Generatio	on Settings					
✓ Use Kenesa	s Code Formatter					
?		< 戻る(<u>B</u>)	次へ(<u>N</u>) >	終了(<u>F</u>)		キャンセル

図 3-43 Project Template Selection

16. プロジェクトが作成されると、e² studioの[Project Explorer]ウインドウにプロジェクト名が表示されま す。[Project Configuration]ウインドウの右上にある[Generate Project Content]ボタンを押すと、ご使用の ボード用のファイルを作成することができます。



入門ガイド

ummary				Conception Generate Project	ct Conter
Project Summary	/		REN	ΕՏΔՏ	1
Board:	RZ/G2L Evaluation Kit (SMARC)				
Device:	R9A07G044L23GBG_CM33				
Toolchain:	GCC for Renesas RZ				
Toolchain Version:	9.3.1.20200408				
FSP Version:	1.2.0				
Project Type:	Flat				
Location:	C:/workspace/Blinky 去				
Selected software c	omponents				
Board Support Pac	ckage Common Files	v1.2.0			
General Timer		v1.2.0			
I/O Port		v1.2.0			
Board support pac	kage for R9A07G044L23GBG CM33	v1.2.0			1

図3-44 FSP Configuration

これでプロジェクトが作成され、ビルドの準備ができました。



3.2.4 RZ/T2, N2 のプロジェクトの作成(プラットフォームインストーラ有り) Renesas Starter Kit+ for RZ/T2M を使用する場合の例を説明します。

プラットフォームインストーラによって RZ/T2 用あるいは RZ/N2 用 FSP パッケージがインストールされ ていることが必要です。

まだインストールしていない場合、「2.3 プラットフォームインストーラを用いたインストール」を参照 しインストールを行ってください。

Windows の [スタート] メニューから e² studio を起動し、ワークスペースディレクトリを指定します。こ のワークスペースに作成したプロジェクトが追加されます。

ワークスペース起動時、PC に新しいツールチェーンが存在する場合は、登録を行うことができます。

New toolchains available for integration	I.		С
ソールチェーンの登録			
ツールチェーン・タイプ ❤ ☑ GNU ARM Embedded	インストール・パス		
GNU ARM Embedded - 9.3.1.2	0200408 C:¥Program Files (x86)¥GNU Arm Embeddeo	d Toolchain¥9 2
<			>
Select all Deselect all			
] 起動時に 'ツールチェーンの登録' を有効にす	3		

図3-45 新規ツールチェーンの登録

新規にプロジェクトを作成するには、以下の手順を実行してください。



[ファイル] → [新規] → [C/C++ Project] の順にクリックすると、新規プロジェクト作成ウィザードが起動します(下図)。

ファ	イル(F) 編集(E) ソース(S) リファクタリング(T)	ナビゲート(N)	検索(A) プロジェクト(P)	Renesas Vie	ews 実行(R)	ウィンドウ(W)	ヘルプ(H)
	新規(N)	Alt+シフト+N	>	Renesas C/C++	- Project			>
	ファイルを開く(.)		C.	Makefile Project	t with Existir	ng Code		
	ファイル・システムからプロジェクトを開く		C	C/C++ Project		C または C++	+ プロジェクトを	新規作成
	最近のファイル		> 🖸	プロジェクト(R)				
		ChilliaM	100	$C(C_{\pm\pm} \exists \Pi \wr J_{\pm})$	7トに 変換 (C)	アート ネーチャー	_ たら白 加 \	

図 3-46 新規プロジェクト作成ウィザード

2. 新規プロジェクトのRZ/T用テンプレートを選択します。[次へ(N)>] をクリックすると次の画面に進みます。

🎒 新規 C/C++ フロ゙		×
新規 C/C++ プロジ	ジェクトのテンプレート	
• 11		
All	Renesas RZ/G C/C++ FSP Project	^
CMake	Greate an executable or static library C/C++ FSP project for Renesas F	?Z/G.
Make		
Renesas Debug	Renesas RZ/N C/C++ ESP Project	
Renesas RZ	Create an executable or static library $C/C++$ FSP project for Renesas F	Z/N.
	Renesas RZ/T C/C++ FSP Project	
	Create an executable or static library C/C++ FSP project for Renesas F	RZ/T.
	Renesas RZ/V C/C++ FSP Project	
	Create an executable or static library C/C++ FSP project for Renesas F	Z/V.
		~
	11	
?	< 戻る(B) 次へ(N) > 終了(F) キャンセ	IL
U		

図 3-47 プロジェクトテンプレートの選択



3. プロジェクト名を入力し、[次へ(N)>] で進めます。

Proiect Name a	nd Location
<u>P</u> roject name	
Blinky	
🗹 デフォルト・	ロケーションの使用(<u>D</u>)
ロケーション(<u>L</u>):	C:¥workspace¥Blinky 参照(R)
	ファイル・システムを選択(Y): デフォルト ~
You can downlo	oad more Renesas packs here
<u>You can downl</u>	oad more Renesas packs here
<u>(ou can downl</u>	<u>pad more Renesas packs here</u>
<u>You can downl</u>	<u>pad more Renesas packs here</u>

図 3-48 プロジェクト名の指定



4.

- Board : "RSK+RZT2M"
- ツールチェーン: "GNU ARM Embedded"
- ツールチェーン・バージョン: "9.3.1.20200408"
 を選択し、[次へ(N)>] で進めます。

Device Selection	on			
FSP Version:	1.2.0 ~	Board Description	on	
Board:	RSK+RZT2M (16-bit bus NOR flash $ imes $	Ronocae Startor	Kit+ for R7/T2M	CPLI Roard (16-hit k
Device:	R9A07G075M24GBG	Device Details		
Corre		TrustZone	No	^
Core:	CK52_0 V	Pins	320	~
oolchains		Debugger		
GNU ARM Er	nbedded	J-Link ARM		×
GCC ARM A-	Profile (AArch64 bare-metal)			

図 3-49 Device and Tool Selection: RZ/G2L Evaluation Kit の場合



5. [Build Artifact]に 'Executable' を、[RTOS Selection]に' No RTOS' を選択してください。 [次へ(N)>] ボタンで次の画面に進みます。

🔄 Renesas RZ/T C/C++ FSP Project		
Renesas RZ/T C/C++ FSP Project		-
Build Artifact and RTOS Selection		
Build Artifact Selection	RTOS Selection	
 Executable Project builds to an executable file 	No RTOS	~
 Static Library Project builds to a static library file 		
, ,		

図 3-50 Build Artifact, RTOS Selection



6. [Blinky]テンプレートを選択し、[終了(F)]ボタンを押してください。

Project Temp	ate Selection	
0	🕈 Bare Metal - Blinky	^
EO	Bare metal FSP project that includes BSP and will blink LEDs if available. This project will	
	initialize clocks, pins, stacks, and the C runtime environment.	
	[Renesas.RZT.1.2.0.pack]	
	Baro Motal - Minimal	
	Bare metal FSP project that includes BSP. This project will initialize clocks, pins, stacks, and the Cruptime environment.	
	[Poposac PZT1 2.0 pack]	
		~
Code Genera	tion Settings	
🗸 Use Rene	sas Code Formatter	

☑ 3-51 Project Template Selection



プロジェクトが作成されると、e² studioの[Project Explorer]ウインドウにプロジェクト名が表示されます。[Project Configuration]ウインドウの右上にある[Generate Project Content]ボタンを押すと、ご使用のボード用のファイルを作成することができます。

[Blinky] FSP Configurat	ion ×			
ummary			Senerate Project (Content
Project Summary	/	REN	IESAS	
Board:	RSK+RZT2M (16-bit bus NOR flash boot mode)			
Device:	R9A07G075M24GBG			
Toolchain:	GCC for Renesas RZ			
Toolchain Version:	9.3.1.20200408			
FSP Version:	1.2.0			
Project Type:	Flat			
Location:	C:/workspace/Blinky 😓			
Selected software co	omponents			
Simple application	that blinks an LED. No RTOS included.	v1.2.0)	
Board Support Pac	ckage Common Files	v1.2.0)	
I/O Port		v1.2.()	
Arm OMOLO Maraia	Dr. F. Coro (D)	UE 7 (Uronocco 1	

図3-52 FSP Configuration

これでプロジェクトが作成され、ビルドの準備ができました。



3.2.5 RZ/A3UL のプロジェクトの作成(プラットフォームインストーラ有り)

RZ/A3UL Evaluation Board Kit OCTAL Edition を使用する場合の例を説明します。

プラットフォームインストーラによって RZ/A3UL 用 FSP パッケージがインストールされていることが必要です。

まだインストールしていない場合、「2.3 プラットフォームインストーラを用いたインストール」を参照 しインストールを行ってください。

Windows の [スタート] メニューから e² studio を起動し、ワークスペースディレクトリを指定します。こ のワークスペースに作成したプロジェクトが追加されます。

ワークスペース起動時、PC に新しいツールチェーンが存在する場合は、登録を行うことができます。

New toolchains available for integratio	'n			C	
ソールチェーンの登録					
ツールチェーン・タイプ ✔ ✔ GNU ARM Embedded	インスト	~~ル・パス			
GNU ARM Embedded - 9.3.1.	20200408 C:¥Pro	ogram Files (x86)¥	GNU Arm Embed	ded Toolchai	n¥9 2
<					>
Select all Deselect all					
起動時に 'ツールチェーンの登録' を有効にす	5				

