

RZ/A1Hグループ

Renesas Starter Kit+ チュートリアルマニュアル

e² studio版

ルネサスマイクロプロセッサ
RZファミリ／RZ/Aシリーズ

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

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 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に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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。
5. 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）、交通用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じて、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 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、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害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。
8. 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。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている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。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

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その総株主の議決権の過半数を直接または間接に保有する会社をいいます。

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

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. 未使用端子の処理

【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】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. リザーブアドレス（予約領域）のアクセス禁止

【注意】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、将来の機能拡張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。

4. クロックについて

【注意】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。

5. 製品間の相違について

【注意】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っていると、内部 ROM、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、ノイズ輻射量などが異なる場合があります。型名が異なる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。

CAUTION

With reference to Directive 2014/30/EU Article 2, clause 2 (e) this is a custom built evaluation kit destined for professionals to be used solely at research and development facilities for such purposes. This equipment can cause radio frequency noise when used. In such cases, the user/operator of the equipment may be required to take appropriate countermeasures under his responsibility.

CAUTION

This equipment should be handled like a CMOS semiconductor device. The user must take all precautions to avoid build-up of static electricity while working with this equipment. All test and measurement tool including the workbench must be grounded. The user/operator must be grounded using the wrist strap. The connectors and/or device pins should not be touched with bare hands.

EEDT-ST-004-10

For customers in the European Union only

The WEEE (Waste Electrical and Electronic Equipment) regulations put responsibilities on producers for the collection and recycling or disposal of electrical and electronic waste. Return of WEEE under these regulations is applicable in the European Union only. This equipment (including all accessories) is not intended for household use. After use the equipment cannot be disposed of as household waste, and the WEEE must be treated, recycled and disposed of in an environmentally sound manner. Renesas Electronics Europe GmbH can take back end of life equipment, register for this service at <http://www.renesas.eu/weee>

このマニュアルの使い方

1. 目的と対象者

このマニュアルは、RSK+プラットフォーム用ソフトウェアを開発し、デバッグするために e² studio を使用する方法を理解していただくためのマニュアルです。様々な周辺装置を使用して、RSK+プラットフォーム上のサンプルコードを設計するユーザを対象としています。

このマニュアルは、段階的に e² studio 中のプロジェクトをロードし、デバッグする指示を含みますが、RSK+プラットフォーム上のソフトウェア開発のガイドではありません。RZ/A1H マイクロプロセッサの動作に関する詳細は、ハードウェアマニュアルおよび、サンプルプログラムを参照してください。

このマニュアルを使用する場合、注意事項を十分確認の上、使用してください。注意事項は、各章の本文中、各章の最後、注意事項の章に記載しています。

改訂記録は旧版の記載内容に対して訂正または追加した主な箇所をまとめたものです。改訂内容すべてを記録したものではありません。詳細は、このマニュアルの本文でご確認ください。

次のドキュメントが RZ/A1H グループ製品に用意されています。ドキュメントは最新版を使用してください。最新版はルネサスエレクトロニクスのホームページに掲載されています。

ドキュメントの種類	記載内容	資料名	資料番号
クイックスタートガイド	RSK+ および開発環境のセットアップ方法とサンプルコードの実行方法を記載した A4 紙一枚の簡単なセットアップガイド	RSK+RZA1H クイックスタートガイド	R20UT3006JG
ユーザーズマニュアル	RSK+ ハードウェア仕様の説明	RSK+RZA1H ユーザーズマニュアル	R20UT3007JG
チュートリアルマニュアル	RSK+ および開発環境のセットアップ方法、サンプルコードの実行とデバッグ方法の説明	RSK+RZA1H チュートリアルマニュアル	R20UT3008JG (本マニュアル)
回路図	RSK+RZA1H の回路図詳細	RSK+RZA1H 回路図	R20UT2586EG
ユーザーズマニュアル ハードウェア編	RZ/A1 グループ製品のハードウェアの仕様（ピン配置、メモリマップ、周辺機能の仕様、電気的特性、タイミング）と動作説明	RZ/A1H グループ、 RZ/A1M グループ ユーザーズマニュアル ハードウェア編	R01UH0403JJ

