

## RX63T グループ

## Renesas Starter Kit チュートリアルマニュアル

ルネサス 32 ビットマイクロコンピュータ  
RX ファミリ / RX600 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も、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、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

### 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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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くと、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 High-performance Embedded Workshop(HEW)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 HEW 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。

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。

RSKRX63T 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	RSKRX63T ユーザーズマニュアル	R20UT0957JG
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グ方法の説明	RSKRX63T チュートリアルマニュアル	R20UT0958JG (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	RSKRX63T クイックスタートガイド	R20UT0959JG
回路図	CPU ボードの回路図	RSKRX63T CPU ボード回路図	R20UT0956EG
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	RX63T グループ ユーザーズマニュアル ハードウェア編	R01UH0238JJ

## 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
CMT	Compare Match Timer	コンペアマッチタイマ
CPU	Central Processing Unit	中央処理装置
DVD	Digital Versatile Disc	デジタルヴァーサタイルディスク
E1	Renesas On-chip Debugging Emulator	ルネサスオンチップデバッグエミュレータ
GUI	Graphical User Interface	グラフィカルユーザインタフェース
GPT	General PWM Timer	汎用 PWM タイマ
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
MTU	Multi-Function Timer Pulse Unit	マルチファンクションタイマパルスユニット
PC	Personal Computer	パーソナルコンピュータ
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	Realtime Clock	リアルタイムクロック
SAU	Serial Array Unit	シリアルアレイユニット
SCI	Serial Communications Interface	シリアルコミュニケーションインタフェース
SFR	Special Function Registers	周辺機能を制御するためのレジスタ
TAU	Timer Array Unit	タイマアレイユニット
TPU	Timer Pulse Unit	タイマパルスユニット
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

# 目次

1. 概要 .....	7
1.1 目的 .....	7
1.2 特徴 .....	7
2. はじめに .....	8
2.1 ソースコードについて .....	8
3. チュートリアルプロジェクトワークスペース .....	9
4. プロジェクトワークスペース .....	10
4.1 はじめに .....	10
4.2 HEW の開始 .....	10
4.3 ビルドコンフィグレーションとデバッグセッション .....	11
4.3.1 ビルドコンフィグレーション .....	11
4.3.2 デバッグセッション .....	11
5. チュートリアルプログラムのビルド .....	12
5.1 コードのビルド .....	12
5.2 エミュレータの接続 .....	13
5.3 E1によるターゲットの接続 .....	13
6. チュートリアルのダウンロードと実行 .....	16
6.1 プログラムコードのダウンロード .....	16
6.2 コードの実行 .....	16
7. チュートリアルレビュー .....	17
7.1 プログラム初期化 .....	17
7.2 メイン関数 .....	20
8. 追加情報 .....	24

## 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グ
- スイッチ、LED、ポテンシオ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 2 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。

- 'Debug' はデバッグのサポートを含むプロジェクトを構築します。
- 'Release' は製品リリース用に適したコードを構築します。

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。

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、High-performance Embedded Workshop、コンパイラツールチェーンまたは E1 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

ワークスペースには 2 種類のビルドコンフィグレーション用の全ファイルを含みます。チュートリアルコードは、デバッグおよびリリースのビルドコンフィグレーションの両方で共通です。

High-performance Embedded Workshop のビルドコンフィグレーションのメニューを使用し、各々のビルドコンフィグレーションから特定のファイルを除外して、プロジェクトを作成することができます。これにより、デバックビルドにはモニタを含み、リリースビルドには含まないといった設定が可能になります。共通の C ファイルの内容は、ビルドコンフィグレーションオプションの defines セットアップおよび同ファイル内の#ifdef ステートメントで管理されます。