図3-53 新規ツールチェーンの登録

新規にプロジェクトを作成するには、以下の手順を実行してください。



[ファイル] → [新規] → [C/C++ Project] の順にクリックすると、新規プロジェクト作成ウィザードが起動します(下図)。

ファ	イル(F) 編集(E) ソース(S) リファクタリング(T)	ナビゲート(N) 核	検索(A) プロジェクト(P)	Renesas Vie	ews 実行(R)	ウィンドウ(W)	ヘルプ(H)
	新規(N)	Alt+シフト+N >		Renesas C/C++	Project			>
	ファイルを開く(.)		C.	Makefile Project	t with Existir	ng Code		
<u></u>	ファイル・システムからプロジェクトを開く		¢	C/C++ Project		C または C++	+ プロジェクトを	新規作成
	最近のファイル	>	Ľ	プロジェクト(R)				
		Ctul 11M	C 11	<i>CIC</i> ±± プロミジェノ	カトに変換 (で)	パュュ ネーチャー	_た;色力の)	

図 3-54 新規プロジェクト作成ウィザード

2. 新規プロジェクトのRZ/A3UL用テンプレートを選択します。[次へ(N)>] をクリックすると次の画面に進みます。



図 3-55 プロジェクトテンプレートの選択



3. プロジェクト名を入力し、[次へ(N)>] で進めます。

🛐 Renesas RZ//	A C/C++ FSP Projec	ct				
Renesas RZ/A Project Name a	A C/C++ FSP Proj	ect				ightharpoonup
<u>P</u> roject name						
Blinky ✓ デフォルト・	・ロケーションの使用(D)				
ロケーション(<u>L</u>):	C:¥workspace¥Bli	nky				参照(<u>R</u>)
	ファイル・システムを達	᠍択(⊻): デフォルト ∨				
<u>You can downl</u>	load more Renesas	<u>packs here</u>				
?		< 戻る(<u>B</u>)	次へ(<u>N</u>) >	終了	(<u>F</u>)	キャンセル

図 3-56 プロジェクト名の指定



4.

- Board : "RZ/A3UL Evaluation Board Kit OCTAL Edition"
- ツールチェーン: "GNU ARM A-Profile"
- ツールチェーン・バージョン: "9.3.1.20200408"
 を選択し、[次へ(N)>] で進めます。

📴 Renesas RZ	/A C/C++ FSP Project			— 🗆 X
Renesas RZ/	A C/C++ FSP Project			-
Device and To	pols Selection			
Device Selection	on			
FSP Version:	1.2.1 ~	Board Descriptior	١	
Board:	RZ/A3UL Evaluation Board Kit OCTAL Ed $ imes$	Device Details		
Device:	R9A07G063U02GBG	TrustZone	No	^
Core:	Core 0 V	Pins	361	~
Toolchains	~ ~	Debugger		
GCC ARM A-	Profile (AArch64 bare-metal)	J-Link ARM		×
10 3 1 202106	521			
10.5.1.202100				
?	< 戻る(<u>B</u>)	次へ(<u>N</u>) >	終了(<u>F</u>)	キャンセル

図 3-57 Device and Tool Selection: RZ/A3UL Evaluation Board Kit OCTAL Edition の場合



5. [Build Artifact]に 'Executable' を、[RTOS Selection]に' No RTOS' を選択してください。 [次へ(N)>] ボタンで次の画面に進みます。

Renesas RZ/A C/C++ FSP Project	— 🗆 X
Renesas RZ/A C/C++ FSP Project Build Artifact and RTOS Selection	
Build Artifact Selection	RTOS Selection
 Executable Project builds to an executable file 	No RTOS ~
 Static Library Project builds to a static library file Executable Using an RZ/A Static Library Project builds to an executable file Project uses an existing RZ/A static library project 	
(?) < 戻る(<u>B</u>)	次へ(<u>N</u>) > 終了(<u>F</u>) キャンセル

図 3-58 Build Artifact, RTOS Selection



6. [Blinky]テンプレートを選択し、[終了(F)]ボタンを押してください。

😰 Renesas RZ/A	C/C++ FSP Project				×
Renesas RZ/A	C/C++ FSP Project				\$
Project Templat	Selection				4
Project Templat	Selection				
•	Bare Metal - Blinky Bare metal FSP project that includes BSP and clocks, pins, stacks, and the C runtime enviro [Renesas.RZA.1.2.1.pack]	will blink LEDs if available. T nment.	his projec	t will initia	lize
0	Bare Metal - Minimal Bare metal FSP project that includes BSP. This runtime environment. [Renesas.RZA.1.2.1.pack]	project will initialize clocks,	pins, stacl	ks, and the	e C
Code Generatic	n Settings Code Formatter				
?	< 戻る(<u>B</u>) 次へ(⊻)> 終了(E)		キャンセル	

図 3-59 Project Template Selection



プロジェクトが作成されると、e² studioの[Project Explorer]ウインドウにプロジェクト名が表示されます。[Project Configuration]ウインドウの右上にある[Generate Project Content]ボタンを押すと、ご使用のボード用のファイルを作成することができます。

[Blinky] FSP Configurat			
ummary			Generate Project Conten
Project Summary	/	RENE	SAS
Board:	RZ/A3UL Evaluation Board Kit OCTAL Edition (eXecute-In-Place)	
Device:	R9A07G063U02GBG		
Core:	Core 0		
Toolchain:	GCC for Renesas RZ (A-Profile AArch64 bare-metal)		
Toolchain Version:	10.3.1.20210621		
FSP Version:	1.2.1		
Project Type:	Flat		
Location:	C:/workspace/Blinky 😓		
Selected software c	omponents		
Simple application	that blinks an LED. No RTOS included.	v1.2.1	
RZA Configuration	i Template File	v1.2.1	1
Doord Cupport Do	akaga Camman Eilaa	v1 0 1	

図3-60 FSP Configuration

これでプロジェクトが作成され、ビルドの準備ができました。

3.3 プロジェクトをバックアップする際の注意事項

- 「.」(ドット)で始まる名前のファイルやフォルダ(.project など)にはプロジェクトの設定情報が含まれますので、バックアップを取る際にはこれらのファイルやフォルダも含めてプロジェクトのフォルダ全体を圧縮するなどしてください。
- 他のプロジェクトのファイルを参照する設定など、プロジェクト間で共有される設定を保存するためには、ワークスペース全体をバックアップする必要があります。



4. ビルド

この章では、e² studio 統合開発環境の主要なビルド機能と設定方法について説明します。

4.1 ビルドオプションの設定

プロジェクト作成時のデフォルトのビルドオプションでビルド・実行する事は可能です。オプションを変 更したい場合には、以下で説明する設定画面をお使いください。

1. プロパティ設定画面の起動

まずプロジェクトのプロパティ設定画面を起動します。e² studio のプロジェクト・エクスプローラで、 プロジェクト名の箇所を右クリックしてコンテクスト・メニューの「プロパティ」を選択するか、プロ ジェクトを選択した状態で「プロジェクト」メニューの「プロパティ」を選択するか、[Alt]+[Enter]を押 してください。

プロジェクト・エクスプローラのどこを選択してプロパティの設定画面を表示したかで設定の及ぶ範囲 が変わります。ワークスペース、プロジェクト、フォルダ、個別のソースコードそれぞれのレベルでの 設定が可能です。プロジェクトを選択しない状態でプロパティ設定画面を起動するとワークスペース全 体のプロジェクトやファイルに対し一括して設定を行うことができます。



図 4-1 プロジェクトのプロパティ設定画面を起動する



2. ツールチェーン・バージョンの確認

[C/C++ビルド] → [設定] の「Toolchain」タブをクリックすると、ツールチェーンとそのバージョンが表示されます。作成・インポート済のプロジェクトでもここの設定でツールチェーン・バージョンを変更できます。

ツールチェーン管理画面で有効にしたバージョンが選択肢として表示されます。

	设定	
> リソ-ス ▼C/C++ ビルド ツールチェイン・エディタ- ビルド変数	Configuration: HardwareDebug [アクティブ]	→ 構成の管理
ロギング 環境 設定 > C/C++ 一般 Renesas QE Task Tags > Validation ビルダー プロジェクト・ネーチャー プロジェクト・ネーチャー プロジェクト参照 実行/デバッグ設定	 >> ツール設定 >> Toolchain >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	エラー・パーサー
?	適用して閉じる	キャンセル

図 4-2 ツールチェーン・バージョンの確認・変更画面



3. ビルド環境変数

環境変数は e² studio を経由してビルドコマンドに引き渡されます。環境変数の内容を確認するには、 [C/C++ ビルド] → [環境]を選択してください。