2. 略語および略称の説明

略語／略称	英語名	日本語名
ACIA	Asynchronous Communication Interface Adapter	調歩同期式通信アダプタ
bps	bits per second	転送速度を表す単位、ビット/秒
CRC	Cyclic Redundancy Check	巡回冗長検査
DMA	Direct Memory Access	CPU の命令を介さずに直接データ転送を行う方式
DMAC	Direct Memory Access Controller	DMA を行うコントローラ
GSM	Global System for Mobile Communications	FDD-TDMA の第二世代携帯電話の方式
Hi-Z	High Impedance	回路が電氣的に接続されていない状態
IEBus	Inter Equipment bus	—
I/O	Input/Output	入出力
IrDA	Infrared Data Association	赤外線通信の業界団体または規格
LSB	Least Significant Bit	最下位ビット
MSB	Most Significant Bit	最上位ビット
NC	Non-Connection	未接続
PLL	Phase Locked Loop	位相同期回路
PWM	Pulse Width Modulation	パルス幅変調
SFR	Special Function Registers	周辺機能を制御するためのレジスタ
SIM	Subscriber Identity Module	ISO/IEC 7816 規定の接触型 IC カード
UART	Universal Asynchronous Receiver/Transmitter	調歩同期式シリアルインタフェース
VCO	Voltage Controlled Oscillator	電圧制御発振器

略語／略称	英語名	日本語名
ADC	Analog-to-Digital Converter	A/D 変換器
e2 studio	Renesas Eclipse Embedded Studio (Integrated Debugging Environment)	ルネサス統合開発環境 Eclipse Embedded Studio
EMC	Electromagnetic Compatibility	電磁両立性
ESD	Electrostatic Discharge	静電気放電
I2C, IIC	Philips™ Inter-Integrated Circuit (Serial bus)	フィリップス社開発の 2 線双方向シリアルバス
IRQ	Interrupt Request	割り込み要求
J-LINK Lite	SEGGER's compact JTAG emulator	SEGGER 製 JTAG エミュレータ (小型版)
LCD	Liquid Crystal Display	液晶ディスプレイ
LED	Light Emitting Diode	発光ダイオード
MCU	Micro-controller Unit	マイクロコントローラユニット
N/A	Not Applicable	該当なし
PC	Personal Computer	パーソナルコンピュータ
QSPI	Quad Serial Peripheral Interface	クアッドシリアルペリフェラルインタフェース
RSK	Renesas Starter Kit	ルネサススタータキット
RSK+	Renesas Starter Kit+ (denotes extra functionality over standard RSK)	ルネサススタータキットプラス (標準の RSK に対し追加機能を有するもの)
SPI	Serial Peripheral Interface	シリアルペリフェラルインタフェース
USB	Universal Serial Bus	ユニバーサルシリアルバス

すべての商標および登録商標は、それぞれの所有者に帰属します。

目次

1. 概要.....	9
1.1 目的.....	9
1.2 特徴.....	9
2. はじめに.....	10
2.1 ソースコードについて.....	10
3. チュートリアルプロジェクトワークスペース.....	11
3.1 はじめに.....	11
3.2 ジャンパとスイッチの設定.....	11
3.3 e ² studioの起動とサンプルコードの読込.....	11
3.4 ビルド構成とデバッグ・セッション.....	14
3.4.1 ビルド構成.....	14
3.4.2 デバッグ構成.....	14
4. チュートリアルプログラムの確認.....	17
4.1 メイン関数.....	17
5. 追加情報.....	21

1. 概要

1.1 目的

本 RSK+はルネサスマイクロプロセッサ用の評価ツールです。本マニュアルは、コードのダウンロードや基本的なデバッグ操作について説明しています。

1.2 特徴

RSK+は、次の機能を提供します。

- ルネサスマイクロプロセッサのプログラミング
- ユーザコードのデバッグ
- スイッチ、LED、ポテンショメータ等のユーザ回路
- サンプルアプリケーション
- 周辺機能初期化のサンプル

RSK+ボードはマイクプロセッサの動作に必要な回路を全て備えています。

2. はじめに

本マニュアルは Renesas Starter Kit+ (RSK+) をご使用の際、最も多く寄せられる質問に対し、チュートリアル形式でお答えするものです。チュートリアルマニュアルでは以下の項目について説明しています。

- RSK+でプログラムをコンパイル、リンク、ダウンロードおよび実行する方法は？
- 組み込みアプリケーションの構築方法は？
- ルネサスツールの使用方法は？

本マニュアルで引用されたファイルはチュートリアルを進めていく過程でインポートウィザードを使用してインストールされます。本チュートリアルマニュアルの使用例はクイックスタートガイドに記載のインストールが完了していることを前提としています。使用設定の詳細については、クイックスタートガイドを参照ください。