プロジェクトファイルは 1 つのセットのみを取扱うことで、管理の簡素化が図れます。

## 4. プロジェクトワークスペース

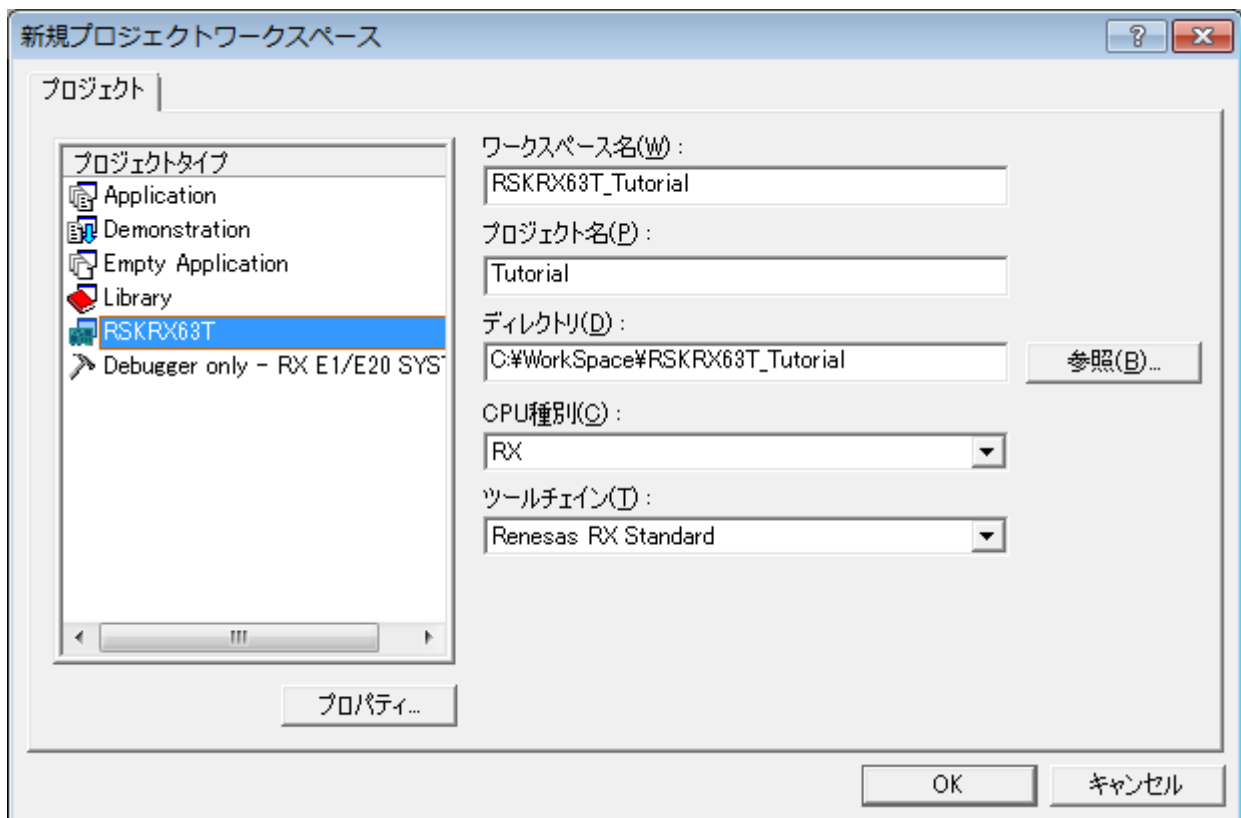
### 4.1 はじめに

High-performance Embedded Workshop 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、プログラム、デバッグすることができます。High-performance Embedded Workshop は Renesas Starter Kit 製品インストール時にインストールされます。本チュートリアルでは、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。

### 4.2 HEW の開始

まず、Windows のスタートメニューから High-performance Embedded Workshop を起動して、チュートリアルプログラムを見てみましょう。

[ファイル -> 新規ワークスペース...]メニューから新規ワークスペースを開くか、または'ようこそ!'ダイアログで'新規プロジェクトワークスペースの作成'を選択してください。



上図は RSKRX63T 選択時の新規プロジェクトワークスペースの一例です。

- 'RX'CPU 種別および'Renesas RX Standard'ツールチェーンを選択します。
- プロジェクトリストから'RSKRX63T'を選択します。
- ワークスペース名を入力します。全てのファイルはこの名称のディレクトリ下に置かれます。
- プロジェクト名欄は、上記ワークスペースと同じ名前でも自動的に入力されますが、これは変更可能です。
- High-performance Embedded Workshop では複数のプロジェクトを1つのワークスペースに追加できます。後に、サンプルコードのプロジェクトを保存する可能性がありますので、ここではチュートリアルプロジェクトに適した名称をつけることを推奨します。
- <OK>をクリックし、Renesas Starter Kit プロジェクトジェネレータを起動します。

次のダイアログに、利用可能なプロジェクトが表示されます。後に説明する Tutorial コードを選択してください。その他のオプションとして、各種周辺機能の使用例を示す Sample コードがあります。これを選択すると、新たなダイアログが開き、デバイス周辺機能用のサンプルコードがいくつか表示されます。最後のオプションは、アプリケーションビルド用で、デバッガは設定されていますが、プログラムコードはありません。これは、ユーザがデバッガを設定せずにコードを新規作成したい場合に適しています。

- プロジェクトとして'Tutorial'を選択し、<Next>をクリックします。
- <Finish>をクリックし、プロジェクトを作成します。

プロジェクトジェネレータのウィザードが確認ダイアログを表示します。<OK>をクリックすると、プロジェクトを作成し、必要なファイルをコピーします。

このプロジェクトの全ファイルを示すツリーが High-performance Embedded Workshop に表示されます。

- ワークスペース画面で'main.c'ファイルをダブルクリックします。画面にコードが表示されます。

### 4.3 ビルドコンフィグレーションとデバッグセッション

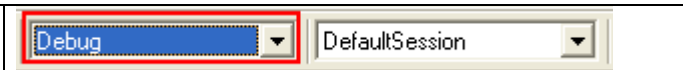
作成されたワークスペースには、2 つのビルドコンフィグレーションと 2 つのデバッグセッションが含まれています。ビルドコンフィグレーションでは、同じプロジェクトを異なるコンパイラオプションでビルドすることが可能です。ユーザが利用できるオプションは、High-performance Embedded Workshop のマニュアルに詳しく記載されています。

#### 4.3.1 ビルドコンフィグレーション

ツールバーの左側のドロップダウンリストからビルドコンフィグレーションを選択します。利用可能なオプションは、Debug と Release です。Debug ビルドは、デバッガとの使用に設定されています。Release ビルドは、最終 ROM コード用の設定です。

これら 2 種のビルドの違いとして、最適化設定が挙げられます。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、デバッグをスムーズに処理する為には、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。

- 'Debug'コンフィグレーションを選択します。



#### 4.3.2 デバッグセッション

デバッグセッションはツールバーの右側のドロップダウンリストから選択します。Renesas Starter Kit の種類によってオプションは異なりますが、どのオプションも必ずデバッグを可能にする同様のデバッグインタフェースを含みます。その他の選択として'DefaultSession'があります。デバッグセッションの目的は、同一プロジェクトで異なったデバッガ・ツールの使用や、異なったデバッガ設定を可能にすることにあります。

- 'SessionRX\_E1\_E20\_SYSTEM' デバッグセッションを選択します。

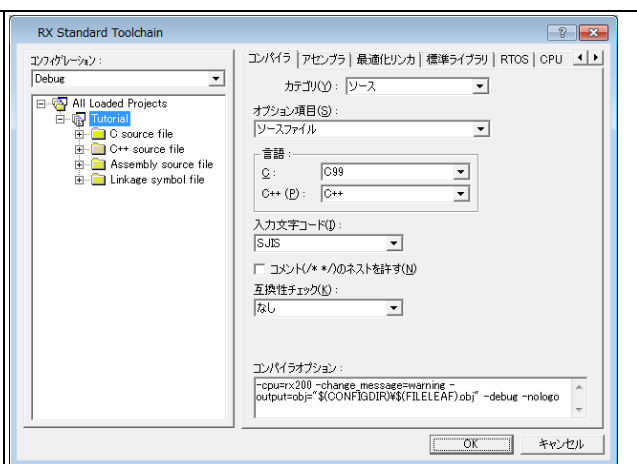


## 5. チュートリアルプログラムのビルド

チュートリアルプロジェクトのビルド設定は、ツールチェーンオプションで既に設定されています。ツールチェーンオプションを表示する為には、'ビルド'メニュー項目のツールチェーンを選択してください。表示されるダイアログは、選択したツールチェーンにより異なります。