プロジェクトで独自の環境変数を設定することも可能です(下記の例ではテンポラリフォルダを指定)。

/タ入力 ソース /C++ ビルド ツールチェイン・エディター	環境 Configuration: Hardw	vareDebug [アクティブ]	<-> ▼ <->
ビルド変数 ロギング 環境	設定する環境変数		jį
ixxと i/C++ 一般 kenesas QE ask Tags falidation fluジェクト・ネーチャー fluジェクト・ネーチャー fluジェクト・参照 冬行/デパッグ設定	変数 CWD GCC_VERSION PATH PWD TCINSTALL TC_VERSION	値 C:¥Renesas¥workspace¥GSG¥Tutorial¥HardwareDebug 9.3.1 C:¥Program Files (x86)¥GNU Arm Embedded Toolchain¥9 202 C:¥Renesas¥workspace¥GSG¥Tutorial¥HardwareDebug C:¥Program Files (x86)¥GNU Arm Embedded Toolchain¥9 202 9.3.1.20200408	由来 ビルド・システム ビルド・システム ビルド・システム ビルド・システム ビルド・システム ビルド・システム
	< ネイティブ環境へ変数 ネイティブ環境を指定 	な追加 された環境と置換 デフォノ	> レトの復元(II) 適用(

図 4-3 環境変数設定画面

4. ビルドオプションの設定

ビルドオプションによって、パス名を含むすべてのツールチェーン設定を維持することができます。図 4-3 に示すとおり、現在のビルド構成は、「HardwareDebug [アクティブ]」です。

ビルドオプションの詳細は、各コンパイラ製品のユーザーズマニュアル(ヘルプ)を参照してください。 インストールディレクトリの doc フォルダ内にあります。

(例:C:¥Program Files (x86)¥GNU Arm Embedded Toolchain¥9 2020-q2-update¥share¥doc)

注: 上記画面とは別に「Tool chain エディター」設定画面もありますが、これは Renesas 製ビルドプラグ インがサポートしていないコンパイラ製品で設定が必要になるものです。通常は使用しないでください。



4.2 プロジェクトのビルド

以下いずれかの方法でプロジェクトがビルドできます。

- プロジェクト・エクスプローラでプロジェクトを右クリックして「プロジェクトのビルド」を選択する。
- プロジェクト・エクスプローラでプロジェクトを選択状態にし、プロジェクトメニューの「プロジェクトのビルド」を選択する。
- 3. 2.でプロジェクトメニューの代わりに「ファイル」ツールバーの 🔦 ボタンを押す。
- 4. 2.でプロジェクトメニューの代わりに、Ctrl+Bを押す。



図 4-4 プロジェクトメニューによるビルド

- 5. ビルドが成功すると、[コンソール]に "Build Finished." というメッセージが表示されます。ビルドの最後 に\${CWD}ディレクトリに出力されるファイルは、"makefile"、"<プロジェクト名>.elf"、"<プロジェクト 名>.map"、"<プロジェクト名>.hex" で構成されます。
- 6. "<プロジェクト名>.elf"は、デバッグに使用するELF/DWARFフォーマットの標準ロードモジュールで す。

入門ガイド

 e^2 studio ユーザーズマニュアル

xtracting support files 1:56:40 **** プロジェクト Tutorial 伝対する構成 HardwareDebug のビルド **** hake -j8 all prm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp prm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc @"Tutorial.elf.in" prm-none-eabi-gcc @"Tutorial.elf.in" prm-none-eabi-sizeformat=berkeley "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf				
1:56:40 **** プロシェクト Tutorial に対する構成 HardwareDebug () ビルド **** hake -j8 all irm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp irm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp irm-none-eabi-gcc @"Tutorial.elf.in" irm-none-eabi-gcc @"Tutorial.elf.in" irm-none-eabi-sizeformat=berkeley "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf	Extracting suppor	t files		^
<pre>nake -j8 all rm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp rm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc @"Tutorial.elf.in" rm-none-eabi-gcc @"Tutorial.elf.in" rm-none-eabi-objcopy -0 srec "Tutorial.elf" "Tutorial.srec" rm-none-eabi-sizeformat=berkeley "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf</pre>	11:56:40 **** フロシ	バロクト Tutorial に	E対する構成 HardwareDebug ()) ビルド ****	
<pre>irm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp irm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc @"Tutorial.elf.in" irm-none-eabi-gcc @"Tutorial.elf.in" irm-none-eabi-objcopy -0 srec "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf</pre>	make -j8 all			
<pre>irm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp irm-none-eabi-gcc @"Tutorial.elf.in" irm-none-eabi-gcc @"Tutorial.elf.in" irm-none-eabi-objcopy -0 srec "Tutorial.elf" irm-none-eabi-sizeformat=berkeley "Tutorial.elf" itext data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf</pre>	arm-none-eabi-gcc	-mcpu=cortex-a	a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp	
<pre>Imm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp Imm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp Imm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc @"Tutorial.elf.in" Imm-none-eabi-gcc @"Tutorial.elf.in" Imm-none-eabi-gcc @"Tutorial.elf.in" Imm-none-eabi-sizeformat=berkeley "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf</pre>	arm-none-eabi-gcc	-mcpu=cortex-a	a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp	
<pre>rm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp rm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc @"Tutorial.elf.in" rm-none-eabi-objcopy -0 srec "Tutorial.elf" "Tutorial.srec" rm-none-eabi-sizeformat=berkeley "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf</pre>	arm-none-eabi-gcc	-mcpu=cortex-a	a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp	
<pre>rm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp rm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc @"Tutorial.elf.in" rm-none-eabi-objcopy -0 srec "Tutorial.elf" "Tutorial.srec" rm-none-eabi-sizeformat=berkeley "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf</pre>	arm-none-eabi-gcc	-mcpu=cortex-a	a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp	
<pre>urm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp arm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mf urm-none-eabi-gcc @"Tutorial.elf.in" urm-none-eabi-objcopy -0 srec "Tutorial.elf" "Tutorial.srec" urm-none-eabi-sizeformat=berkeley "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf</pre>	arm-none-eabi-gcc	-mcpu=cortex-a	a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfp	
arm-none-eabi-gcc -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mf mrm-none-eabi-gcc @"Tutorial.elf.in" mrm-none-eabi-objcopy -O srec "Tutorial.elf" "Tutorial.srec" mrm-none-eabi-sizeformat=berkeley "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf	arm-none-eabi-gcc	-mcnu-contex-a	a9 -march-armv7-a -marm -mlittle-endian -mfloat-ahi-hard -mfr	
rm-none-eabi-gcc @"Tutorial.elf.in" rm-none-eabi-objcopy -O srec "Tutorial.elf" "Tutorial.srec" rm-none-eabi-sizeformat=berkeley "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf	arm none eaur-gcc		ay march-army, a marmy miller charan milloac abi-hara mip	
rm-none-eabi-objcopy -O srec "Tutorial.elf" "Tutorial.srec" rm-none-eabi-sizeformat=berkeley "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf	'arm-none-eabi-gcc	c -mcpu=cortex-	-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mf	:
rm-none-eabi-sizeformat=berkeley "Tutorial.elf" text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf	'arm-none-eabi-gcc arm-none-eabi-gcc	c -mcpu=cortex- @"Tutorial.elf	-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mf f.in"	
text data bss dec hex filename 15380 0 0 15380 3c14 Tutorial.elf	'arm-none-eabi-gcc arm-none-eabi-gcc arm-none-eabi-gcc	@"Tutorial.elf copy -0 srec "T	-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mf f.in" Tutorial.elf" "Tutorial.srec"	
15380 0 0 15380 3c14 Tutorial.elf	'arm-none-eabi-gcc arm-none-eabi-gcc arm-none-eabi-gcc arm-none-eabi-objo arm-none-eabi-sizo	@"Tutorial.elf copy -0 srec "Teformat=berk	-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mf f.in" Tutorial.elf" "Tutorial.srec" kelev "Tutorial.elf"	
	'arm-none-eabi-gcc arm-none-eabi-gcc arm-none-eabi-obj arm-none-eabi-sizo text data	@"Tutorial.elf copy -0 srec "T eformat=berk	-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mf f.in" Tutorial.elf" "Tutorial.srec" keley "Tutorial.elf" bex filename	
	'arm-none-eabi-gcc arm-none-eabi-gcc arm-none-eabi-obj arm-none-eabi-sizi text data 15380 0	@"Tutorial.elf copy -0 srec "T eformat=berk bss dec 0 15380	-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mf f.in" Tutorial.elf" "Tutorial.srec" keley "Tutorial.elf" hex filename 3c14 Tutorial elf	
	'arm-none-eabi-gcc arm-none-eabi-gcc arm-none-eabi-obj arm-none-eabi-siz text data 15380 0	@"Tutorial.elf @"Tutorial.elf copy -0 srec "T eformat=berk bss dec 0 15380	<pre>-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mf f.in" Tutorial.elf" "Tutorial.srec" keley "Tutorial.elf" hex filename 3c14 Tutorial.elf</pre>	
1:56:41 Build Finished. 0 errors, 0 Warnings. (took /29ms)	'arm-none-eabi-gc arm-none-eabi-gc arm-none-eabi-obj arm-none-eabi-siz text data 15380 0 11:56:41 Build Fin	e -mcpu-contex- @"Tutorial.elf copy -O srec "T eformat=berk bss dec 0 15380 nished. 0 error	<pre>-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mf f.in" Tutorial.elf" "Tutorial.srec" keley "Tutorial.elf" hex filename 3c14 Tutorial.elf rs, 0 warnings. (took 729ms)</pre>	
1:56:41 Build Finished. 0 errors, 0 Warnings. (Took 729ms)	'arm-none-eabi-gc arm-none-eabi-gc arm-none-eabi-obj arm-none-eabi-siz text data 15380 0 11:56:41 Build Fin	e -mcpu=cortex- @"Tutorial.elf copy -O srec "T eformat=berk bss dec 0 15380 nished. 0 error	<pre>-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mf f.in" Tutorial.elf" "Tutorial.srec" keley "Tutorial.elf" hex filename 3c14 Tutorial.elf rs, 0 warnings. (took 729ms)</pre>	~

