User's Manual

32

RX72T グループ

Renesas Starter Kit for RX72T

チュートリアルマニュアル

(CS+)

ルネサス 32 ビットマイクロコントローラ
RX ファミリ／RX700 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。

ルネサス エレクトロニクス
www.renesas.com

Rev.1.00 2018.11

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生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、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、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。
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。
6. 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。
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、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。
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。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. 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。
10. 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。
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。
12. 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.4.0-1　2017.11)

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うと、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

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CS+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にしています。

このマニュアルは、段階的にCS+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。

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。

本マニュアル中のスクリーンショットと実際に表示される画面が一部異なる場合があります。読み進めるにあたって問題はありません。

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。

RSKRX72T では次のドキュメントを用意しています。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
ユーザーズマニュアル	RSK ハードウェア仕様の説明	Renesas Starter Kit for RX72T ユーザーズマニュアル	R20UT4272JG
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とデバッギング方法の説明	Renesas Starter Kit for RX72T チュートリアルマニュアル	R20UT4273JG (本マニュアル)
クイックスタートガイド	A4 紙一枚の簡単なセットアップガイド	Renesas Starter Kit for RX72T クイックスタートガイド	R20UT4274JG
スマート・コンフィグレータ チュートリアルマニュアル	スマート・コンフィグレータの使用方法の説明	Renesas Starter Kit for RX72T スマート・コンフィグレータ チュートリアルマニュアル	R20UT4275JG
回路図	CPU ボードの回路図	Renesas Starter Kit for RX72T CPU ボード回路図	R20UT4271EG
ユーザーズマニュアル ハードウェア編	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	RX72T グループユーザーズマニュアル ハードウェア編	R01UH0803JJ

2. 略語および略称の説明

略語／略称	英語名	備考
ADC	Analog-to-Digital Converter	A/D コンバータ
API	Application Programming Interface	アプリケーションプログラムインタフェース
bps	bits per second	転送速度を表す単位、ビット/秒
CMT	Compare Match Timer	コンペアマッチタイマ
COM	COMmunications port referring to PC serial port	シリアル通信方式のインタフェース
CPU	Central Processing Unit	中央処理装置
E1/E2 Lite	Renesas On-chip Debugging Emulator	ルネサスオンチップデバッギングエミュレータ
GUI	Graphical User Interface	グラフィカルユーザインタフェース
IDE	Integrated Development Environment	統合開発環境
IRQ	Interrupt Request	割り込み要求
LCD	Liquid Crystal Display	液晶ディスプレイ
LED	Light Emitting Diode	発光ダイオード
LSB	Least Significant Bit	最下位ビット
LVD	Low Voltage Detect	電圧検出回路
MCU	Micro-controller Unit	マイクロコントローラユニット
MSB	Most Significant Bit	最上位ビット
PC	Personal Computer	パーソナルコンピュータ
PLL	Phase-locked Loop	位相同期回路
Pmod™	-	Pmod™は Digilent Inc.の商標です。Pmod™インタフェース明細は Digilent Inc.の所有物です。Pmod™明細については Digilent Inc.の Pmod™ License Agreement ページを参照してください。
RAM	Random Access Memory	ランダムアクセスメモリ
ROM	Read Only Memory	リードオンリーメモリ
RSK	Renesas Starter Kit	ルネサススタータキット
RTC	Real Time Clock	リアルタイムクロック
SCI	Serial Communications Interface	シリアルコミュニケーションインタフェース
SPI	Serial Peripheral Interface	シリアルペリフェラルインタフェース
TFT	Thin Film Transistor	薄膜トランジスタ
UART	Universal Asynchronous Receiver/Transmitter	調歩同期式シリアルインタフェース
USB	Universal Serial Bus	シリアルバス規格の一種
WDT	Watchdog Timer	ウォッチドッグタイマ

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 7
1.1 目的 7
1.2 特徴 7

2. はじめに 8
2.1 スマート・コンフィグレータについて 8

3. チュートリアルプロジェクトワークスペース 9
3.1 はじめに 9
3.2 CS+の開始 9
3.3 デバッグ・ツールの設定 13
3.4 ビルド設定 14

4. チュートリアルプログラムのビルド 15
4.1 コードのビルド 15
4.2 エミュレータの接続 16
4.3 プロジェクトの保存 16

5. チュートリアルのダウンロードと実行 17
5.1 プログラムコードのダウンロード 17
5.2 コードの実行 17

6. チュートリアルレビュー 18
6.1 プログラム初期化 18
6.2 メイン関数 19

7. 追加情報 22

1. 概要

1.1 目的

本 RSK はルネサスマイクロコントローラ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

本 RSK は以下の特徴を含みます：

- ルネサスマイクロコントローラのプログラミング
- ユーザコードのデバッギング
- 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
- 周辺機能初期化コードのサンプル

CPU ボードはマイクロコントローラ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？

プロジェクトジェネレータは、選択可能な 3 種類のビルドコンフィグレーションを持つチュートリアルプロジェクトを作成します。

- ‘DefaultBuild’はデバッガのサポートおよび最適化レベル 2 を含むプロジェクトを構築します。
- ‘Debug’はデバッガのサポートを含むプロジェクトを構築します。最適化レベルは 0 に設定されています。
- ‘Release’は最適化された製品リリース用に適したコードを構築します。最適化レベルは 2 に、デバッグ情報を出力しないように設定されています。

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プロジェクトジェネレータを使用してインストールされます。本チュートリ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。

本マニュアル中のソースコード画面のライン番号が実際のソースコードと異なる場合がありますが、本マニュアルで記載されている内容と機能的違いはございません。

> チュートリアルは RSK の使用方法の説明を目的とするものであり、CS+、コンパイラまたは E2 エミュレータ Lite 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 スマート・コンフィグレータについて