チュートリアルマニュアルは RSK+の使用方法の説明を目的とするものであり、e² studio の環境、コンパイラまたは J-Link LITE エミュレータの入門書ではありません。これらに関する詳細情報は各関連マニュアルを参照してください。

2.1 ソースコードについて

本マニュアル中のソースコードはコードジェネレータにて生成されるため、実際の操作画面上のソースコードのライン番号とマニュアル記載のソースコードのライン番号が異なる場合があります。同様にソースコードのアドレスも、マニュアルと操作画面で違う場合がありますが、マニュアルに記載されている内容と機能的違いはございません。

3. チュートリアルプロジェクトワークスペース

3.1 はじめに

e² studio はルネサス統合開発ツールであり、ユーザはこれを使用してルネサス RZ ファミリマイコンのソフトウェアプロジェクトを作成、コンパイル、プログラム、デバッグすることができます。e² studio は Renesas Starter Kit 製品をインストールする際、一緒にインストールされます。本マニュアルでは、チュートリアルコードの作成およびデバッグに必要な作業を段階的に説明します。

3.2 ジャンパとスイッチの設定

ジャンパとスイッチには、下記の表の設定になっていることを確認してください。

ジャンパ	設定	スイッチ	設定	スイッチ	設定
JP11	1 - 2	SW4-1	OFF	SW6-1	OFF
JP12	1 - 2	SW4-2	OFF	SW6-2	ON
JP18	1 - 2	SW4-3	OFF	SW6-3	OFF
JP21	1 - 2	SW4-4	OFF	SW6-4	ON
PWR_SEL	2 - 3	SW4-5	OFF	SW6-5	ON
--	--	SW4-6	OFF	SW6-6	ON
--	--	SW4-7	OFF	--	--
--	--	SW4-8	OFF	--	--

3.3 e² studio の起動とサンプルコードの読込

- e² studio を起動するには、
スタートメニュー → すべてのプログラム → Renesas Electronics e² studio 3.1 → Renesas e² studio を選択ください。最初にワークスペース・ランチャーダイアログが表示されます。
- 「参照」をクリックして、使用するワークスペースを指定してください。新規に作成する場合は「新しいフォルダの作成」を実行し、「OK」をクリックしてください。

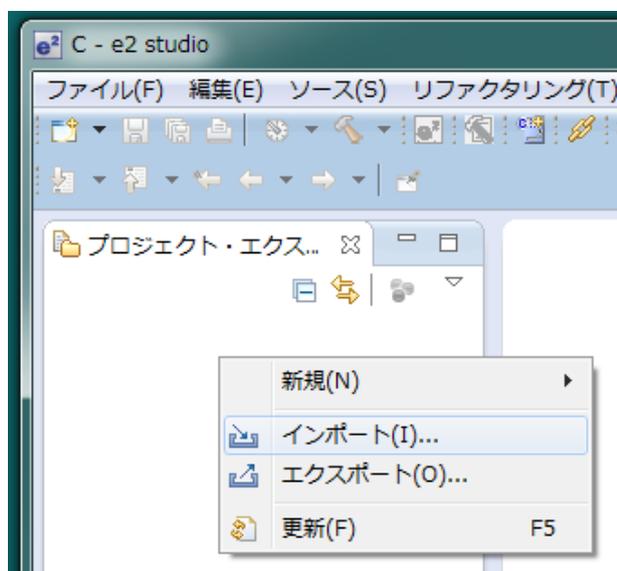


注：ワークスペースフォルダにはプロジェクトファイルが含まれている必要はありませんが、ワークスペースはツール構成を含む複数プロジェクトをグループ化することができます。プロジェクトは、この位置から参照することが可能です。また、このディレクトリの下にプロジェクトを格納することも可能です。