図 4-5 ビルドコンソールの表示例 (ビルド成功時)



4.3 ビルド構成のレポート

プロジェクトレポート機能により、プロジェクトとビルド構成を e² studio からファイルに出力して、プロ ジェクトやビルド環境の設定を容易にチェックし、比較することができます。

- 1. [プロジェクト・エクスプローラー]を右クリックしてコンテクスト・メニューを開きます。
- 2. [Renesas C/C++ Project Settings] から[Save build settings report] を選択してビルド構成レポートを保存 します。



図 4-6 プロジェクトレポート



5. デバッグ

この章では、e² studio 統合開発環境のデバッグ構成と主要なデバッグ機能の使い方について説明します。 以下では、Segger J-Link エミュレータと RSK RZ/A1H ボードを動作環境とする "Tutorial" プロジェクト(4.2 節)を例に説明します。

デバッグに使用する特定のパースペクティブを開くには、e² studio の "Tutorial" プロジェクトワークスペースを開き、[デバッグ] パースペクティブをクリックします。



図 5-1 [デバッグ] パースペクティブへの切り替え

パースペクティブでワークベンチのウィンドウのレイアウト表示(開発ツール関連)を定義します。それぞれ のパースペクティブは、ビュー、メニュー、ツールバーの組み合わせで構成され、それによりユーザは特定 のタスクを実行できます。

例えば、[C/C++] パースペクティブは、ユーザが C/C++プログラム開発を行うために必要なビューを持ち、[デバッグ] パースペクティブは、プログラムのデバッグに必要なビュー表示をすることができます。ユ ーザが [C/C++] パースペクティブ表示中にデバッガに接続しようとする場合、e² studio は [デバッグ] パー スペクティブに切り替えるようユーザを促します。

ーつのワークベンチのウィンドウのなかに、一つまたは複数のパースペクティブを表示することができま す。ユーザはそれらをカスタマイズしたり、新しいパースペクティブを追加することができます。

注意 : デバッグについての詳細は、e² studio メニューの [ヘルプ] → [ヘルプ目次] → [e2 studio ユーザーガイド] → [デバッグに関する機能] を参照してください。



5.1 既存デバッグ構成の変更

初めてデバッグを行う際には、一度だけデバッグ構成を設定する必要があります。既存のデバッグ構成は 以下のように変更できます。

- 1. [プロジェクト・エクスプローラー]の "Tutorial" プロジェクトをクリックします。
- 2. [実行] → [デバッグの構成...] あるいは ^歩 アイコン(下向き矢印) → [デバッグの構成] の順にクリック し、[デバッグ構成] ウィンドウを開きます。

SG - e ² studio Tァイル(E) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(E) Renesas <u>V</u> iews w ● ◆ デバッグ(B) ◇ ■ Tutorial HardwareDebu ◇ 参 : □ ▼ ■ № ▼ 冬 ▼ ■ : □	ー 実行(R) ウィンドウ(W) ヘルプ(出) 図 TraceX 図 Tracealyzer	
	 ・ 実行(R) *・ デバッグ(D) 実行履歴(T) ● 実行(S) 	Ctrl+F11 F11 >
	デバッグ履歴(H)	>
記問題	 ブレークポイントの切り替え(K) 行ブレークポイントの切り替え(L) 監視ポイントの切り替え(W) メソッド・ブレークポイントの切り替え(M) すべてのブレークポイントを非いブ(I) すべてのブレークポイントを削除(V) 	Ctrl+シフト+B
<	♣ 外部ツール(E)	>
S Tutorial	2048M 中の 1099M iii	

図 5-2 [デバッグ構成]ウィンドウを開く



[デバッグ構成] ウィンドウで、"Renesas GDB Hardware Debugging" デバッグ構成の表示を展開し、既存のデバッグ構成をクリックしてください(例: "Tutorial HardwareDebug")。[メイン] タブを選択し、デバッグ対象のロードモジュールが指定されているかを確認してください(例: "Tutorial.elf")。

📴 デバッグ構成					
構成の作成、管理、および実行					to.
 ○ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	名前(M): Tutorial HardwareDebug ■メイン 参 Debugger ● Startup プロジェクト(P): Tutorial C/C++ アプリケーション: HardwareDebug/Tutorial.elf 起動前に必要に応じてビルド Build Configuration: Select Auto ○ 自動ビルドを有効にする ● ワークスペース設定の使用	□ 共通(<u>C</u>) watically	-ス 変数(⊻) ○ 自動じル ワークスペー	プロジェクトの検索(出)… ドを無効にする ス設定の構成…	参照(B) 参照(R)
く > > 16 項目のうち 14 項目がフィルターに一致			前回	回保管した状態に戻す(V)	適用(<u>Y</u>)
0				デバッグ(<u>D</u>)	閉じる

図 5-3 ロードモジュールの選択

4. [Debugger] タブに切り替え、エミュレータの種類とデバイス名を選択してください。この例ではDebug hardwareに "J-Link ARM"、Target Deviceに "R7S721001" を設定しています。

🖹 メイン (参 Debugger) 🕨 Startup 🗉 共対	通(C) [シソース]
Debug hardware: J-Link ARM 🛛 🗸 🗸	Target Device: R7S721001
GDB Settings Connection Settings デ	バッグ・ツール設定
GDB 接続設定:	
● ローカル GDB サーバーを自動起動	ホスト名または IP アドレス: localhost
○ リモート GDB サーバーへ接続	GDBポート番号: 61234

図 5-4 ターゲットデバイスの選択

- 5. [Debugger] タブの中の [Connection Settings] サブタブでは、Segger J-Linkエミュレータ、および RSK RZ/A1Hボードの設定を元に、以下を設定します。
 - Interface
 - Type = "SWD"
 - Speed = "Auto"
 - 注意: 以上の設定内容は例として示したものです。誤った設定は誤作動やハードウェアの故障の原因となりますの で、接続の前に設定内容がボードの仕様と合っているかを慎重にご確認ください。

デバッグ構成		- 0
成の作成、管理、および実行		X
	名前(<u>M</u>): Tutorial HardwareDebug	
© C/C++ アノリケーション © C/C++ リモート・アプリケーション ■ EASE Script ■ GDB OpenOCD Debugging	Debug hardware: J-Link ARM Y Target Devic GDB Settings Connection Settings デバッグ・ツール話	ce: R7S721001 3 2 定
GDB Simulator Debugging (RH850)	✓ J-Link	
▣ GDB ハードウェア・デバッギング	Туре	USB 🗸
II Java アプリケーション	J-Link Serial	(Auto)
🖻 Java アプレット	設定ファイル	<pre>\${workspace_loc:\\${ProjName}}\\${LaunchConfigN</pre>
Launch Group (Deprecated)	Script File	
🖻 Renesas GDB Hardware Debugging	Low Power Handling	No
Tutorial HardwareDebug	✓ Interface	
🖻 Renesas Simulator Debugging (RX, F	Туре	SWD
🖳 リモート Java アプリケーション	Speed (kHz)	Auto
☞ 起動グループ	✓ JTAG Scan Chain	
	Multiple Devices	
	IRPre	0
	DRPre	0
	▼ 接続	144.2
	レジスターの初期化	
	接続時にリセット	
	実行前にリセット	
	ID コード (バイト単位)	
項目のうち 14 項目がフィルターに一致		前回保管した状態に戻す())適用())
<u>?</u>		デバッグ(<u>D)</u> 閉じる

図 5-5 接続設定の変更



- 6. サブタブ [デバッグ・ツール設定] では、RSK RZ/A1H ボードに基づいて、以下を設定してください。
 メモリー
 - エンディアン = "リトル・エンディアン"