本製品で提供しているサンプルコードの一部はスマート・コンフィグレータを使用してコードを生成しています。スマート・コンフィグレータは C ソースコードの生成、マイクロコントローラのプロジェクト設定を行うための連携ツールです。スマート・コンフィグレータは直感的な GUI を使用することで、様々なマイクロコントローラの周辺機能や動作に必要なパラメータを設定することができ、開発工数の大幅な削減が可能です。

スマート・コンフィグレータによって生成されるコードは、特定の周辺機能ごとに 3 つのコードを生成します（「Config_xxx.h」、「Config_xxx.c」、「Config_xxx_user.c」）。例えば A/D コンバータの場合、周辺機能を表す xxx は’ S12AD’と名付けられます。これらのコードはユーザの要求を満たすために、カスタムコードを自由に加えることができます。カスタムコードを加える場合、以下に示すコメント文の間にカスタムコードを加えてください。

```
/* Start user code for adding. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
```

スマート・コンフィグレータの GUI 上で設定した内容を変更したい場合等、再度コード生成を行う場合にスマート・コンフィグレータはこれらのコメント文を見つけて、コメント文の間に加えられたカスタムコードを保護します。

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

CS+はルネサス統合開発ツールで、ユーザはこれを使用してルネサスマイクロコントローラのソフトウェアプロジェクトをコンパイル、プログラミング、デバッグすることが可能です。CS+は Renesas Starter Kit 製品インストール時にインストールされます。本マニュアルでは、Tutorial 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。

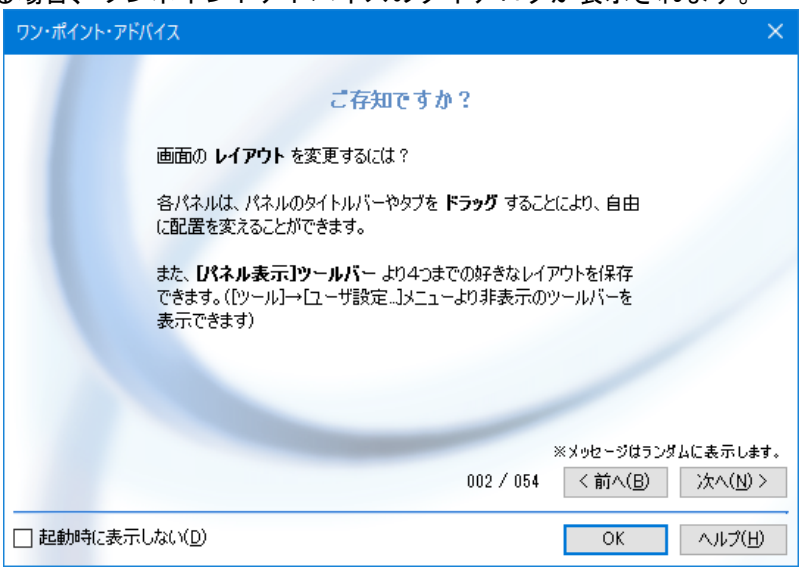
3.2　CS+の開始

Windows™ 7: スタートメニュー > すべてのプログラム > Renesas Electronics CS+ > CS+ for CC (RL78,RX, RH850)

Windows™ 8.1/8: ⓓをクリックして[アプリ]ビューを表示 > ‘CS+ for CC (RL78,RX, RH850)’アイコン

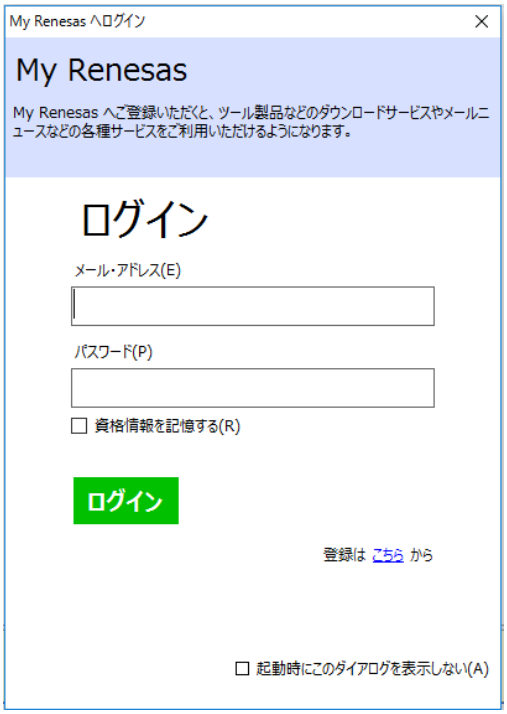
Windows™ 10: スタートメニュー > すべてのアプリ > Renesas Electronics CS+ > CS+ for CC (RL78,RX, RH850)