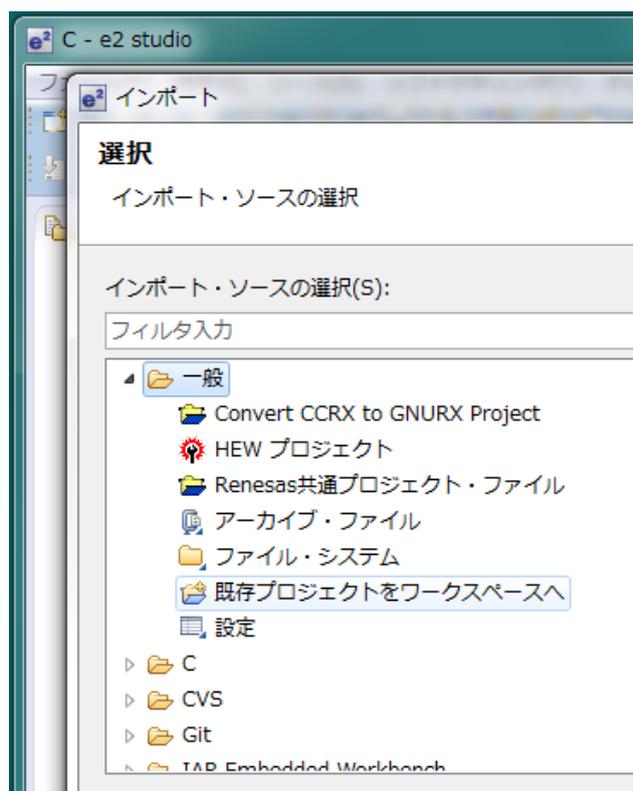
- e² studio は起動時に、右図に示す「ようこそ」タブが表示されます。
- 「ようこそ」タブ横の「×」をクリックし、このタブを閉じてください。



- 「プロジェクト・エクスプローラー」上で右クリックし「インポート」を選択してください。



- インポートダイアログが表示されます。
「一般」フォルダアイコンを展開して「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。



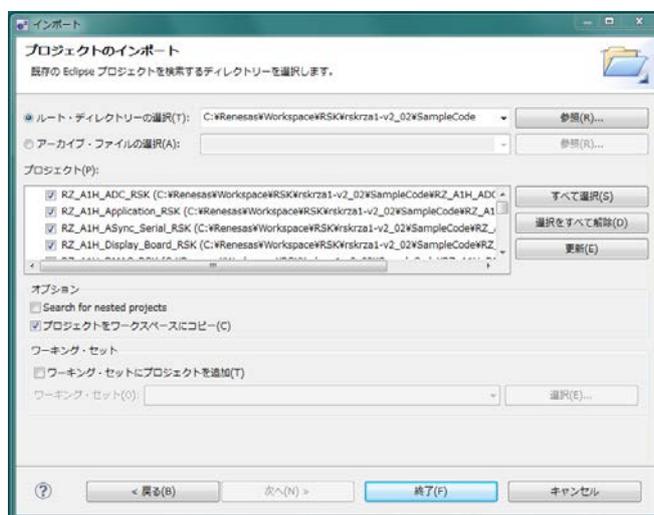
- インポートするプロジェクトを指定できるようになります。「参照」ボタンをクリックし、以下に示すディレクトリを指定します。

C:\Renesas\Workspace\RSK\RSK+RZA1_E2

- オプション「プロジェクトをワークスペースにコピー」がチェックされていることを確認してください。

注：このボックスにチェックをすると、各プロジェクトをインストールされた場所からコピーします。これはインストール時の初期状態を保存しておけるため、このオプションを選択しておくことを推奨します。

- 「終了」をクリックすると、プロジェクトのコピーが始まります。



3.4 ビルド構成とデバッグ・セッション

3.4.1 ビルド構成

e² studio ワークスペースは、「HardwareDebug」と「Release」の2つのビルド構成で生成されます。

HardwareDebug

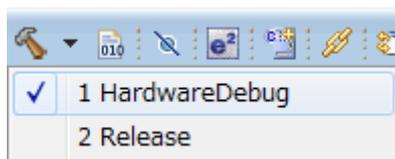
このデフォルトのビルドモードは、全ての最適化オプションがオフの状態、全デバッグ情報を出力する設定です。このモードではCコードが逐次実行されるため、コード開発に最適の構成です。

「HardwareDebug」ビルド構成はチュートリアルプログラムのために提供され、RAMに直接コードをロードするように設定されています。

- 「チュートリアル」の最上位フォルダをもう一度クリックし、ビルドボタン（ハンマーアイコン）横の▼をクリックして、「HardwareDebug」オプションを選択してください。