ቬ成の作成、管理、および実行)
1 🖗 🕼 🗙 🖻 🍸 ▾	夕前(M): Tutorial HardwareDobug		
		- 1	
E C/C++ アプリケーション	■ X1 2 ^w Debugger V Startup □ 共通(<u>C</u>)	y-x	
■ C/C++ リモート・アプリケーション	Debug hardware: J-Link ARM 🛛 🗸 Targ	et Device: R7S721001	
EASE Script			
• GDB OpenOCD Debugging	GDB Settings Connection Settings デバッグ・	ツール設定	
🖻 GDB Simulator Debugging (RH85	✓ 10		^
🖻 GDB ハードウェア・デバッギング	デフォルト IO ファイル名を使用	はい	~
🗉 Java アプリケーション	10 ファイル名	\${support_area_loc}	
🖭 Java アプレット		14	
Launch Group (Deprecated)	タリンロート後にリゼットする	(40)	~
🖻 Renesas GDB Hardware Debuggii		リトル・エンディアン	
Tutorial HardwareDebug		5196-1227172	· · · ·
Renesas Simulator Debugging (R	* 〒町 フラッシュ・ブレークポイントを使用する	はい	~
営 リモート Java アフリクーション = おまれガル ニプ	シミュレーションを許可する	いいえ	~
	✓ Flash		
	Flash Bus Type		~ ~
く 5 項目のうち 14 項目がフィルターに一致		前回保管した状態に戻す(火)	適用(Y)
2		デバッグ(ワ)	問じる

図 5-6 デバッグ・ツール設定の変更

- [適用] ボタンを押すと設定が保存されます。[デバッグ] ボタンを押すと、デバッガが起動し、Segger J-Link エミュレータ、および RSK RZ/A1H ボードへの接続が開始されます。
- Eしく接続できた場合は、図に示すような [Debug] ビュー画面を選択します。接続後はエントリポイント(この例では "start.S" の "PowerON_Reset()")でプログラムの実行が一旦中断されます。

図 5-7 [Debug] ビューのユーザターゲット接続

5.2 新規デバッグ構成の作成

別な種類のエミュレータを使用したいなどで、デバッグ構成を追加する場合に、簡単な方法は既存の構成 を複製する方法です。以下の手順で行います。

- 1. [プロジェクト・エクスプローラー]の "Tutorial" プロジェクトをクリックします。[デバッグ構成] ウィ ンドウを開きます(呼び出し方は「5.1. 既存デバッグ構成の変更」参照)。
- [デバッグ構成] ウィンドウで、デバッグ構成 (例: "Tutorial HardwareDebug")を選択し クリックします (現在選択している起動構成をコピーします)。新規デバッグ起動構成 (例: "Tutorial HardwareDebug (1)") が作成されます。設定を確認するために、"名前" テキストボックスに別の名前を 入力して [適用] ボタンをクリックすることもできます。

			-
脉の作成、官理、およい美行			20
° 🕼 🔊 🗎 🗶 📄 🍸 🔻			
	名則(N): Tutorial HardwareDebug 4k		
	📔 メイン 🏁 Debugger 🕒 Startup 🦻 ソース 💷 夫	ŧ通(<u>C</u>)	
	Debug hardware: I-Link ARM	t Device: R7S721001	
	auge landware. Senic runn		
EASE SCIPT	GDB Settings Connection Settings デバッグ・ツ	ノール設定	
GDB Simulator Debugging (BH850	✓ J-Link		^
■ GDB Jindiator Debugging (Kiloso	Туре	USB	~
I lava アプリケーション	J-Link Serial	(Auto)	
Iava アプレット	設定ファイル	\${workspace_loc:\\${ProjName}}\\${LaunchC	Confi
Launch Group (Deprecated)	Script File		
 Renesas GDB Hardware Debugginc 	Low Power Handling	No	~
Tutorial HardwareDebug	✓ Interface		
Tutorial HardwareDebug 4k	Туре	JTAG	~
🖻 Renesas Simulator Debugging (RX,	Speed (kHz)	4000	~
🖫 リモート Java アプリケーション	✓ JTAG Scan Chain		
尋 起動グループ	Multiple Devices	いいえ	~
	IRPre	0	~
		Λ	
< >		前回視答におけ能に言すい。	帝田小小
7 項目のうち 15 項目がフィルターに一致		11回休官した状態に戻9(⊻) 1	◎用(<u>Y</u>)
?		デバッグ(<u>D)</u>	閉じる

図 5-8 選択したデバッグ起動構成の複製

3. デバッグ構成は、5.1節で述べたように作成されます。例えば、インタフェースの "Speed" を "4000" に 変更します。


4. "[local]"の付いた構成を起動してください。"*"印は、その構成がまだどのプロジェクトにも適用されて いないことを示します。[共通] タブでプロジェクト名を指定してください。

💽 デバッグ構成			— 🗆 X
構成の作成、管理、および実行			Ť
 □ ○ □ × □ マ ▼ □ 7/ルタ入力 □ C/C++ アプリケーション □ C/C++ リモート・アプリケーション □ EASE Script 	名前(N): Tutorial HardwareDebug 4k ■ メイン [™] Debugger ▶ Startur: □ 共通(別名保存 ○ ローカル・ファイル(の)	コーソース	奈昭 (の)
3 GDB OpenOCD Debugging 3 GDB Simulator Debugging (RH850) 3 GDB ハードウェア・デバッギング 3 Java アプリケーション 3 Java アプレット	● 共用リアイル(且): お気に入りメニューに表示(B) □ ☆デバッグ	エンコード ● デフォルト - 継承(<u>U</u>) (UTF-8) ○ その他(<u>E</u>) ISO-8859-1	Sector 10 mm
Renesas Simulator Debugging Tutorial HardwareDebug * Tutorial HardwareDebug Renesas Simulator Debugging (RX, RL78)	標準入出力 「コンソールに割り当て(A) (入力に必要) 入力ファイル(E):		
ユリモート Java アプリケーション 14 起動グループ	□ 出力ファイル(L):	リークスペース… ノアイル・システム ワークスペース(<u>W</u>)… 参照(<u>R</u>)…	···
く 17 項目のうち 15 項目がフィルターに一致		前回保管した状態に見	戻す(⊻) 適用(⊻)
(?)		デバッグ	<u>D)</u> 閉じる

図 5-9 特定プロジェクトへの起動構成の適用



5.3 Launch Bar

本節では、V6.0.0 以降で追加された Launch Bar(下図のツールバー)の使い方を説明します。 Launch Bar は e² studio メインウィンドウのツールバーエリア内に表示されます。

この簡単なインタフェースは、デバッガで起動対象とするターゲットに対して、ビルドボタンやデバッグ ボタンが用意されています。



図 5-10 Launch Bar の表示例

Launch Bar は以下の動作を行います:

- ビルド操作: 5 ボタンを押すと、右欄で選択されたデバッグ対象のロードモジュールをビルドします。
- 注意: 「プロジェクト管理」ツールバーにも同様のボタンがありますが、そちらはプロジェクト・エクスプローラ で選択およびアクティブ状態にしたビルド構成に対してビルド操作を行うものです。状況によっては Launch Bar のビルドボタンを押した時とはビルドされるものが異なる場合がありますのでご注意ください。
- デバッグ操作: * ボタンでデバッグの開始/終了(ブレークではなくデバッグセッションの終了)を行います。

Launch Bar の機能が不要な場合は、「ウィンドウ」メニュー → 「設定」→ 「実行/デバッグ」→ 「起動 中」の「Launch Bar」の設定でツールバーやボタンを非表示にすることができます。



5.4 基本的なデバッグ機能

本節では、e² studio がサポートする典型的なデバッグビューを説明します。

- 標準的な GDB デバッグ (Eclipse フレームワークによってサポートされている): ブレークポイント、 式、レジスタ、メモリ、逆アセンブル、変数
- 標準的な GDB デバッグの拡張:イベントポイント、IO レジスタ、トレース

[デバッグ・ツールバー]を開くには、プルダウンメニューボタンをクリックして [デバッグ・ツールバーを表示]を選択します。以下に示すのは、[デバッグ] ビューの便利なツールバーです。



図 5-11 デバッグビューの便利なツールバー

プログラムを実行するには 🕕 ボタンをクリックするか [F8] キーを入力します。

プログラムは、ブレークポイントで、あるいは III ボタンをクリックすることで一時停止します。一時 停止中は以下の操作が可能です。

- 浸 ボタンまたは [F5] キーは、現在実行中のプログラム行にある次の関数呼び出しへステップイン実行 します(関数内に入って1ステップ実行)。
- 🐼 ボタンまたは [F6] キーは、現在実行中のプログラム行にある次の関数呼び出しをステップオーバ 実行します(1 行実行するが関数内には入らない)。
- 🕪 ボタンで、実行を再開します。

デバッグセッションの停止は、選択したデバッグセッション/プロセスを 📕 ボタンで終了するか、選択したプロセスとデバッガを 💦 ボタンで切断します。

- 他に以下のような操作が可能です。
- 🍪 ボタンは、エントリポイントからプログラムを再実行します (🏁 \rightarrow 마 の順にクリックするの と同じ)。
- 🏂 ボタンは、プログラムをパワーオンリセットのエントリポイントにリセットします。

注意: 以降の章で機能について説明するため、ルネサスのウェブサイトから、RZ/A1Hのサンプルコードを ダウンロードしてください。

- 1. ルネサスのウェブサイトから、Starter Kit Sample Code for RZ/A1H をダウンロードします。 https://www.renesas.com/sg/en/software/D6000665.html.
- 2. 第3.1章を参照しインポートしてワークスペースに「RZ_A1H_Tutorial_RSK」プロジェクトをインポート してください。
- プロジェクトのプロパティを開き、画面の左側で、[C/C++ ビルド] → [設定] を選択します。
 [Toolchain] タブで、プロジェクトの GNU ARM Embedded toolchain を選択します。[Apply and Close] をクリックしてください。