CS+を初めて使用する場合、ワンポイントアドバイスのダイアログが表示されます。



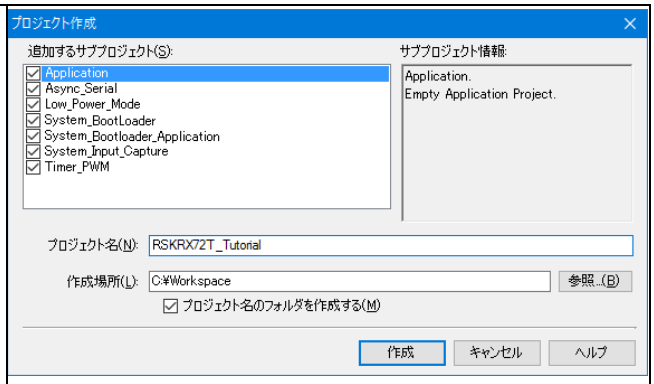
<OK>をクリックし、ダイアログを閉じてください。

なお、初回起動時は「My Renesas」へのログイン画面が表示されます。メールアドレス、パスワードを入力してログインボタンをクリックしてください。



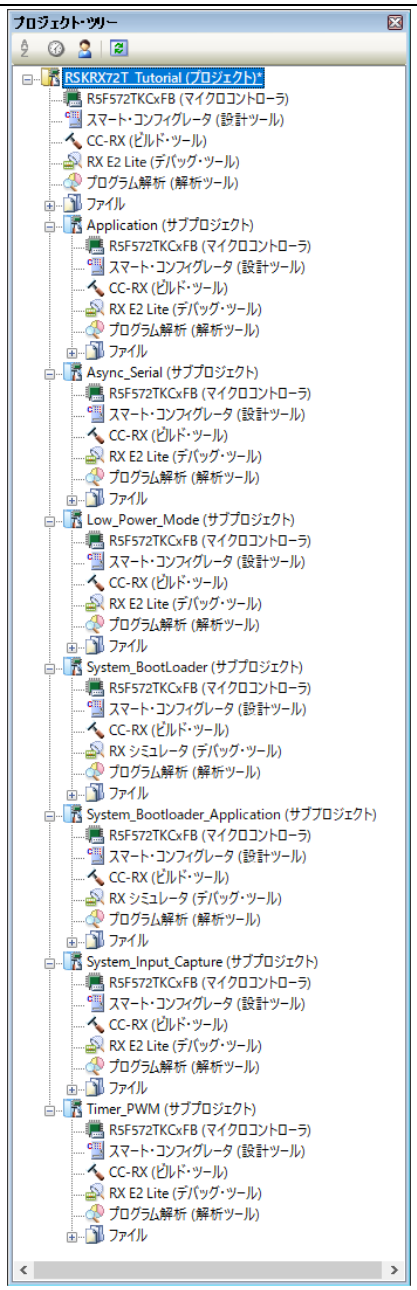
その後、スタートパネルが現れます。‘RSK のサンプル・プロジェクトを開く’から RSKRX72T_Tutorial を選択し、<GO>をクリックしてください。この操作によって、RSKRX72T_Tutorial プロジェクトのコピーを保存します。

- CS+は‘プロジェクト作成’ダイアログを表示します。
- 各サブプロジェクト名のチェックボックスをチェックし、サブプロジェクトをすべて追加してください。各サブプロジェクトの情報はダイアログ上のサブプロジェクト情報の下に表示されます。
- プロジェクト名を入力し、作成場所を指定して<作成>をクリックしてください。
- コピーされたファイルを参照するには、プロジェクト・ツリーにリスト化されたファイルをダブルクリックしてください。新しいウィンドウが開きます。

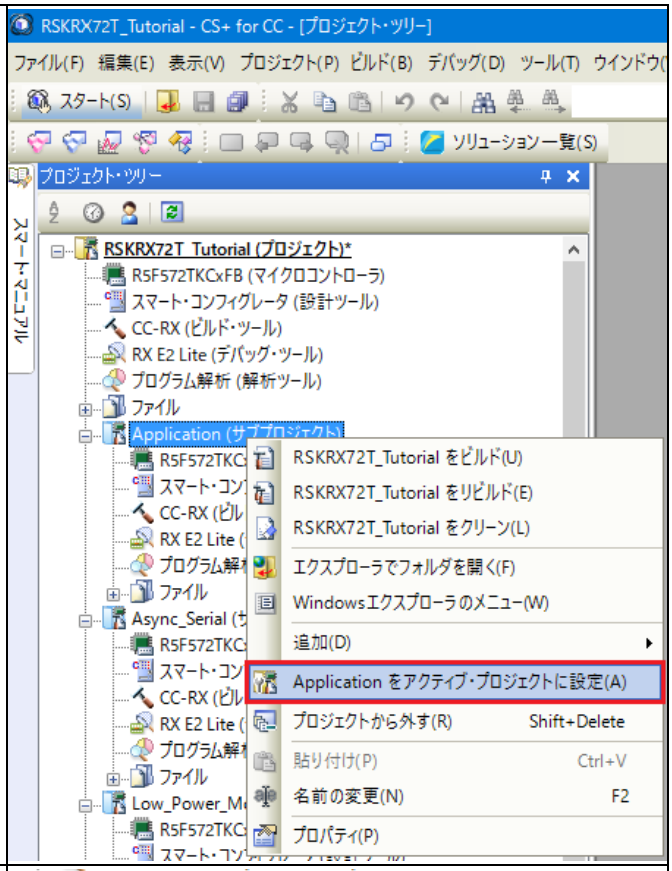


- CS+はスクリーンショットに見られるようなプロジェクト・ツリーを持つプロジェクトを作成し、開きます。
- スクリーンショットのファイルフォルダはマスタープロジェクト RSKRX72T_Tutorial に属します。
- このフォルダは個別のフォルダ構造で用意されたテキストファイルを含むプロジェクトソースおよびヘッダファイルをすべて含んでおり、リストします。
- ファイルフォルダの下にサブプロジェクトがリストされます。
- 現在アクティブなプロジェクトはプロジェクト名に下線が含まれます。初期設定によってRSKRX72T_Tutorial がアクティブ・プロジェクトに設定されています。