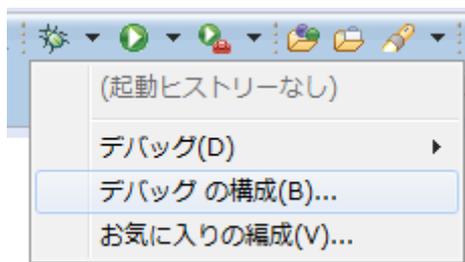
e² studio がビルドを開始します。



- ビルド過程は、e² studio のコンソールウィンドウに出力されます。

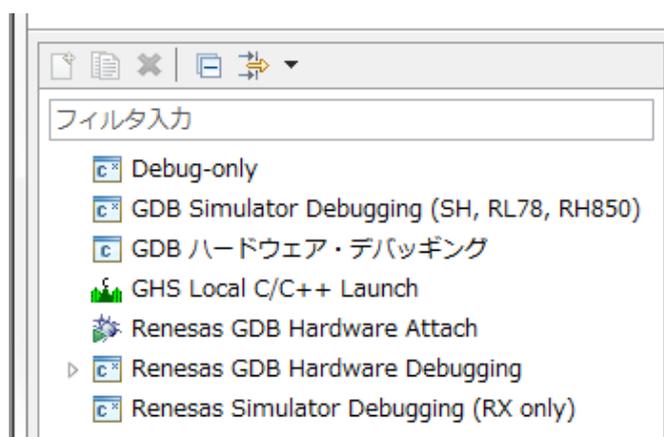
3.4.2 デバッグ構成

- デバッグボタン（バグアイコン）をクリックして、「デバッグの構成」を選択してください。



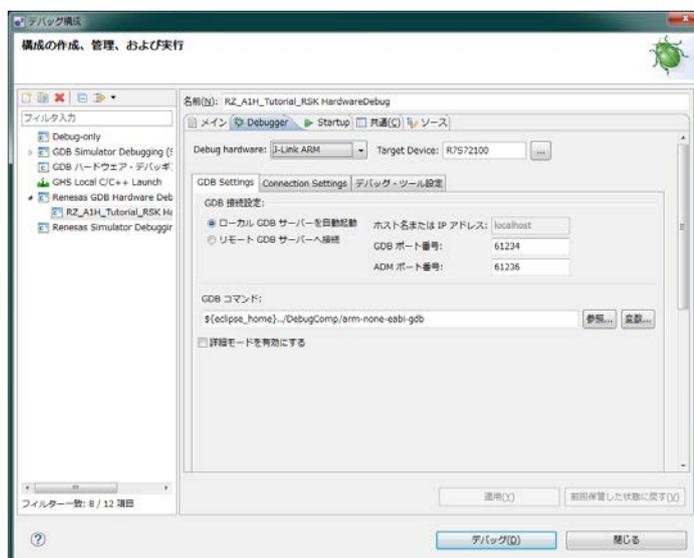
- 「デバッグ構成」ダイアログが表示されたら、「Renesas GDB Hardware Debugging」横の[] ボタンをクリックして展開表示してください。
- デフォルトでは、e² studio は既存のビルドモードのデバッグ構成を生成します。RSK+RZA1H に付属しているプロジェクトは、すぐに使用できる状態のデバッグ構成を提供しています。

注：手で新しい e² studio デバッグ構成を作成するには、「Renesas GDB Hardware Debugging」をクリックして「新規」ボタンをクリックします。

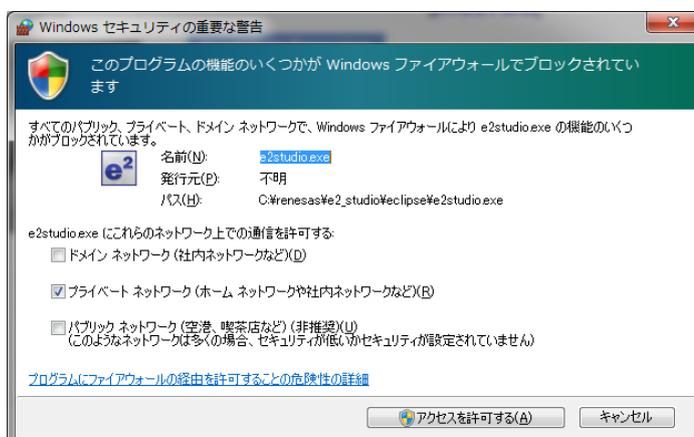


- デバッグ構成コントロールページが開くので、各タブの設定（下記）を確認してください。
- 「Debugger」タブで、「Debugger hardware」オプションが「J-Link ARM」に、「Target Device」が「R7S72100」に設定されていることを確認してください。