圆 プロパティ: RZ_A1H	_Tutorial_RSK		\times
フィルタ入力 ×	設定	<> ▼ <	> ▼ 8
 > リソース > リソース > C/C++ビルド > ツールチェイン・ご ビルド変数 ロギング 環境 設定 > C/C++ 一般 > MCU Renesas QE Task Tags > Validation ビルダー プロジェクト・ネーチ プロジェクト・ネーチ 	Image: State Configuration: HardwareDebug [アクティブ] ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	構成の管理	 A A
フロシェクト参照 実行/デバッグ設定 く ^ 	Additional Tools □ Create Library generator ✓ Create Flash image 適用して閉じる	キャンセル	•

図 5-12 プロジェクトツールチェーンの更新



[Device] タブに切り替えて、使用中のデバイスがボードに合っているかを確認してください。合っていない場合は、インポートしたプロジェクトを右クリックして "Change Device" を選択し、正しいデバイスを選択します。

ファイル(F) 編集(F) ソース(S) リファクタリング(T)	- 🗈	コピー(C)	Ctrl+C
≪ ● * デバッグ(B) × ■ RZ A1H		貼り付け(P)	Ctrl+V
	•	削除(D) ソース	削际
🛯 🕾 RZ_A1H_Tutorial_RSK [HardwareDebu		移動(V)	,
		名前を変更(M)	F2
	24 24	インポート(I) エクスポート(O)	
	Ł	プロジェクトのビルド(B) プロジェクトをクリーンにする 更新(F) プロジェクトを閉じる(S)	F5
		無関係なプロジェクトを閉じる(U)	
		ビルド・ターゲット	>
		インデックス ビルド構成	>
	0	実行(R)	>
	*	デバツグ(D) プロファイル(P)	>
		チーム(E)	>
		比較対象(A)	>
		ローカル履歴から復元(Y)	
		MISRA-C Save build settings report	>
	Г	Change Device	
	\$ **	C/C++ Project Settings	Ctrl+Alt+P
	>>> • • • • • • • • • • • • • • • • • •	C/C++ Jート解析を実行 System Explorer	
	65	Command Prompt	
×		Validate	
リファクタリンク		-	– 🗆 X
Change Device			
Select the new device for RZ_A1H_Tutorial	_RSK		E
Current: R7S72100			
ターゲット・デバイス: R7S721001			
			デバイスのアンロック
⑦ < 戻る(B) X	たへ(<u>)</u>	Ŋ > 終了(<u>F)</u>	キャンセル

図 5-13 デバイスの変更



図 5-14 デバイス変更後に再生成した(変更した)ファイルの比較

入門ガイド

プロジェクトのプロパティを開き、画面の左側で、[C/C++ ビルド] → [設定] を選択します。[ツール設定] タブに切り替えて、[Cross ARM C Linker] → [General] を選択し、コンパイラが生成したスクリプトファイルを、スクリプトファイルの一覧に追加します。[Apply and Close] をクリックしてください。



図 5-15 コンパイラが生成したスクリプトファイルの追加

7. プロジェクトをビルドし、成功したことを確認してください。



e² studio ユーザーズマニュアル

5.4.1 ブレークポイントビュー

ブレークポイントビュー(Breakpoints view)には、プログラムの実行可能な行に設定されたブレークポイントが表示されます。ブレークポイントがデバッグ中にイネーブルになった場合、当該行が実行される前に中断されます。e² studio ではソフトウェアブレークポイントとハードウェアブレークポイントを区別して設定できます。ダブルクリックによりブレークポイントを設定すると、デフォルトのブレークポイント型が適用されます。ハードウェアブレークポイントの場合、ブレークポイント用のハードウェアリソースが残っていないときは設定できません。その場合は、ソフトウェアブレークポイントに変更するようエラーメッセージが表示されます。

ブレークポイントの設定方法は2通りあります。

方法1: エディタ画面上での設定

 ソースコードの左余白を右クリックして [Toggle Software Breakpoint] または [Toggle Hardware Breakpoint] を選択してハードウェアブレークポイント 20 またはソフトウェアブレークポイント 20 を 設定します。

方法2: 「ブレークポイント」ビュー

ブレークポイント設定方法2では、

- ソースコードの左余白を右クリックしてコンテクスト・メニューを開き、[ブレークポイント型] → [e2 studio Breakpoint] (デフォルトのハードウェアブレークポイント) または [ブレークポイント型] → [C/C++ ブレークポイント] (ソフトウェアブレークポイント) を選択します。
- 2. ソースコードの左余白をダブルクリックして、ハードウェアブレークポイントまたはソフトウェアブレ ークポイントを設定します。
- [ビューの表示] → [ブレークポイント] またはアイコン ⁹ をクリックし(あるいはショートカットキー [ALT]+[Shift]+[Q], [B] を使い)、[ブレークポイント] ビューを開いて、設定したハードウェアブレークポ イントまたはソフトウェアブレークポイントを表示します。

ブレークポイントを無効にする際は、選択したブレークポイントのみを無効にするか、あるいはすべての ブレークポイントを無効にするか選択できます。

- [ブレークポイント] ビューで、ブレークポイントを有効または無効に切り替えることができます。無効 になったソフトウェアブレークポイントは白い丸が表示されます。無効になったハードウェアブレーク ポイントは白い丸が表示されます。
- すべてのブレークポイントを無効にするには、[ブレークポイント] ビューにあるアイコン
 をクリックします。斜線を重ねた青い丸(⁴⁰)が [ブレークポイント] ビューとエディタ画面の両方に表示されます。



RENESAS

e² studio ユーザーズマニュアル

5.4.2 式ビュー

式ビュー(Expressions view)では、デバッグ中のグローバル変数、静的変数、ローカル変数の値をモニタで きます。「リアルタイムリフレッシュ」を有効にすると、デバッガ実行中に設定された周期で変数の値が更 新されます。



図 5-17 [式]ビュー

グローバル変数を見るには、

- 1. [ビューの表示] → [式] あるいはアイコン 🙀 をクリックし、[式] ビューを開きます。
- "main.c"の49行目で、グローバル変数(例えば "gPeriodic_Delay")を[式] ビューヘドラッグ&ドロップ します。(または、グローバル変数を右クリックして "監視式を追加(A)…"メニューアイテムを選択し、 [式] ビューに追加します。)
- [式] ビューで、"リアルタイム・リフレッシュ"メニューアイテムを右クリックして選択します。これで、プログラムの動作中、表示した値はリアルタイムに更新されます。"R"は、このグローバル変数がリアルタイムで更新されることを意味します。
- 4. "リアルタイム・リフレッシュ"を無効にするには、"リアルタイム・リフレッシュを無効にする"メニュ 一項目を右クリックして選択します。



5.4.3 レジスタービュー

レジスタービュー(Registers view)は汎用レジスタについての情報を表示します。プログラムを停止する と、変化のあった値を強調表示します。

		1 - 1	≓9 =# #2a 8 ⊡ F
₩レジスター ≈			
名前	値	記述/説明	
🖌 🛗 汎用レジスタ		General Purpose and FPU Register Group	
¹⁰¹⁰ rO	0x0		
1010 r1	0x8		
1919 r2	0x80000000		
⁸⁸⁸ r3	0x0		
			>
ame : r0			
16進:0x0			
10進数:0			
8 進:0 パイナリー・0			
Float:0			
デフォルト:0			
C			>

図 5-18 [レジスター]ビュー

汎用レジスタ "r0" を表示するには、

- 1. [ビューの表示] → [レジスター] あるいはアイコン ¹⁰⁰ をクリックし、[レジスター] ビューを開きます。
- 2. "r0"をクリックすると、他の基数フォーマットで値を表示します。

プログラムを停止すると、前回の表示以降で変化のあった値が [レジスター] ビューの中で強調表示(黄色など)されます。



5.4.4 メモリービュー

メモリービュー(Memory view)では、ユーザは "メモリーモニター" でメモリを表示し編集することができ ます。各モニターは "ベースアドレス" と呼ばれる格納位置によって特定される記憶場所を表します。各メモ リーモニターの中のメモリデータは異なる "メモリレンダリング"で表示することができます。メモリレンダ リングはあらかじめ設定したデータフォーマット (例えば、16 進数、符号付き整数、符号なし整数、 ASCII、イメージなど)です。

変数 (例えば "g_data_block_content") のメモリを表示するには、

- 1. [ビューの表示] → [メモリー] あるいはアイコン 🕕 をクリックし、[メモリー] ビューを開きます。
- 2. アイコン 📌 をクリックし、[モニター・メモリー] ダイアログボックスを開きます。変数 "g_data_block_content"のアドレスを入力します。