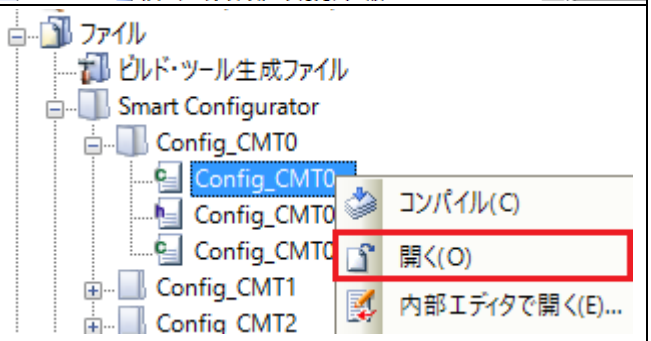
注：CS+とスマート・コンフィグレータを連携した後のスクリーンショットです。そのため、‘スマート・コンフィグレータ(設計ツール)’がプロジェクト・ツリー上に表示されています。



- アクティブ・プロジェクトを変更するには、アクティブに変更したいプロジェクト/サブプロジェクトを右クリックし、‘プロジェクト名/サブプロジェクト名をアクティブ・プロジェクトに設定’を選択します。
- スクリーンショットは Application サブプロジェクトをアクティブ・プロジェクトに変更する例です。



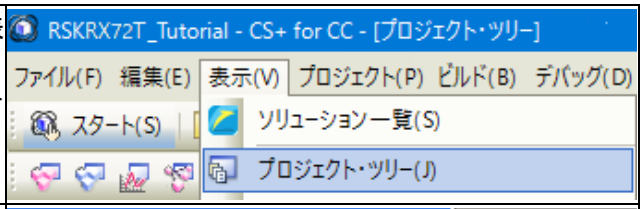
- ファイルフォルダは 4 つのサブフォルダを含んでいます。
- いくつかのソース・ファイルはスマート・コンフィグレータによって生成され、‘Smart Configurator’フォルダにグループ化されます。これらのファイルはスマート・コンフィグレータによって生成されたことを示すため、ファイル名の前に‘Config’が付けられます。他のユーザによって作成/インクルードされたファイルは Smart Configurator とは別のフォルダにリストされます。
- ファイルを参照するには、参照したいファイルを右クリックし、‘開く(O)’を選択します。ファイルをダブルクリックしても参照できます。



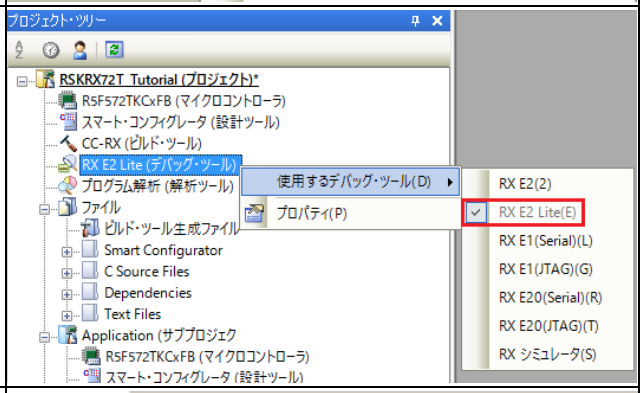
3.3　デバッグ・ツールの設定

注：Tutorial プロジェクトは既にデバッグ・ツールの設定がされています。このセクションは新しいプロジェクトを作成するための説明です。

- プロジェクト・ツリーは CS+の左側ウィンドウに表示されます。
- プロジェクト・ツリーはメニューバーから起動できます。（表示 -> プロジェクト・ツリー）



- RX xxx(デバッグ・ツール)を右クリックし、RX E2 Lite を選択してください。スクリーンショットでは RX E2 Lite が選択されています。

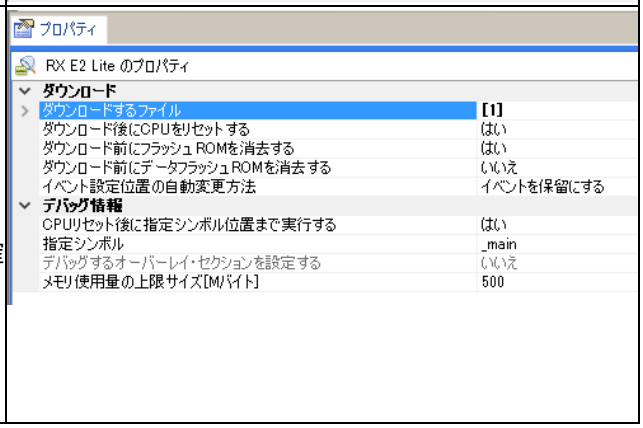


- RX E2 Lite(デバッグ・ツール)を右クリックし、プロパティを選択してください。
- 接続用設定タブをクリックしてください。
  設定内容がスクリーンショットと同じであることを確認してください。

プロジェクトはコードをダウンロードした後にメイン関数の先頭でコード実行を停止させる設定になっています。エントリポイントを別の関数に指定する場合：

- ‘ダウンロード・ファイル設定’タブをクリックしてください。
- 指定シンボルを別の関数に変更してください。
- 関数名の前にアンダースコア（“_”）があることを確認してください。