コンフィグレーション画面は、全ツールチェーンオプションに存在します。どのような設定を変更する場合でも、変更する部分の現在のコンフィグレーションに注意してください。全てのまたは複数のビルドコンフィグレーションの変更は、'コンフィグレーション'ドロップダウンリストから'All Configurations'または'Multiple Configurations...'を選択することで可能になります。

- 各タブの'カテゴリ'ドロップダウンリストをチェックして、利用可能なオプションを確認してください。ここでは、オプションの変更は不要です。
- 選択終了後に<OK>をクリックしてダイアログを閉じます。



### 5.1 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。

<ul style="list-style-type: none"> <li>• ツールバーの'すべてをビルド'ボタンです。プロジェクト中の全ファイルをビルドします。標準ライブラリは一度だけビルドされます。</li> </ul>	
<ul style="list-style-type: none"> <li>• ツールバーの'ビルド'ボタンです。前回から変更のあった全ファイルをビルドします。オプションを変更しない限り、標準ライブラリはビルドされません。</li> </ul>	
<ul style="list-style-type: none"> <li>• キーボードの'F7'ボタンです。上記の'ビルド'ボタン選択の場合と同じです。</li> </ul>	

ここで、'F7'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ます。ビルド中の各段階で、アウトプット画面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。

## 5.2 エミュレータの接続

本チュートリアルでは、外部から CPU ボードに電源を供給する必要はありません。電源は USB ポートから取得されます。USB ポートに多くのデバイスが接続している場合、Windows がシャットダウンするかもしれないので注意してください。この問題が発生した場合、一部のデバイスを削除して、もう一度やり直してください。外部電源を供給する際、極性および電源電圧が適切であることを必ず確認してください。

このキットに同梱の電圧検出回路（LVD）サンプルコードは電圧検出のために可変電源を必要とします。その場合には、外部電源を使用する必要があります。詳細は RSKRX63T グループユーザズマニュアルを参照してください。

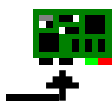
E1 のホストコンピュータへの接続方法は、クイックスタートガイドに詳しく記載されています。以下は、クイックスタートガイドの手順が踏まれ、E1 用のドライバが既にインストールされていることを前提としています。

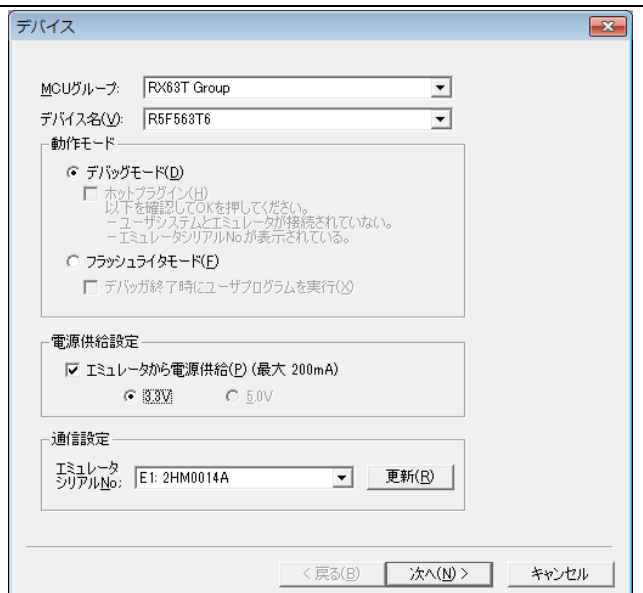
- LCD モジュールを CPU ボードの LCD コネクタに取り付け、コネクタの全てのピンが正しく接続されていることを確認してください。
- E1 をご使用のコンピュータの USB ポートに接続してください。
- E1 を CPU ボードに接続します。'E1'のシルク印字のある E1 コネクタに接続してください。
- 外部電源を CPU ボードに供給する場合、'PWR'のシルク印字のある PWR コネクタに接続してください。