モニターするアドレスと式を入力し &g_data_block_content	レマイださい: ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	グロー の整数 されま	バル変数"g >"フォーマ す。	_data_blocl ットのメモ	k_content"は リレンダリン	、"<16進 ?グで表:
⑦ ОК 0 ХЕЛ- №	キャンセル		a 101	19 1010 🚸 📑 🛋	∞4 ⊞ (25) = = ▼	8 -
E_9- F 👘	&g data block content	: 0x2005AF0C	<16 進数の整数	数> 🛛 🗣 新河	規レンダリング	
&g_data_block_content	 アドレス	0 - 3	4 - 7	8 - B	C - F	^
	000000002005AF00	00000000	00000000	00000029	000000FF	
"g data block content"	000000002005AF10	00000000	00000000	00000000	00000001	
っ のメモリーモニターはア	000000002005AF20	00000000	00000000	00000000	00000000	
ドレス	000000002005AF30	00000000	00000000	00000000	00000000	
"&g data block content"	000000002005AF40	00000000	00000000	00000000	00000000	
で指定されます。	000000002005AF50	00000000	00000000	00000000	00000000	
	📙 000000002005AF60	00000000	00000000	00000000	00000000	
	000000002005AF70	00000000	00000000	00000000	00000000	
	000000002005AF70 000000002005AF80	00000000 00000000	00000000 00000000	00000000 00000000	00000000 00000000	
	000000002005AF70 000000002005AF80 000000002005AF90	00000000 00000000 00000000	00000000 00000000 00000000	00000000 00000000 00000000	00000000 00000000 00000000	
	000000002005AF70 000000002005AF80 000000002005AF90 000000002005AFA0	00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000	00000000 00000000 00000000	00000000 00000000 00000000 00000000	
	00000002005AF70 000000002005AF80 000000002005AF90 000000002005AFA0 000000002005AFB0	00000000 00000000 00000000 00000000 0000	00000000 00000000 00000000 00000000 0000	00000000 00000000 00000000 00000000 0000	0000000 0000000 0000000 0000000 0000000	

図 5-19 [メモリー] ビュー (1/2)



変数 "g_data_block_content"用の新しいレンダリングフォーマット(例えば ASCII)を追加するには、

1. 🔷 新規レンダリング... タブをクリックし、"ASCII"を選択してレンダリングを追加します。

これで、"&g_data_block_content <16 進数の整数>" タブの横に新しい "&g_data_block_content < ASCII>" タブが作成されます。

				י 🍪	01 <u>9</u> 1010 🔗 [3 🛃 📑 🛛	11 🔁 🗄 🔻	8
モニター 🔶 🗶 🐐	&g_data_block_content <	:16 進数(の整数>	♣ 新規	しレンダリング	ヷ		
&g_data_block_content	メモリー・モニター: &g data	block co	ontent : 0	x2005AF	DC			
	作成するレンダリングを選択	 そしてくだき	<u>د،</u>					
	固定小数点					1	レンダリング	ブの追加
	イメージ							
	RAW イメージ							
	浮動小数点							
	Traditional							
	16 進							
	ASCI							
	付号付き整数							
	付ちなし釜数							
S ×モリー ∞				\$	1010 1010 🗞	🗋 🛃 🎫		▼ 8 □ □
E_9- 🕂 🖌 💥	&g_data_block_content ·	<16進	&g_dat	a_block_	content :	0x20 🖾	● 新規レン	ンダリング
&g_data_block_content	アドレス	0 - 3	4 - 7	8 - B	C - F			^
	000000002005AF00	0000	0000)000	ÿ00			
	000000002005AF10	0000	0000	0000	0000			
	000000002005AF20	0000	0000	0000	0000			
	000000002005AF30	0000	0000	0000	0000			
	000000002005AF40	0000	0000	0000	0000			
	000000002005AF50	0000	0000	0000	0000			
	000000002005AF60	0000	0000	0000	0000			
	000000002005AF70	0000	0000	0000	0000			
	000000002005AF80	0000	0000	0000	0000			
	00000000000054590	mm	nnn	m	m			

図 5-20 [メモリー] ビュー (2/2)



5.4.5 逆アセンブルビュー

逆アセンブルビュー(Disassembly view)は、ロードしたプログラムのソースコードとアセンブラ命令を混在 して表示します。現在実行中の行は画面上で矢印のマーカで強調表示されます。逆アセンブルビューでは、 アセンブラ命令へのブレークポイントの設定、ブレークポイントの有効化/無効化、逆アセンブル命令のス テップ実行、プログラムの特定の命令へのジャンプが可能です。



図 5-21 [逆アセンブル] ビュー

混合モードでCコードとアセンブリコードの両方を表示するには、

- [ビューの表示] → [逆アセンブル] あるいはアイコン = をクリックし、[逆アセンブル] ビューを開きます。
- アイコン なクリックして、アセンブリソースとCソース(アクティブなデバッグコンテクスト)の リンクを有効にします。
- [逆アセンブル] ビューのアドレス列で右クリックし、"オペコードを表示する"と"関数オフセットを表示する"を選択します。
- 4. コンテクスト・メニューを使用すればエディタ内でSource Addressesを有効にすることができます。



図 5-22 ソースアドレスメニュー



5.4.6 変数ビュー

変数ビュー(Variables view)は、現在実行中のスコープ内で表示可能な全てのローカル変数を表示します。 グローバル変数、あるいは、現在実行中のスコープ外の外部変数を見るには、'式'ビューを参照してください。

141		^	名前	型	値	Name : strdata
143	<pre>* Function Name: main</pre>		🔸 🥔 strdata	char [32]	0x2090ffc4	Details :'\0' <repea< td=""></repea<>
L52 20042ae4			> ⇒ pmptr	unsigned char *	0x20054b3c <	10 准数:546373572
154 20042af0	<pre>char strdata[32] = "";</pre>					16 進:0x2090ffc4
155 20042b50 156	uint8_t * pmptr = &g_rgb888_logo[6);				バイナリー:100000100100 8 進:04044177704
157 20042b5c	<pre>R_RIIC_rza1h_rsk_init();</pre>					
158	/* Initialise SPI */					
160 20042560 161	R_SPI_Init();					
162	/* Initialise the PMod Colour LCD	disţ				

図 5-23 [変数]ビュー

ローカル変数(例えば、関数 "main()" の "strdata") を見るには、

- 1. [ビューの表示] → [変数] またはアイコン 💴 をクリックし、[変数] ビューを開きます。
- 2. ステップ実行で関数 "main()" の中に入ると、[変数]ビュー内にローカル変数 "strdata" の値が表示されます。
 - 注意: コンパイラやリンカの最適化処理により、変数の実体がなくなる(変数に対応するメモリを保持する必要がな い場合や、変数や変数の演算に使用するレジスタが他の用途にも使われる場合など)と、変数ビューに値が表 示されないことがあります。その際は逆アセンブリビューで変数に対応するレジスタやメモリがどれかを確 認し、レジスタビュー等でその値を直接見てください。



e² studio ユーザーズマニュアル

5.4.7 イベントポイントビュー

イベントは、プログラム実行中にブレークあるいはトレース機能を実行するために設定された条件の組み 合わせです。ユーザはイベントポイントビュー(Eventpoints view)で、異なる種類の定義されたイベント、た とえば、トレース開始、トレース終了、トレース・レコード、イベント・ブレーク、実行前 PC ブレーク、 タイマー開始、およびタイマー終了、などを設定、表示することができます。

設定できるイベント数や設定条件は MCU によって異なります。以下に挙げる 2 種類のイベントがあります。

- 実行アドレス:エミュレータは CPU が特定のアドレスの命令を実行しようとしたことを検出します。これが"実行前 PC"ブレーク(例えば、イベントにより、条件は指定アドレスで命令の実行直前に成立する)、あるいは他のイベント(例えば、イベントにより、条件は指定アドレスで命令の実行直後に成立する)となります。
- データ・アクセス:エミュレータは指定された条件での指定アドレスあるいは指定アドレス範囲へのアクセスを検出します。これにより、アドレスとデータを組み合わせた条件を設定することができます。

イベントの組み合わせは(OR, AND(およびその組み合わせ),およびシーケンシャル)は2つ以上のイ ベントに使用できます。

変数をアクセスする条件(例えば、g_adc_result が新しい値として割り当てられたとき)で、グローバル 変数にイベント・ブレークを設定するには、

- [ビューの表示] → [イベントポイント] あるいはアイコン をクリックし、[イベントポイント] ビュー を開きます。
- "イベント・ブレーク"オプションをダブルクリックし、[編集イベント・ブレーク]ダイアログボックス を開きます。
- 3. [追加...] ボタンをクリックし、以下の操作を行います。

💡 編集 イベ	ント・ブレーク						×
				ł	組み合わせ条(牛: OR	~ \$ t
タイプ	アドレス	データ	パス・カウント	917-	ハンドル番号	イイメロ	
追加 編	扁集 削除	PC: 0/0 OA: 0/4 す^	ኣて: 0/4				
					OK		キャンヤル

図 5-24 イベント・ブレークの編集

- 4. イベントポイントの種類に"Data Access"を選択します。
- 5. [アドレス設定] タブに進み、アイコン ----- をクリックしてシンボル "g_adc_result" を検索します。(こ のグローバル変数のアドレスは "&g_adc_result" です。)