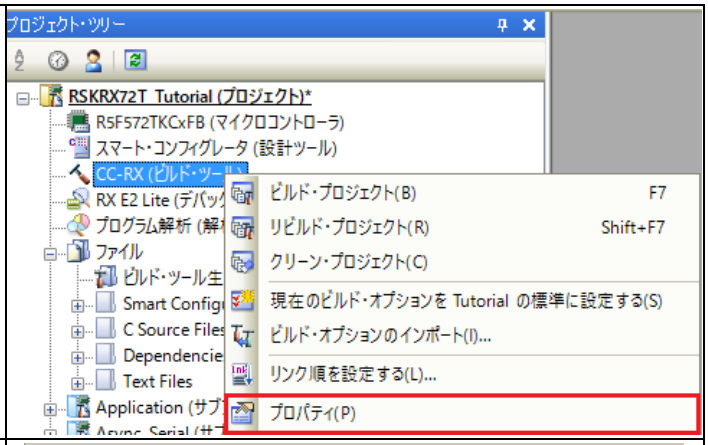
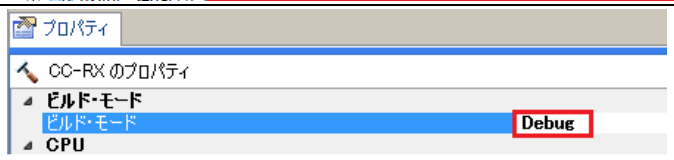
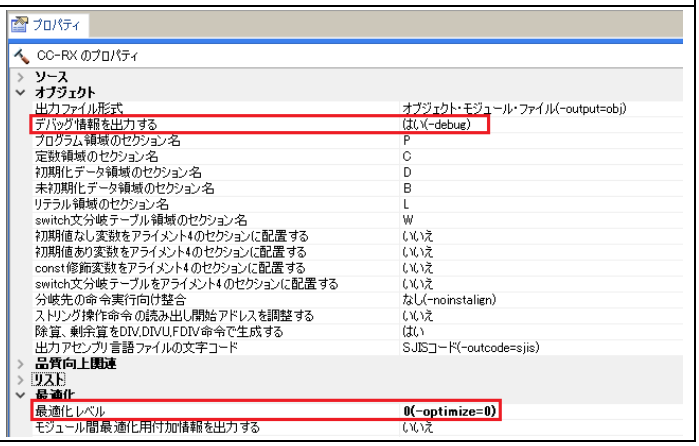
注：割り込みハンドラをエントリポイントとして指定しないでください。



3.4　ビルド設定

ビルド設定は CC-RX (ビルド・ツール)のプロパティから選択できます。利用可能なオプションは‘DefaultBuild’、‘Debug’、‘Release’です。‘DefaultBuild’および‘Debug’はデバッガを備えた設定になっています。‘Release’は最終の ROM 化用プログラムのために設定されます。

3 つのビルド間の共通の違いは、最適化セットおよびデバッガ設定です。最適化が有効の場合、デバッガがコードを予想外の順序で実行するようなケースがあります。デバッグをスムーズに処理するためには、デバッグされるコードの最適化を無効にすることを推奨します。

• CC-RX (ビルド・ツール)を右クリックし、プロパティを選択してください。	
• 共通オプションタブを選択してください。 • ビルド・モードを‘Debug’に設定してください。	
• ‘コンパイル・オプション’タブを選択してください。	
• デバッグ情報を出力するが‘はい(-debug)’に設定されていることを確認してください。 • 最適化レベルが‘0(-optimize=0)’に設定されていることを確認してください。	

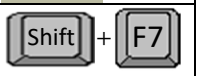
4. チュートリアルプログラムのビルド

Tutorial プロジェクトのビルド設定は、ツールチェインオプションで既に設定されています。ツールチェインオプションを表示するためには、プロジェクト・ツリーの CC-RX (ビルド・ツール)をダブルクリックし、利用可能なタブを選択してください。

• 各タブで利用可能なオプションを確認してください。ここでは、デフォルトのオプション設定にしてください。 • 選択終了後に<X>をクリックしてプロパティ画面を閉じます。	

4.1 コードのビルド

プロジェクトのビルド用に 3 つのショートカットがあります。

• ツールバーの‘プロジェクトをビルドします。’ボタンです。プロジェクト・ツリー中の全プロジェクトをビルドします。	
• キーボードの‘F7’ボタンです。上記のボタン選択の場合と同じです。	
• ツールバーの‘プロジェクトをリビルドします’ボタンです。プロジェクトファイルをすべてリビルドします。	
• キーボードの’Shift’ + ‘F7’ボタン同時押しです。上記のボタン選択の場合と同じです。	
• ツールバーの‘ビルド後デバッグ・ツールへプログラムをダウンロードします。(F6)’ボタンです。プロジェクトのビルドを行い，ビルド後にアクティブ・プロジェクトで現在選択しているデバッグ・ツールにダウンロードを実行します。	
• キーボードの‘F6’ボタンはツールバーの‘ビルド後デバッグ・ツールへプログラムをダウンロードします。’ボタンと同じです。	

ここで、キーボードの‘F7’ボタンを押すか、または上記アイコンの 1 つを選択し、プロジェクトをビルドしてください。ビルド中の各段階で、アウトプットウィンドウにビルド状況が表示されます。ビルド終了時、ビルド中に発生したエラーおよび警告の表示がされます。

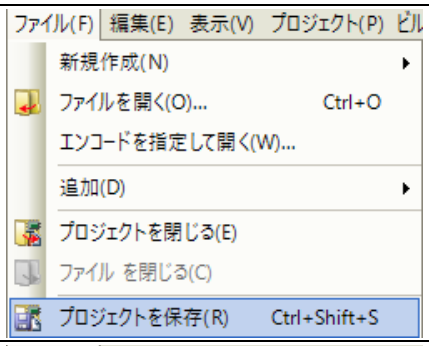
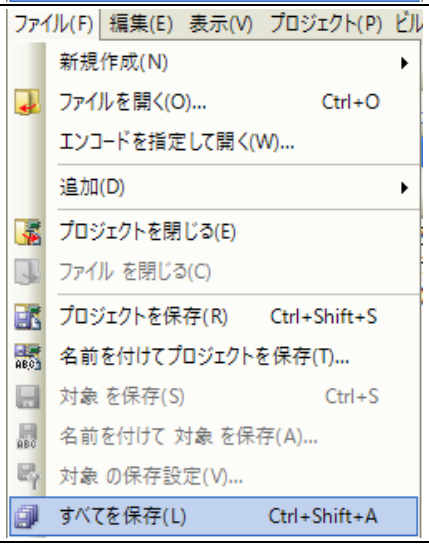
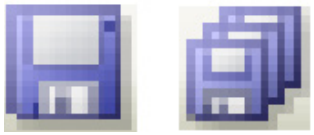
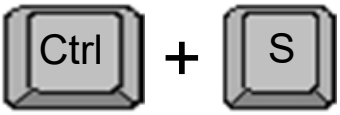
4.2　エミュレータの接続