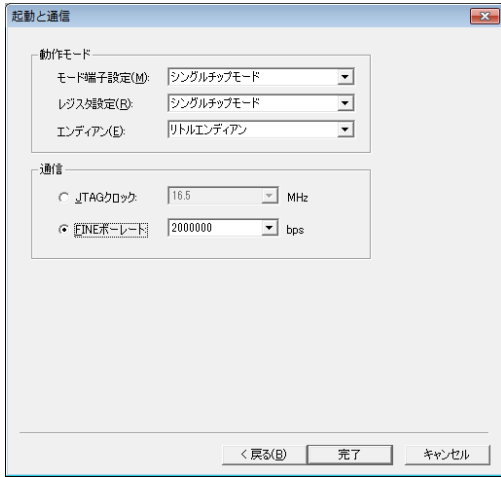

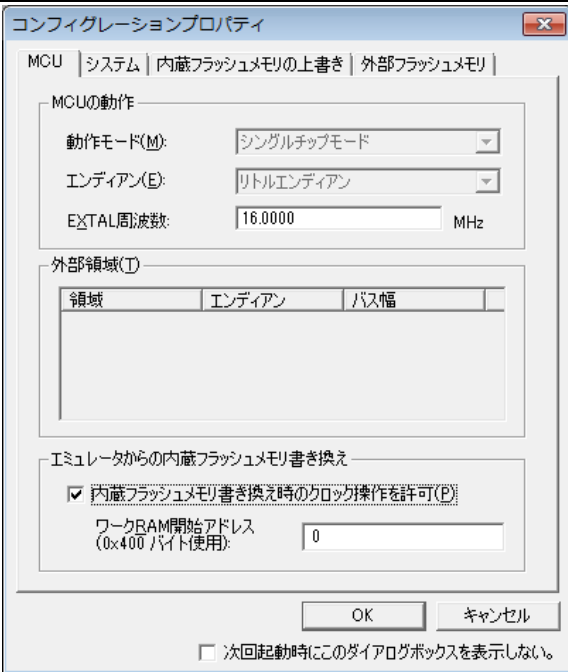
## 5.3 E1 によるターゲットの接続

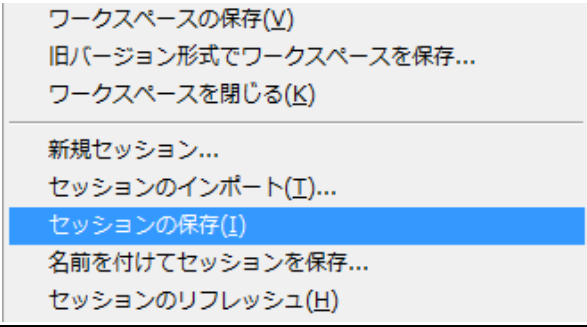
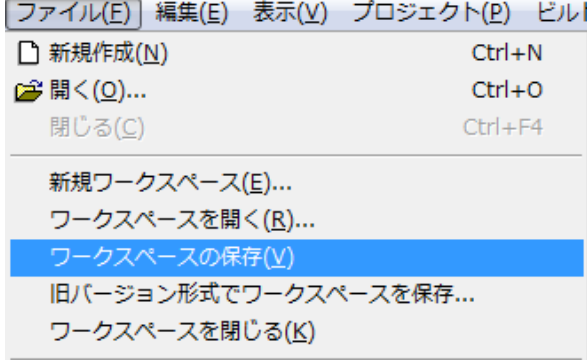
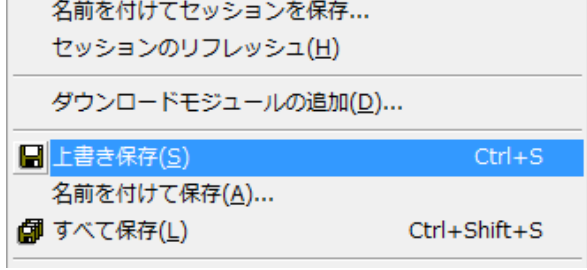
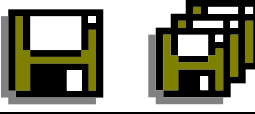
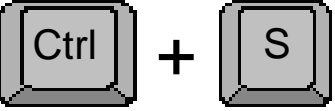
ここでは、デバイスへの接続、フラッシュへのプログラミングおよびコード実行について説明します。

初回接続時と 2 回目以降の接続時とでダイアログ表示が異なります。一部の接続オプションは初回接続時に設定した内容が有効になっていますので、変更の必要がない限り、同じ接続オプションを選択してください。

- 'SessionRX\_E1\_E20\_SYSTEM'デバッグセッションを選択します。
- デバッグツールバーの<接続>ボタンをクリックします。  

- '起動設定'ダイアログが表示されます。以下の通り設定します：
  - MCU グループ: RX63T Group
  - デバイス名: R5F563T6
  - 動作モード: デバッグモード
- E1 が CPU ボードに電源を供給する場合は、'エミュレータから電源供給'を選択し、'3.3V'を選択します。それ以外の場合、適切な電源を接続してください。(詳細については、RSKRX63T のユーザズマニュアルを参照してください。)



<ul style="list-style-type: none"> <li>• ‘起動と通信’タブをクリックします。‘モード端子設定’、‘レジスタ設定’を‘シングルチップモード’に、‘エンディアン’を‘リトルエンディアン’に設定してください。そして‘FINE ボーレート’を‘2000000’bps に設定して&lt;OK&gt;をクリックしてください。</li> <li>• CPU ボードとの接続を開始します。</li> </ul>	
<ul style="list-style-type: none"> <li>• 接続中にプロセスのステータスを示すダイアログが表示されます。初期設定では、接続処理が終了したらダイアログが閉じるように設定されていません。</li> </ul>	
<ul style="list-style-type: none"> <li>• 接続処理が終了したら‘コンフィグレーションプロパティ’ダイアログが表示されます。</li> <li>• 以下の通り設定します： <ul style="list-style-type: none"> <li>— 動作モード：シングルチップモード</li> <li>— エンディアン：リトルエンディアン</li> <li>— EXTAL 周波数：16.0000 MHz</li> <li>— ワーク RAM 開始アドレス：0</li> </ul> </li> <li>• &lt;OK&gt; をクリックしてください。High-performance Embedded Workshop のアウトプット画面に‘Connected’と表示されます。</li> <li>• ツールバー上のデバッグボタンがアクティブになります。これらのボタンの機能は本マニュアル中で説明されます。</li> </ul>	

<p>ここで、High-performance Embedded Workshop のセッションを保存することを推奨します。</p> <ul style="list-style-type: none"> <li>'ファイル'   'セッションの保存'を選択します。</li> </ul>	
<p>ワークスペースの設定を変更した場合、ワークスペースを保存することを推奨します。</p> <ul style="list-style-type: none"> <li>'ファイル'   'ワークスペースの保存'を選択します。</li> </ul>	
<p>ファイルを変更した場合、次の操作でファイルを保存することができます。</p> <ul style="list-style-type: none"> <li>'ファイル'   '上書き保存'を選択します。</li> </ul>	
<p>ツールバーの'上書き保存'ボタンまたは'すべて保存'ボタンでも保存することができます。</p>	
<p>また、キーボードからも保存することができます。</p>	

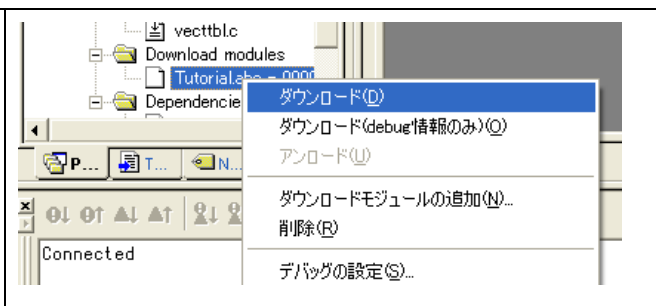
## 6. チュートリアルダウンロードと実行

### 6.1 プログラムコードのダウンロード

High-performance Embedded Workshop でのコード作成が完了したら、それを 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。

この時点でワークスペースビューに'Download Modules'のカテゴリが追加されます。

- ダウンロードモジュールリストから関連するモジュールを右クリックし、'ダウンロード'を選択します。
- ダウンロードが完了すると、コードの実行およびデバッグ準備が整います。



### 6.2 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、プログラムを実行することができます。





## 7. チュートリアルレビュー

本章では、チュートリアルコードがどのように動作し、より複雑なコードへ実装されるためにどのようにそれを変更することができるかを確認します。

### 7.1 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。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。

チュートリアル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。



- チュートリアルコードの開始位置でファイルが開きます。矢印と黄色のハイライトは現在のプログラムカウンタ位置を示します。
- コード表示を下のボタンで'ソースモード'、'逆アセンブリモード'、'混合モード'に切り替えることができます。