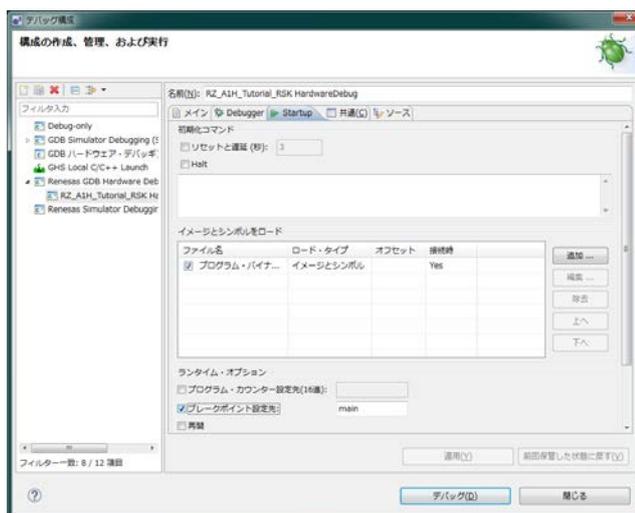
注：どの設定も変更しないでください。



- セキュリティダイアログが表示され、Windows ファイアウォールが Eclipse プラットフォームの一部の機能をブロックしたことを示す場合があります。
- 「e2studio.exe にこれらのネットワーク上での通信を許可する」というテキストの下に、「プライベートネットワーク（ホームネットワークや社内ネットワークなど）」をチェックし「アクセスを許可する」をクリックしてください。



- 「Startup」タブを選択し、「ランタイム・オプション」の「ブレークポイント設定先」が「main」に設定されていることを確認してください。



- 「デバッグ」をクリックしてください。



- コードをダウンロードする前に、ダイアログが表示され、「デバッグパースペクティブ」への切り替えるかを聞かれます。切り替える場合は「はい」をクリックしてください。「常にこの設定を使用する」をチェックすれば、以降このダイアログは表示されません。
- e² studio は、デバッグ用に最適化された新規パースペクティブをロードします。
- もとの「C/C++」パースペクティブに変更するには、「ウィンドウ>パースペクティブを開く>その他」を選択してください。
- 「パースペクティブを開く」ダイアログが表示されるので、希望のパースペクティブを選択し、その後「OK」をクリックしてください。
- また、右画面に示す「C」ボタンをクリックすれば、C/C++パースペクティブになります。



4. チュートリアルプログラムの確認

この章では、e² studio のチュートリアルコードの各セクションおよび基礎的なデバッグ機能を説明します。

4.1 メイン関数

この章では、main () 関数から呼び出されるプログラムコードがどのように動作するかを説明します。

- 第3章で説明したチュートリアルプログラムのデバッグセッションから開始します。エミュレータが接続されていれば、右画面に示すようにプログラムは main () 関数の最初の行で停止しています。
- main()関数から flashed()関数を呼び出している行をクリックし、カーソルを合わせてください。右クリックして、「指定行まで実行」を選択するとその行まで実行します。R_LCD_Init()関数はLCDを初期化し、Display_LCD()関数はLCDの一番上に「RSK+RZA1H」を、次の行に「Tutorial Sample」を出力します。

```

150 int_t main(void)
151 {
152     char strdata[32] = "";
153     uint8_t * pmptr = &g_rgb888_logo[0];
154
155     R_RIIC_rza1h_rsk_init();
156
157     /* Initialise SPI */
158     R_SPI_Init();
159
160     /* Initialise the PMod Colour LCD display */
161     R_LCD_Init();
162
163     /* Initialise direct connected LED */
164     R_LED_Init();
165
166     /* Initialise board switches (SW1,SW2,SW3) */
167     init_switches();
168
169     /* Display 'Renesas' Logo at bottom of screen */
170     display_image (pmptr, 128, 24, 0, 104);
171     display_on();
172
173     display_lcd(0, (uint8_t *)" RSK+RZA1H");
174     display_lcd(2, (uint8_t *)" Tutorial Sample");
175
176     flashed();
177
178     clear_display_area();
179
180     static_test();

```

- ブレークポイント欄をダブルクリックしてstatic_test()関数にブレークポイントを設定してください。

```

176 flashed();
177
178 clear_display_area();
179
180 static_test();