本チュートリアルでは、外部から CPU ボードに電源する必要はありません。電源は USB ポートから供給されます。

E2 Lite のホストコンピュータへの接続方法は、クイックスタートガイドに詳しく記載されています。以下は、クイックスタートガイドの手順が踏まれ、E2 Lite 用のドライバが既にインストールされていることを前提としています。

- Pmod LCD を CPU ボードの PMOD1 コネクタに取り付け、コネクタの全てのピンが正しくソケットに収まっていることを確認してください。
- E2 Lite をご使用のコンピュータの USB ポートに接続してください。
- E2 Lite を CPU ボードに接続します。‘E1/E2 Lite’のシルク印字のある E1/E2 Lite コネクタに接続してください。

4.3　プロジェクトの保存

プロジェクトの設定を変更した場合、プロジェクトを保存することを推奨します。 • ‘ファイル’	‘プロジェクトを保存’ を選択します。	
CS+中のファイルを変更した場合、次の操作で保存できます。 • ‘ファイル’	‘すべてを保存’ を選択します。	
ツールバーの‘保存’または‘すべてを保存’ボタンでファイルを保存できます。	
また、キーボードでもファイルを保存できます。	

5.　チュートリアルのダウンロードと実行

5.1　　プログラムコードのダウンロード

CS+でコードのビルドが完了したら、プログラムを CPU ボード上のマイクロコントローラにダウンロードする必要があります。

• ツールバーの’ダウンロード’ボタンをクリックしてください。	
• ダウンロード完了後、デバッガおよびコードは実行準備ができています。プログラムカウンタ表示はメイン関数内の最初のインストラクションを示します。これは‘main.c’のプログラムエントリポイントです。	/*** * Function Name: main * Description : This function implements main function. * Arguments : None * Return Value : None *** void main(void) { /* Initialize the switch module */ R_SWITCH_Init(); /* Set the call back function when SW1 or SW2 is pressed */ R_SWITCH_SetPressCallback(cb_switch_press); /* Initialize the debug LCD */ R_LCD_Init(); /* Displays the application name on the debug LCD */ R_LCD_Display(0, (uint8_t *)" RSKRX72T"); R_LCD_Display(1, (uint8_t *)" Tutorial "); R_LCD_Display(2, (uint8_t *)" Press Any Switch ");

コードを実行する前に、コンピュータの USB ポートと CPU ボード上の USB シリアルポート（シルク印字‘G1CUSB0’）を USB ケーブルで接続する必要があります。はじめて接続した場合、コンピュータの画面にドライバのインストールメッセージが表示され、自動的にデバイスドライバはインストールされます。デバイスマネージャ上のポート(COM と LPT)に’RSK USB Serial Port (COMx)’が現れますので、COM ポート番号を確認し、ターミナルソフトを起動して確認した COM ポート番号の設定を行ってください(ボーレート：19200、データ長：8、パリティ：なし、ストップビット：1、フロー制御：なし)。

5.2　　コードの実行

プログラムが CPU ボード上のマイクロコントローラにダウンロードされると、プログラムを実行できます。現在のプログラムカウンタ位置からプログラムを始めるため‘実行’ボタンまたは、‘F5’を押してください。	

6. チュートリアルレビュー

本章では、CS+中の Tutorial コードの各セクションおよび基礎的なデバッグ機能を確かめます。

6.1 プログラム初期化

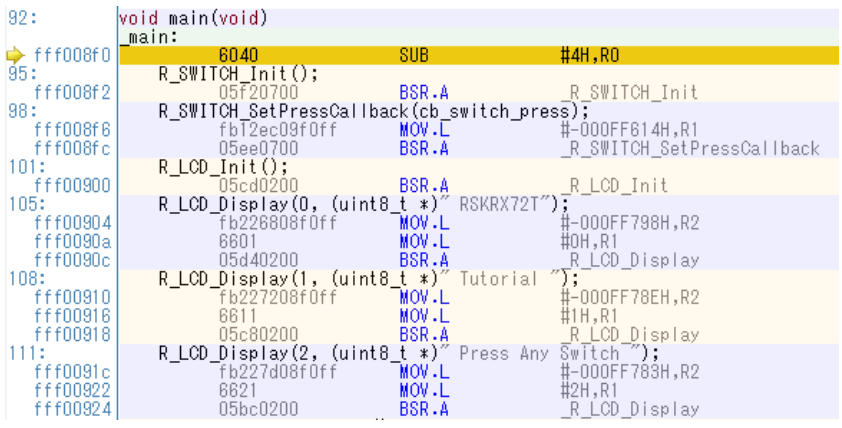
メインプログラムが実行される前にマイクロコントローラは初期化されます。Tutorial プロジェクトおよび残りのサンプルプロジェクトはデバッグ・ツールの設定により、ユーザはハードウェア初期化コードの実行工程を見ることができません。プログラムダウンロード後のエントリポイントを変更する場合はセクション 3.3 を参照してください。ハードウェアの初期化を見る場合、関数名は‘_R_Systeminit’を指定してください。