← 'ソースモード' ボタン



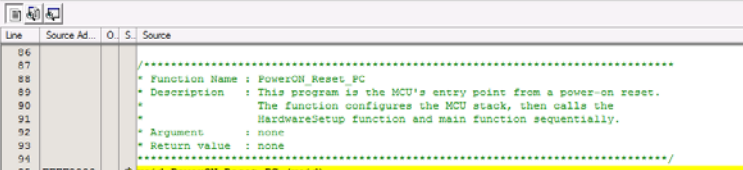

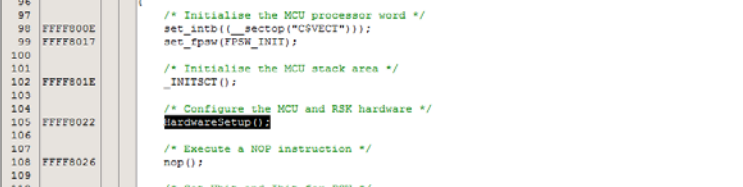

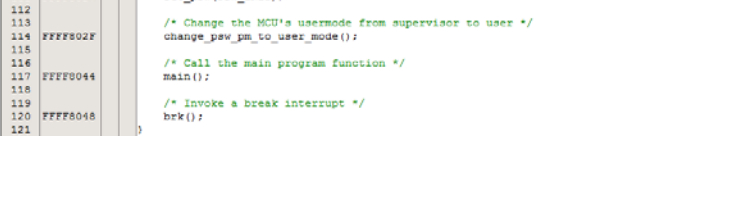
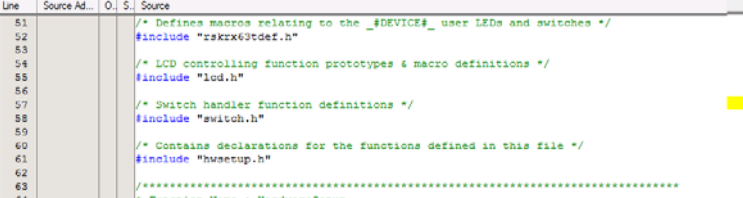

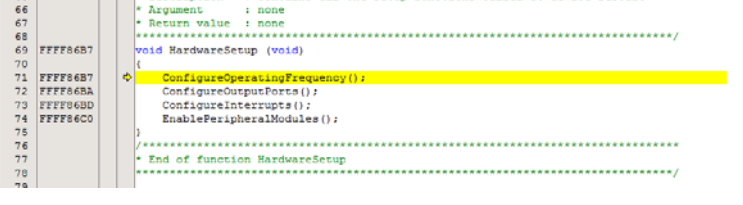
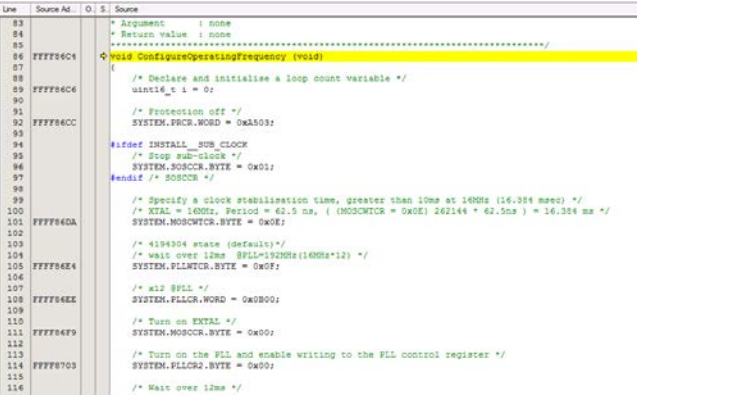
← '混合モード' ボタン



← '逆アセンブリモード' ボタン

コード表示を'ソースモード'に設定してください。

Line	Source Addr...	O.	S.	Source
75				/* Power-on reset function declaration */
76				void PowerON_Reset_FC(void);
77				/* Main program function declaration */
78				void main(void);
79				
80				.....
81				* Outline : PowerON_Reset_FC
82				* Description : This program is the MCU's entry point from a power-on reset.
83				* The function configures the MCU stack, then calls the
84				* HardwareSetup function and main function sequentially.
85				* Argument : none
86				* Return value : none
87				.....
88	FFFF0000			void PowerON_Reset_FC(void)
89				{
90				/* Initialise the MCU processor word */
91	FFFF000E			set_intb(__sectop("CSVECT"));
92				
93				/* Initialise the MCU stack area */
94	FFFF0017			_INITISCT();
95				
96				/* Configure the MCU and RSK hardware */
97	FFFF001B			HardwareSetup();
98				
99				/* Change the MCU's usermode from supervisor to user */
100	FFFF001F			nop();
101	FFFF0020			set_psw(PSW_init);
102	FFFF0028			Change_PSW_EM_to_UserMode();
103				
104				/* Call the main program function */
105	FFFF003D			main();
106				
107				/* Invoke a break interrupt */
108	FFFF0041			brk();
109				}



<ul style="list-style-type: none"> <li>• 'HardwareSetup()'を左クリックでハイライト表示にしてください。</li> </ul>	
<ul style="list-style-type: none"> <li>• デバッグツールバーの'カーソル位置まで実行'をクリックして、選択した行までプログラムを実行させます。</li> </ul> 	
<ul style="list-style-type: none"> <li>• 実行後、'ステップイン'をクリックして、HardwareSetup 関数にエントリします。</li> </ul> 	
<ul style="list-style-type: none"> <li>• プログラムカウンタは HardwareSetup 関数に移ります。この関数はマイクロコントローラが正確にセットアップされるための関数を持ち、メインプログラムが実行される前に実行されます。</li> </ul>	
<ul style="list-style-type: none"> <li>• 'ステップイン'をクリックして ConfigureOperatingFrequency 関数にエントリします。</li> </ul> 	
<ul style="list-style-type: none"> <li>• ConfigureOperatingFrequency 関数はシステムクロックの設定に使用されます。</li> </ul>	