RENESAS

6. 次に、[データ・アクセス設定] タブに切り替え、[データ・アクセス設定] に "Write" を選択して、[OK] をクリックしてください。

			6	ーベントオ	ポイントの追加		;
				イベントポ	イント・タイプ:	Data Access	\sim
イベント	ポイントの追加		×	アドレス設	定 データ・ア	クセス設定	ו
イベントオ	ペイント・タイプ: Dat	a Access	~	データ・ア リード/ライ	クセス設定: 'ト:	Write	~
	変正」テータ・アクセ	人設定		アクセス・	サイス:	Not Spec	cified ~
♀! 編集 イ⁄	ジト・ブレーク	<u> </u>			組み合わせ条件	#: OR	
タイプ	アドレス	データ	パス・カウン	ト タイマー	ハンドル番号	1	
<mark>⊡ ≜</mark> OR	&g_adc_result	Write All]
追加 着	編集 削除 p	C: 0/0 OA: 1/4 す	べて: 1/4				
					OK		キャンセル

図 5-25 イベントポイントの追加

7. [イベントポイント] ビューで、イベント・ブレークが "g_adc_result" に設定されて有効になっていることを確認してください。プログラムを最初から実行するためにリセットします。

🕯 start.S 🛛 🖻 main.c	Initsct.S ■ adc.c ≅		● イベントポイント 🛛	F 🔌 🗶 🎉 📔	🕓 PC: 0/6	OA: 1/4	8 🖻	- 8
297 298 299 200402e4 300 301 302	<pre>{ /* Clear ADC complete flag */ ADC.ADCSR &= 0x7FFFu; } /* Read the result register for AN7 */</pre>	^	9イブ □●トレース開始 ■ トレース停止 ■ トレース・レコード マーク・レース・レコード	7522	データ	パス	タイ	ハン
303 20040308 304 305	<pre>g_adc_result = ((uint32_t)ADC.ADDRH >> /* Check if the upper and lower limits</pre>	4u);	 ✓ OR ✓ Ø7マー開始 	&g_adc_result (0x2005aefc)	Write All			6
306 20040330 ⊖ 307	<pre>if ((uint32_t)0u == g_adc_result) {</pre>	-	□●タイマー停止					
308 20040344 309 210	<pre>g_adc_complete_flag = 0x80u; }</pre>		<		_			>
310	<pre></pre>	>	プロジェクト 保管されたテンプ	プレート				

図 5-26 イベント・ブレークの実行

図 5-33 は、g_adc_result に新しい値が割り当てられたときにプログラムの 303 行目で実行が停止する様子 を示します。 5.4.8 IO Registers ビュー

IO レジスタは Special Function Register (SFR) のことです。IO Registers ビュー(IO レジスタビュー)は、 ターゲット専用の IO ファイルで定義された全レジスタセットの名前、アドレス、値(16 進数および 2 進 数)を表示します。ユーザは、[選択されたレジスター] に必要な IO レジスタを選択して追加することによっ て、IO レジスタビューをカスタマイズすることができます。

名前	値	Value (Bin)		アドレス	アクセス	^
> MMC						
> MTU2						
✓ OSTM0						
OSTMOCI	MP 0x00000000	0000000000000	0000000	0xfcfec000	RW	
OSTMOCI	0x0000000	0000000000000	0000000	0xfcfec004	RW	
OSTMOTE	0x00		00000000	0xfcfec010	RW	
OSTMOTS	0x00		00000000	0xfcfec014	RW	
OSTM0TT	0x00		00000000	0xfcfec018	RW	
OSTM0C	0x00		00000000	0xfcfec020	RW	
> OSTM1						~
すべてのレジスター ■ IO Registers ¤	れていにレンスタ	/	ŧ.	i 🗢 🔕 🎓 🖸	, (=) (1) (1	2 8 - E
名前	值	Value (Bin)	アドレス	アクセス		
> • OSTM0						

図 5-27 [IO Registers] ビュー

選択した IO レジスタ (例えば、OSTM0 の OSTM0CMP) を見るには、

- IO Registersビューが表示されていない場合は、「ウィンドウ」メニューの「ビューの表示」から、「その他」のカテゴリ「デバッグ」の中から「IO Registers」を選択してください。
- 2. [すべてのレジスター] タブの [IO Registers] ビューに、[OSTM0] を表示します。OSTM0 IOレジスタの 一覧を展開してください。
- 3. "OSTMOCMP"を [選択されたレジスター] ヘドラッグ&ドロップしてください。IOレジスタの横にある 緑色のドット ● は、選択されているレジスタの状態を示します。
- 4. [選択されたレジスター] タブに切り替えて、"OSTM0" IOレジスタの "OSTM0CMP" を表示します。

「選択されたレジスター」タブには、指定したレジスタだけを表示させることができます。必要なレジス タを複数まとめて表示するのに便利です。「選択されたレジスター」を使用すると必要な IO レジスタの値だ けをロードするのでデバッガ動作が重くなるのを防ぐこともできます。

5.4.9 トレースビュー

トレースとは、ユーザプログラム実行中、1サイクルごとのバス情報をトレースメモリから取得すること を意味します。取得されたトレース情報はトレースビューに表示されます。それによりユーザはプログラム の実行を追跡し、問題が発生した箇所を探すことができます。

トレースバッファは有限(1~32M バイトのサイズ)なため、バッファが一杯になると、最も古いトレース データを新しいデータで上書きします。

RZ/A1H では、トレースを完全にサポートしていません。



5.5 デバッグ手順(RZ/A2Mの場合)

本節では、RZ/A2M のデバッグ手順について、Segger J-Link エミュレータと RZ/A2M Evaluation Board Kit をターゲットボードとする"Loader"プロジェクト、"Application"プロジェクト(3.2.1 節)を例として説明します。

- 1. [プロジェクト・エクスプローラー] の "Loader" プロジェクトをクリックします。
- 2. (エラー!参照元が見つかりません。節)を参考にビルドし、"Loader.elf"ファイルを作成しておきます。
- 3. Launch Barの[Launch Configuration]にて "Loader HardwareDebug"を選択します。

|--|

図 5-28 [Launch Configuration]の選択

Launch Barの ボタンをクリックし、ターゲットボートとの接続、およびターゲットボードへの"Loader.elf"ダウンロードを実行します。



図 5-29 ダウンロード実行後の[コンソール]ビュー

- 5. Launch Barの 📕 ボタンをクリックして終了します。
- 6. 次に[プロジェクト・エクスプローラー]の "Application" プロジェクトをクリックします。
- 7. (**エラー! 参照元が見つかりません。**節)を参考にビルドし、"Application.elf"ファイルを作成しておきます。
- 8. Launch Barの[Launch Configuration]にて "Application HardwareDebug"を選択します。



図 5-30 [Launch Configuration]の選択

- 9. Launch Barの * ボタンをクリックし、ターゲットボートとの接続、およびターゲットボードへの"Application.elf"ダウンロードを実行します。
- 10. ダウンロード完了後、 🍪 ボタンをクリックし、"Loader.elf"のエントリポイントからプログラムを再実行 させます。

6. ヘルプ

ヘルプシステムによって、ユーザはワークベンチ内の各ヘルプウィンドウやヘルプ画面から、ヘルプドキュメントのブラウズ、検索、ブックマーク、印刷が可能です。また、ヘルプメニューから e² studio 専用のオンラインフォーラムにアクセスできます。

[ヘルプ] をクリックしてヘルプメニューをプルダウンしてください。

虚	ġ	69 (5)	R 90 43		?	6	\sim
e^2 studio (CDL)T(A)	IAR Embedded Workbench plugin	 更新の検査 新規ソフトウェアのインストール Bapasas a2 studio foodback 	RA Helpdesk RenesasRulz コミュニティー・フォーラム Renesas ツールチェーンの追加 Perform Setup Tasks	キー・アシスト(K) 虎の巻の表示(C)	 ヘルプ目次(H) 検索(E) ダイナミック・ヘルプ(D) 	ようこそ(W)	レプ(圧)
	manager		A (3)	Ctrl+シフト+L	(2)	(1)	

図 6-1 ヘルプメニュー

ヘルプメニューの使い方

- (1) [ようこそ] をクリックすると、 e^2 studioの概要、 e^2 studioチュートリアルとサンプルプログラムへのリンク、リリースノートを表示します。
- (2) [ヘルプ目次]をクリックすると、新たにヘルプウィンドウが開きヘルプを検索できます。
- (3) [RenesasRulzコミュニティー・フォーラム] をクリックすると、e² studio関連のディスカッション参加 型オンラインフォーラムにアクセスします。インターネット接続が必要です。

改訂記録 e² studio 統合開発環境ユーザーズマニュアル 入門ガイド

Rev.	発行日			
		ページ	ポイント	
1.00	Sep.13.19	_	初版発行	
1.01	Mar.3.20	22-33	「RZ/A2M のプロジェクトの作成」を追加	
		67	「デバッグ手順(RZ/A2M の場合)」を追加	
1.02	Sep.24.20	_	e ² studio 7.8.0, 2020-04 に対応	
1.03	Jun.21.23	_	e ² studio 2023-04 に対応	

e ² studio 統合	 開発環境ユーザー	ズマニュアル			
入門ガイド					
登行在日口	2023年6日21日	Pov 1 03			
元11千万口	2023-07210	Nev.1.05			
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)		

e² studio