チュートリアルコードの以下の部分は、主要機能が正確に実行できるように、CPU ボード上のマイクロコントローラを初期化するために使用されます。マイクロコントローラはリセットスイッチまたはパワーオンリセットによってリセットされるごとに、初期化コードが実行されます。

Tutorial コードがマイクロコントローラにダウンロードされていることを確認し、デバッグツールバーの‘CPU リセット’をクリックしてください。	

- コード表示をメニューバーの‘逆アセンブル’ボタン、‘混合’ボタンで表示を切り替えることができます。

逆アセンブル表示または混合表示から C ソース表示に戻る場合、プログラムカウンタが指しているコード行を右クリックして、‘ソースへジャンプ(S)’をクリックしてください。



6.2 メイン関数

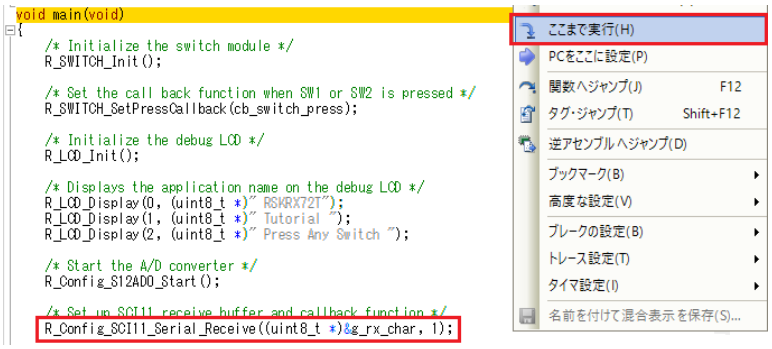
このセクションでは、メイン関数がコールされたプログラムコードがどのように動作するかを見ます。

- ‘R_Config_SCI11_Serial_Receive’ 関数を右クリックして、‘ここまで実行’を選択してください。
- ‘R_LCD_Init’関数は LCD を初期化します。‘R_LCD_Display’関数はストリングデータを LCD に以下のように表示します。

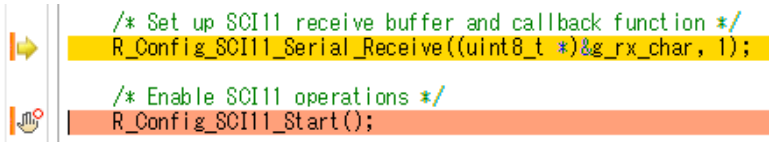
  1 行目： RSKRX72T

  2 行目： Tutorial

  3 行目： Press Any Switch

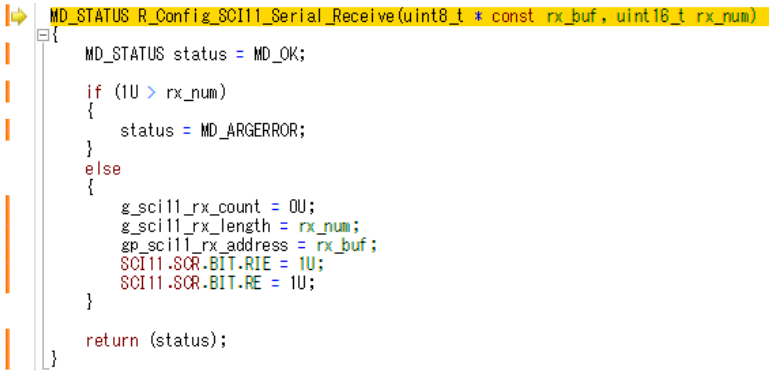


- ‘R_Config_SCI11_Start’関数にソフトウェア・ブレークを設定してください（行数の左側にあるブレークポイント行）。



- ‘ステップ・イン’ボタンをクリックまたは‘F11’ボタンを押して‘R_Config_SCI11_Serial_Receive’関数にエントリします。

- プログラムカウンタは‘R_Config_SCI11_Serial_Receive’関数に移動します。この関数は、スマート・コンフィグレータによって生成され、受信バイトカウンタ、受信バイト数、バッファアドレスを設定します。そして、SCI11 割り込み処理内で設定した受信バイト数に達したとき、コールバック関数に呼び出されます。
- スマート・コンフィグレータを使用したプロジェクトにつきましては、スマート・コンフィグレータチュートリアルマニュアルを参照ください。
- ‘実行’ボタンをクリックしてプログラム実行を再開してください。



- プログラムカウンタは‘R_Config_SCI11_Start’関数で停止します。
- ‘ステップ・オーバー’ボタンをクリックまたは‘F10’ボタンを押してください。

‘R_Config_SCI11_Start’関数は、SCI の開始、割り込みを許可します。
プログラムは、CPU ボードのスイッチ入力、または SCI 割り込みが発生するまで while ループを実行します。スイッチ入力か SCI 割り込み発生で、A/D 変換を実行します。

```
    /* Set up SCI11 receive buffer and callback function */
    R_Config_SCI11_Serial_Receive((uint8_t *)&g_rx_char, 1);

    /* Enable SCI11 operations */
    R_Config_SCI11_Start();

    while (1U)
    {
```

- while ループ内の‘lcd_display_adc’関数を呼び出します。
- ‘lcd_display_adc’処理内の最初の処理上で右クリックし、ハードウェア・ブレークを設定してください。
- プロジェクト・ツリーにある、‘Config_SCI11_user.c’ファイルを開いてください。
  ‘r_Config_SCI11_callback_receiveend’関数までスクロールしてください。

```
    while (1U)
    {
        uint16_t adc_result;

        /* Wait for user requested A/D conversion flag to be set (SW1 or SW2) */
        if (TRUE == g_adc_trigger)
        {
            /* Call the function to perform an A/D conversion */
            adc_result = get_adc();

            /* Display the result on the LCD */
            lcd_display_adc(adc_result);
```