<ul style="list-style-type: none"> <li>• スクリーンショットは初期化されたシステムクロックのリストです。</li> <li>• 次に、HardwareSetup 関数をスキップしてメインプログラムコードを見てみましょう。</li> </ul>	<table border="1"> <thead> <tr> <th>Line</th> <th>Source Ad...</th> <th>O.</th> <th>S.</th> <th>Source</th> </tr> </thead> <tbody> <tr><td>129</td><td></td><td></td><td></td><td>/* Configure the clocks as follows -</td></tr> <tr><td>130</td><td></td><td></td><td></td><td>  Clock Description                  Frequency</td></tr> <tr><td>131</td><td></td><td></td><td></td><td>-----</td></tr> <tr><td>132</td><td></td><td></td><td></td><td>  PLL Clock frequency.....192MHz</td></tr> <tr><td>133</td><td></td><td></td><td></td><td>  System Clock Frequency.....96MHz</td></tr> <tr><td>134</td><td></td><td></td><td></td><td>  Peripheral Module Clock B.....48MHz</td></tr> <tr><td>135</td><td></td><td></td><td></td><td>  Flashif Clock.....48MHz</td></tr> <tr><td>136</td><td></td><td></td><td></td><td>  External Bus Clock.....48MHz */</td></tr> <tr><td>137</td><td></td><td></td><td></td><td>  SYSTEM.SCMCR.LONG = 0x21821211;</td></tr> <tr><td>138</td><td>FFFF8731</td><td></td><td></td><td></td></tr> <tr><td>139</td><td></td><td></td><td></td><td></td></tr> <tr><td>140</td><td></td><td></td><td></td><td>/* Set the clock source to PLL */</td></tr> <tr><td>141</td><td>FFFF873E</td><td></td><td></td><td>  SYSTEM.SCKCR3.WORD = 0x0400;</td></tr> <tr><td>142</td><td></td><td></td><td></td><td></td></tr> <tr><td>143</td><td></td><td></td><td></td><td>/* Protection on */</td></tr> <tr><td>144</td><td>FFFF8749</td><td></td><td></td><td>  SYSTEM.PRCR.WORD = 0xA500;</td></tr> <tr><td>145</td><td></td><td></td><td></td><td>  }</td></tr> <tr><td>146</td><td></td><td></td><td></td><td>  /*****</td></tr> <tr><td>147</td><td></td><td></td><td></td><td>  * End of function ConfigureOperatingFrequency</td></tr> <tr><td>148</td><td></td><td></td><td></td><td>  *****/</td></tr> <tr><td>149</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	Line	Source Ad...	O.	S.	Source	129				/* Configure the clocks as follows -	130				Clock Description                  Frequency	131				-----	132				PLL Clock frequency.....192MHz	133				System Clock Frequency.....96MHz	134				Peripheral Module Clock B.....48MHz	135				Flashif Clock.....48MHz	136				External Bus Clock.....48MHz */	137				SYSTEM.SCMCR.LONG = 0x21821211;	138	FFFF8731				139					140				/* Set the clock source to PLL */	141	FFFF873E			SYSTEM.SCKCR3.WORD = 0x0400;	142					143				/* Protection on */	144	FFFF8749			SYSTEM.PRCR.WORD = 0xA500;	145				}	146				/*****	147				* End of function ConfigureOperatingFrequency	148				*****/	149				
Line	Source Ad...	O.	S.	Source																																																																																																											
129				/* Configure the clocks as follows -																																																																																																											
130				Clock Description                  Frequency																																																																																																											
131				-----																																																																																																											
132				PLL Clock frequency.....192MHz																																																																																																											
133				System Clock Frequency.....96MHz																																																																																																											
134				Peripheral Module Clock B.....48MHz																																																																																																											
135				Flashif Clock.....48MHz																																																																																																											
136				External Bus Clock.....48MHz */																																																																																																											
137				SYSTEM.SCMCR.LONG = 0x21821211;																																																																																																											
138	FFFF8731																																																																																																														
139																																																																																																															
140				/* Set the clock source to PLL */																																																																																																											
141	FFFF873E			SYSTEM.SCKCR3.WORD = 0x0400;																																																																																																											
142																																																																																																															
143				/* Protection on */																																																																																																											
144	FFFF8749			SYSTEM.PRCR.WORD = 0xA500;																																																																																																											
145				}																																																																																																											
146				/*****																																																																																																											
147				* End of function ConfigureOperatingFrequency																																																																																																											
148				*****/																																																																																																											
149																																																																																																															


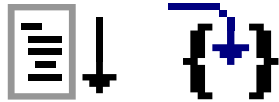
ハードウェア設定に関する詳細情報は、RSKRX63T ユーザーズマニュアルおよび RX63T グループユーザーズマニュアルハードウェア編を参照してください。


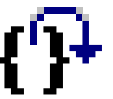
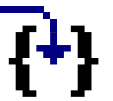
## 7.2 メイン関数