```

- flashed()関数にステップインするための「ステップイン」ボタンをクリックしてください。



- プログラムを再開するため「再開」ボタンをクリックしてください。
- `flashled()`関数のプログラム実行を開始します。この関数は、定期的にユーザスイッチをポーリングしており、ユーザスイッチが押されるまで、LEDを200回点滅させます。点滅回数は、「`loopcount`」に変数指定されています。

```

78 void flashled(void)
79 20040668 {
80 char strdata[256];
81 20040674 uint16_t loopcount = 200;
82 uint32_t delay;
83
84 2004067c display_lcd(5, (uint8_t *)" LED Flashing");
85 2004068c display_lcd(6, (uint8_t *)" Press SW1,SW2,SW3");
86 2004069c display_lcd(7, (uint8_t *)" or wait 200 flashes");
87 200406ac display_lcd(8, (uint8_t *)" to continue demo");
88
89 200406bc R_LED_Off(LED_ALL);
90
91 200406c4 while(loopcount)
92 {
93 200406c8 sprintf(strdata, " Countdown %d ",--loopcount);
94
95 /* Display the application name on the PMOD LCD */
96 200406f0 display_lcd(10, (uint8_t *)strdata);
97
98 20040700 delay = 10000000u;
99 2004070c while (--delay)
100 {
101 20040710 __asm__("nop");
102 }
103
104 /* Toggles the LEDs after a specific delay. */
105 2004072c R_LED_Toggle(LED_ALL);
106
107
108 20040734 if(g_switch_press_flg)
109 {
110 20040748 g_switch_press_flg = 0;
111 20040758 loopcount = 0;
112 }
113 }
114
115 2004076c R_LED_Off(LED_ALL);
116 20040774 }

```

- プログラムカウンタは、`static_test()`関数で停止します。
- 「ステップイン」ボタンをクリックするか、「F5」キーを押して、関数にステップインしてください。

```

176 20042b68 flashled();
177
178 20042b6c clear_display_area();
179
180 20042b70 static_test();

```

- 「F7」キーまたは「ステップ・リターン」ボタン  を押すと、static_test()関数が実行されステップ・リターンします。
- static_test()関数が実行されると、LCDの一番下の行が一文字ずつ「STATIC」から「TESTTEST」に更新されることを確認できます。
- すべての文字が更新された後、再度「STATIC」の表示に戻ります。

```

234
235 20042d20
236 20042d2c
237
238
239 20042d8c
240
241
242 20042d94
243
244
245 20042dac
246
247 20042dc4
248 20042dd4
249 20042de4
250 20042df4
251 20042e04
252 20042e14
253 20042e24
254
255
256 20042e34
257 20042e50
258
259
260 20042e60
261
262
263
264 20042e64

```

```

static void static_test (void)
{
    char strdata[32] = "";

    /* Declare loop count variable */
    uint8_t ui_count = 0;

    /* Declare string variable to hold the string to be copied */
    char c_str[] = "STATIC \0";

    /* Declare variable buffer to store the copied string */
    const char c_replace[] = "TESTTEST\0";

    display_lcd(4, (uint8_t *) " Static Test");
    display_lcd(5, (uint8_t *) " Initialise c_str");
    display_lcd(6, (uint8_t *) " Replace contents ");
    display_lcd(7, (uint8_t *) " of c_str with ");
    display_lcd(8, (uint8_t *) " that of ucReplace");
    display_lcd(9, (uint8_t *) " ucStr = 'STATIC '");
    display_lcd(10, (uint8_t *) " ucReplace='TESTTEST'");

    /* Write ucStr variable, "STATIC" to LCD */
    sprintf(strdata, " c_str = '%s' ", c_str);
    display_lcd(12, (uint8_t *) strdata);

    /* Delay */
    delay();

    /* Begin for loop which writes one letter of ucReplace to the LCD at a time
    The nested while loops generate the delay between each letter change */
    for (ui_count = 0; ui_count < 8; ui_count++)

```

- デバッガは clear_display_area() 関数で実行を停止します。「F6」キーか「ステップ・オーバー」ボタン  を押すと、関数が実行され、LCDの表示がクリアされません。
- 次の処理では、R_OSTM_Init() 関数により割り込みハンドラによるLEDの点滅の間隔を設定し、R_OSTM_Open() 関数によりタイマを開始します。
- LEDの点滅間隔は、while ループ中でADCの値を読み出し、タイマ値として適切な値に変換して設定します。

```

181
182 20042b74
183
184 20042b78
185 20042b88
186 20042b98
187
188
189 20042ba8
190
191
192 20042bb8
193
194
195 20042bc0
196
197
198
199 20042bc4
200
201 20042bc8
202 20042bf0
203
204 20042c00
205 20042c8c
206
207
208 20042c9c
209
210 20042cb0
211
212
213 20042cc0
214
215 20042d1c
216
217