- ‘r_Config_SCI11_callback_receiveend’関数で、画像の箇所にハードウェア・ブレークを設定してください。
- ‘実行’ボタンをクリックまたは‘F5’ボタンを押してプログラムを実行してください。

```
static void r_Config_SCI11_callback_receiveend(void)
{
    /* Start user code for r_Config_SCI11_callback_receiveend. Do

    /* Check the contents of g_rx_char */
    if (('c' == g_rx_char) || ('C' == g_rx_char))
    {
        g_adc_trigger = TRUE;
    }

    /* Set up SCI11 receive buffer and callback function again */
    R_Config_SCI11_Serial_Receive((uint8_t *)&g_rx_char, 1);

    /* End user code. Do not edit comment generated here */
}
```

- ターミナルソフト画面で、キーボードの‘c’キーを押してください。
- プログラムは‘r_Config_SCI11_callback_receiveend’関数のハードウェア・ブレークポイントで停止します。
- ハードウェア・ブレークのアイコンをクリックしてブレークポイントを削除してください。
- ‘実行’ボタンをクリックまたは‘F5’ボタンを押してプログラムを実行してください。

```
static void r_Config_SCI11_callback_receiveend(void)
{
    /* Start user code for r_Config_SCI11_callback_receiveend. Do
    /* Check the contents of g_rx_char */
    if (('c' == g_rx_char) || ('C' == g_rx_char))
    {
        g_adc_trigger = TRUE;
    }
    /* Set up SCI11 receive buffer and callback function again */
    R_Config_SCI11_Serial_Receive((uint8_t *)&g_rx_char, 1);
    /* End user code. Do not edit comment generated here */
}
```

- プログラムは‘main’関数の while ループ内のハードウェア・ブレークポイントで停止します。
- ハードウェア・ブレークのアイコンをクリックしてブレークポイントを削除してください。
- ‘実行’ボタンをクリックまたは‘F5’ボタンを押してプログラムを実行してください。

```
while (1U)
{
    uint16_t adc_result;
    /* Wait for user requested A/D conversion flag to be set (SW1 or SW2) */
    if (TRUE == g_adc_trigger)
    {
        /* Call the function to perform an A/D conversion */
        adc_result = get_adc();
        /* Display the result on the LCD */
        lcd_display_adc(adc_result);
```

プログラムは、CPU ボード上のユーザスイッチを押すことで A/D 変換を実行します。そして、ポテンショメータでコントロールされた電圧値の A/D 変換結果を LCD およびターミナル画面に表示します。さらに、CPU ボード上のユーザ LED で A/D 変換回数をバイナリ形式で点灯表示します。

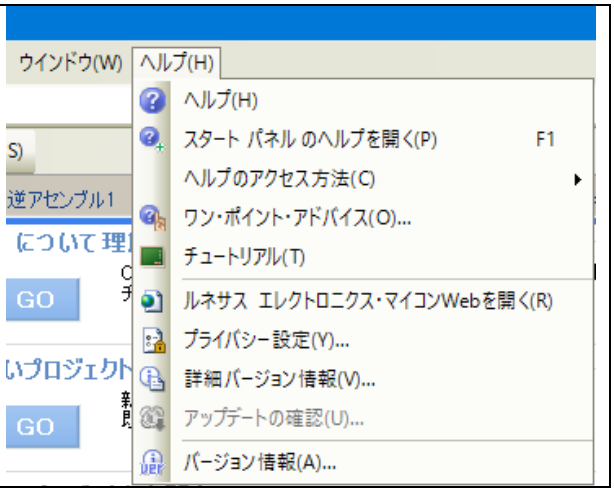
- ‘停止’ボタンをクリックし、プログラム実行を停止してください。

ハードウェアに関する詳細は、RSKRX72T ユーザーズマニュアルおよび RX72T 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。

E2 エミュレータ Lite は本マニュアルでは説明していない高度な機能を持っています。E2 エミュレータ Lite の詳細情報は、 E2 エミュレータ Lite のユーザーズマニュアルを参照してください。

7. 追加情報

サポート

CS+の使用方法等の詳細情報は、CS+のヘルプメニューを参照してください。	

RSKRX72T で提供されるサンプルコードの一部はスマート・コンフィグレータプラグインを使用しています。スマート・コンフィグレータによって生成されたソース・ファイルには「“R_”、“r_”や“Config_”」がプリフィックスされています。

RX72T グループマイクロコントローラに関する詳細情報は、RX72T 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。
アセンブリ言語に関する詳細情報は、RX ファミリユーザーズマニュアルソフトウェア編を参照してください。

オンラインの技術サポート、情報等は https://www.renesas.com/rskrx72t より入手できます。

オンライン技術サポート

技術関連の問合せは、https://www.renesas.com/support/contact.html を通じてお願いいたします。

ルネサスのマイクロコントローラに関する総合情報は、https://www.renesas.com/をご利用ください。

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。

© 2018 Renesas Electronics Europe Limited. All rights reserved.
© 2018 Renesas Electronics Corporation. All rights reserved.

改訂記録	RX72T グループ Renesas Starter Kit for RX72T　チュートリアルマニュアル (CS+)

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2018.11.30	ー	初版発行

RX72T グループ
Renesas Starter Kit for RX72T　チュートリアルマニュアル (CS+)

発行年月日　2018 年 11 月 30 日　　Rev.1.00

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

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 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■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：https://www.renesas.com/contact/

© 2018 Renesas Electronics Corporation. All rights reserved.

Colophon 3.1

RX72T グループ

R20UT4273JG0100

ルネサス エレクトロニクス株式会社