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。

<ul style="list-style-type: none"> <li>画面左にリストアップされている'main.c'をダブルクリック、または右クリックして、'開く main.c'を選択してください。</li> </ul>																																																																																																																																								
<ul style="list-style-type: none"> <li>'main()'に該当するオンチップブレークポイント行をダブルクリックし、ブレークポイントを設定します。</li> <li>2つのブレークポイントが現れますが、これらが同じソースコードアドレスであることを意味します。</li> </ul>	<table border="1"> <thead> <tr> <th>Line</th> <th>Source Ad...</th> <th>O.</th> <th>S.</th> <th>Source</th> </tr> </thead> <tbody> <tr><td>112</td><td></td><td></td><td></td><td></td></tr> <tr><td>113</td><td></td><td></td><td></td><td></td></tr> <tr><td>114</td><td></td><td></td><td></td><td>/* Function Name : main</td></tr> <tr><td>115</td><td></td><td></td><td></td><td>* Description : The main program function. Dis</td></tr> <tr><td>116</td><td></td><td></td><td></td><td>onto the LCD display, then call</td></tr> <tr><td>117</td><td></td><td></td><td></td><td>functions. The function then ca</td></tr> <tr><td>118</td><td></td><td></td><td></td><td>before waiting in an infinite t</td></tr> <tr><td>119</td><td></td><td></td><td></td><td>* Argument : none</td></tr> <tr><td>120</td><td></td><td></td><td></td><td>* Return value : none</td></tr> <tr><td>121</td><td></td><td></td><td></td><td>void main (void)</td></tr> <tr><td>122</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>123</td><td>FFFF8AA0</td><td>●</td><td></td><td>/* Initialise the debug LCD */</td></tr> <tr><td>124</td><td></td><td></td><td></td><td>Init_LCD();</td></tr> <tr><td>125</td><td></td><td></td><td></td><td></td></tr> <tr><td>126</td><td>FFFF8AA0</td><td>●</td><td></td><td>/* Displays the Renesas splash screen */</td></tr> <tr><td>127</td><td></td><td></td><td></td><td>Display_LCD(LCD_LINE1, "Renesas");</td></tr> <tr><td>128</td><td></td><td></td><td></td><td>Display_LCD(LCD_LINE2, NICKNAME);</td></tr> <tr><td>129</td><td>FFFF8AA4</td><td></td><td></td><td></td></tr> <tr><td>130</td><td>FFFF8AB2</td><td></td><td></td><td>/* Begins the initial LED flash sequence */</td></tr> <tr><td>131</td><td></td><td></td><td></td><td>Flash_LED();</td></tr> <tr><td>132</td><td>FFFF8AC1</td><td></td><td></td><td></td></tr> <tr><td>133</td><td></td><td></td><td></td><td>/* Begins the ADC-varying flash Sequence */</td></tr> <tr><td>134</td><td></td><td></td><td></td><td>Timer_ADC();</td></tr> <tr><td>135</td><td></td><td></td><td></td><td></td></tr> <tr><td>136</td><td>FFFF8AC5</td><td></td><td></td><td></td></tr> <tr><td>137</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	Line	Source Ad...	O.	S.	Source	112					113					114				/* Function Name : main	115				* Description : The main program function. Dis	116				onto the LCD display, then call	117				functions. The function then ca	118				before waiting in an infinite t	119				* Argument : none	120				* Return value : none	121				void main (void)	122				{	123	FFFF8AA0	●		/* Initialise the debug LCD */	124				Init_LCD();	125					126	FFFF8AA0	●		/* Displays the Renesas splash screen */	127				Display_LCD(LCD_LINE1, "Renesas");	128				Display_LCD(LCD_LINE2, NICKNAME);	129	FFFF8AA4				130	FFFF8AB2			/* Begins the initial LED flash sequence */	131				Flash_LED();	132	FFFF8AC1				133				/* Begins the ADC-varying flash Sequence */	134				Timer_ADC();	135					136	FFFF8AC5				137				
Line	Source Ad...	O.	S.	Source																																																																																																																																				
112																																																																																																																																								
113																																																																																																																																								
114				/* Function Name : main																																																																																																																																				
115				* Description : The main program function. Dis																																																																																																																																				
116				onto the LCD display, then call																																																																																																																																				
117				functions. The function then ca																																																																																																																																				
118				before waiting in an infinite t																																																																																																																																				
119				* Argument : none																																																																																																																																				
120				* Return value : none																																																																																																																																				
121				void main (void)																																																																																																																																				
122				{																																																																																																																																				
123	FFFF8AA0	●		/* Initialise the debug LCD */																																																																																																																																				
124				Init_LCD();																																																																																																																																				
125																																																																																																																																								
126	FFFF8AA0	●		/* Displays the Renesas splash screen */																																																																																																																																				
127				Display_LCD(LCD_LINE1, "Renesas");																																																																																																																																				
128				Display_LCD(LCD_LINE2, NICKNAME);																																																																																																																																				
129	FFFF8AA4																																																																																																																																							
130	FFFF8AB2			/* Begins the initial LED flash sequence */																																																																																																																																				
131				Flash_LED();																																																																																																																																				
132	FFFF8AC1																																																																																																																																							
133				/* Begins the ADC-varying flash Sequence */																																																																																																																																				
134				Timer_ADC();																																																																																																																																				
135																																																																																																																																								
136	FFFF8AC5																																																																																																																																							
137																																																																																																																																								
<ul style="list-style-type: none"> <li>ブレークポイントは、[F9]キーでも設定することが可能です。</li> </ul> 																																																																																																																																								

E1 エミュレータは本マニュアルでは説明していない高度なイベント機能やブレーク機能を持っています。E1 エミュレータの詳細情報は、RX ファミリ用 E1/E20 エミュレータユーザーズマニュアルを参照してください。