```

```

clear_display_area();

display_lcd(4, (uint8_t *) " Use RV1 to set delay");
display_lcd(5, (uint8_t *) " using timer OSTM ");
display_lcd(6, (uint8_t *) " when flashing LEDs");

/* Initialise OS timer (channel 0) */
R_OSTM_Init(DEVDVRY_CH_0, OSTM_MODE_INTERVAL, 500);

/* Start OS timer (channel 0) */
R_OSTM_Open(DEVDVRY_CH_0);

/* Initialise Analogue input (Potentiometer) on board */
R_ADC_Open();

while(1)
{
    R_ADC_Read();

    sprintf(strdata, " Flash Delay %d ms ", (uint16_t)g_adc_result);
    display_lcd(8, (uint8_t *) strdata);

    sprintf(strdata, " RV1 position %3d %% ", (int16_t)((g_adc_result/1023.0)*100.0));
    display_lcd(10, (uint8_t *) strdata);

    /* Set the minimum flash rate if the ADC value is less than the minimum */
    if (g_adc_result < MIN_FLASH_RATE)
    {
        g_adc_result = MIN_FLASH_RATE;
    }

    OSTM0.OSTMnCMP = (uint32_t)(P0_CLOCK_FREQUENCY_kHz * (g_adc_result + 1));
}

return (0);
}

```

- main.c 下方の Sample_OSTM0_Interrupt() 割込ハンドラ関数までスクロールしてください。
- この関数内の最初の行にブレークポイントを設定し、「再開」ボタンでプログラムを実行してください。
- LED 点滅タイマの設定時間が経過すると割込みが発生し、プログラムはこのブレークポイントで停止します。
- ブレークポイント欄をダブルクリックしてブレークポイントを解除した後、「再開」ボタンを押してプログラムを継続実行してください。

```
300  
301 20042f38  
302 20042f48  
303  
304 20042f50  
305  
306 20042f58  
307 20042f60
```

```
void Sample_OSTM0_Interrupt (uint32_t int_sense)  
{  
    R_INTC_Disable(INTC_ID_OSTM0TINT);  
    R_LED_Toggle(LED0);  
    R_INTC_Enable(INTC_ID_OSTM0TINT);  
}
```

ハードウェア構成の詳細については、RSK+RZA1H ユーザーズマニュアルと RZ/A1H グループ ユーザーズマニュアル ハードウェア編を参照してください。

5. 追加情報

技術サポート

e² studio の使用方法については、メニューバーから [ヘルプ]>[ヘルプ目次]を選択して、ヘルプファイルを参照してください。



RZ/A1H グループのマイクロプロセッサに関する詳細は、RZ/A1H グループ ユーザーズマニュアルハードウェア編を参照してください。

技術サポート窓口

「クイックスタートガイド」の「10. サポート」に記載されている連絡先をご参照ください

ルネサスのマイクロプロセッサに関する総合情報は、以下のウェブサイトより入手可能です：

<http://japan.renesas.com/> (日本サイト)

<http://www.renesas.com/> (グローバルサイト)

ルネサス RZ マイクロプロセッサ (ARM Cortex-A9) のリアルタイムオペレーティングシステムのデモンストレーションは FreeRTOS によって無償で提供されています。これらの情報は、下記の FreeRTOS ウェブサイトより入手可能です：

http://www.freertos.org/Renesas_RZ_Cortex-A9-RTOS.html

商標

本書で使用する商標名または製品名は、各々の企業、組織の商標または登録商標です。

著作権

本書の内容の一部または全てを予告無しに変更することがあります。

本書の著作権はルネサスエレクトロニクス株式会社にあります。ルネサスエレクトロニクス株式会社の書面での承諾無しに、本書の一部または全てを複製することを禁じます。

© 2014 Renesas Electronics Europe Limited. All rights reserved.

© 2014 Renesas Electronics Corporation. All rights reserved.

© 2014 Renesas Solutions Corp. All rights reserved.

改訂記録	RSK+RZA1H チュートリアルマニュアル
------	------------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.10.21	—	初版発行

RSK+RZA1H チュートリアルマニュアル e² studio 版

発行年月日 2014 年 10 月 21 日 Rev.1.00

発行 ルネサス エレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部 1753



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/contact/>

RZA1Hグループ