<ul style="list-style-type: none"> <li>デバッグツールバーの‘リセット後実行’をクリックしてください。</li> </ul> 	<table border="1"> <thead> <tr> <th>Line</th> <th>Source Ad...</th> <th>O.</th> <th>S.</th> <th>Source</th> </tr> </thead> <tbody> <tr><td>112</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>113</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>114</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>115</td><td></td><td></td><td></td><td>/* Function Name : main</td></tr> <tr><td>116</td><td></td><td></td><td></td><td>*/</td></tr> <tr><td>117</td><td></td><td></td><td></td><td>/* Description : The main program function. Displays the Renesas splash screen</td></tr> <tr><td>118</td><td></td><td></td><td></td><td>onto the LCD display, then calls the 'flashLED' and 'TimerADC'</td></tr> <tr><td>119</td><td></td><td></td><td></td><td>functions. The function then calls the statics test routine,</td></tr> <tr><td>120</td><td></td><td></td><td></td><td>before waiting in an infinite while loop.</td></tr> <tr><td>121</td><td></td><td></td><td></td><td>*/</td></tr> <tr><td>122</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>123</td><td>FFFF8AA0</td><td>●</td><td></td><td>void main (void)</td></tr> <tr><td>124</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>125</td><td></td><td></td><td></td><td>/* Initialise the debug LCD */</td></tr> <tr><td>126</td><td>FFFF8AA0</td><td>●</td><td>+</td><td>Init_LCD();</td></tr> <tr><td>127</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>128</td><td></td><td></td><td></td><td>/* Displays the Renesas splash screen */</td></tr> <tr><td>129</td><td>FFFF8AA4</td><td></td><td></td><td>Display_LCD(LCD_LINE1, "Renesas");</td></tr> <tr><td>130</td><td>FFFF8AB2</td><td></td><td></td><td>Display_LCD(LCD_LINE2, NICKNAME);</td></tr> <tr><td>131</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>132</td><td></td><td></td><td></td><td>/* Begins the initial LED flash sequence */</td></tr> <tr><td>133</td><td>FFFF8AC1</td><td></td><td></td><td>Flash_LED();</td></tr> <tr><td>134</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>135</td><td></td><td></td><td></td><td>/* Begins the ADC-varying flash Sequence */</td></tr> <tr><td>136</td><td>FFFF8AC5</td><td></td><td></td><td>Timer_ADC();</td></tr> <tr><td>137</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>138</td><td></td><td></td><td></td><td>/* Begins the static variable test */</td></tr> <tr><td>139</td><td>FFFF8AC9</td><td></td><td></td><td>static_test();</td></tr> <tr><td>140</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>141</td><td></td><td></td><td></td><td>/* Infinite while loop */</td></tr> <tr><td>142</td><td>FFFF8ACC</td><td></td><td></td><td>while (1)</td></tr> <tr><td>143</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>144</td><td></td><td></td><td></td><td>/* Do nothing */</td></tr> <tr><td>145</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>146</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>147</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>148</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>149</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>150</td><td></td><td></td><td></td><td>.....</td></tr> </tbody> </table>	Line	Source Ad...	O.	S.	Source	112				.....	113				.....	114				.....	115				/* Function Name : main	116				*/	117				/* Description : The main program function. Displays the Renesas splash screen	118				onto the LCD display, then calls the 'flashLED' and 'TimerADC'	119				functions. The function then calls the statics test routine,	120				before waiting in an infinite while loop.	121				*/	122				.....	123	FFFF8AA0	●		void main (void)	124				{	125				/* Initialise the debug LCD */	126	FFFF8AA0	●	+	Init_LCD();	127				.....	128				/* Displays the Renesas splash screen */	129	FFFF8AA4			Display_LCD(LCD_LINE1, "Renesas");	130	FFFF8AB2			Display_LCD(LCD_LINE2, NICKNAME);	131				.....	132				/* Begins the initial LED flash sequence */	133	FFFF8AC1			Flash_LED();	134				.....	135				/* Begins the ADC-varying flash Sequence */	136	FFFF8AC5			Timer_ADC();	137				.....	138				/* Begins the static variable test */	139	FFFF8AC9			static_test();	140				.....	141				/* Infinite while loop */	142	FFFF8ACC			while (1)	143				{	144				/* Do nothing */	145				}	146				.....	147				.....	148				.....	149				.....	150				.....
Line	Source Ad...	O.	S.	Source																																																																																																																																																																																																					
112				.....																																																																																																																																																																																																					
113				.....																																																																																																																																																																																																					
114				.....																																																																																																																																																																																																					
115				/* Function Name : main																																																																																																																																																																																																					
116				*/																																																																																																																																																																																																					
117				/* Description : The main program function. Displays the Renesas splash screen																																																																																																																																																																																																					
118				onto the LCD display, then calls the 'flashLED' and 'TimerADC'																																																																																																																																																																																																					
119				functions. The function then calls the statics test routine,																																																																																																																																																																																																					
120				before waiting in an infinite while loop.																																																																																																																																																																																																					
121				*/																																																																																																																																																																																																					
122				.....																																																																																																																																																																																																					
123	FFFF8AA0	●		void main (void)																																																																																																																																																																																																					
124				{																																																																																																																																																																																																					
125				/* Initialise the debug LCD */																																																																																																																																																																																																					
126	FFFF8AA0	●	+	Init_LCD();																																																																																																																																																																																																					
127				.....																																																																																																																																																																																																					
128				/* Displays the Renesas splash screen */																																																																																																																																																																																																					
129	FFFF8AA4			Display_LCD(LCD_LINE1, "Renesas");																																																																																																																																																																																																					
130	FFFF8AB2			Display_LCD(LCD_LINE2, NICKNAME);																																																																																																																																																																																																					
131				.....																																																																																																																																																																																																					
132				/* Begins the initial LED flash sequence */																																																																																																																																																																																																					
133	FFFF8AC1			Flash_LED();																																																																																																																																																																																																					
134				.....																																																																																																																																																																																																					
135				/* Begins the ADC-varying flash Sequence */																																																																																																																																																																																																					
136	FFFF8AC5			Timer_ADC();																																																																																																																																																																																																					
137				.....																																																																																																																																																																																																					
138				/* Begins the static variable test */																																																																																																																																																																																																					
139	FFFF8AC9			static_test();																																																																																																																																																																																																					
140				.....																																																																																																																																																																																																					
141				/* Infinite while loop */																																																																																																																																																																																																					
142	FFFF8ACC			while (1)																																																																																																																																																																																																					
143				{																																																																																																																																																																																																					
144				/* Do nothing */																																																																																																																																																																																																					
145				}																																																																																																																																																																																																					
146				.....																																																																																																																																																																																																					
147				.....																																																																																																																																																																																																					
148				.....																																																																																																																																																																																																					
149				.....																																																																																																																																																																																																					
150				.....																																																																																																																																																																																																					
<ul style="list-style-type: none"> <li>Flash_LED関数、Timer_ADC関数および statics_test関数にブレークポイントを設定してください。</li> </ul>	<table border="1"> <thead> <tr> <th>Line</th> <th>Source Ad...</th> <th>O.</th> <th>S.</th> <th>Source</th> </tr> </thead> <tbody> <tr><td>132</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>133</td><td>FFFF8AC1</td><td>●</td><td></td><td>Flash_LED();</td></tr> <tr><td>134</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>135</td><td></td><td></td><td></td><td>/* Begins the ADC-varying flash Sequence */</td></tr> <tr><td>136</td><td>FFFF8AC5</td><td>●</td><td></td><td>Timer_ADC();</td></tr> <tr><td>137</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>138</td><td></td><td></td><td></td><td>/* Begins the static variable test */</td></tr> <tr><td>139</td><td>FFFF8AC9</td><td>●</td><td></td><td>static_test();</td></tr> <tr><td>140</td><td></td><td></td><td></td><td>.....</td></tr> </tbody> </table>	Line	Source Ad...	O.	S.	Source	132				.....	133	FFFF8AC1	●		Flash_LED();	134				.....	135				/* Begins the ADC-varying flash Sequence */	136	FFFF8AC5	●		Timer_ADC();	137				.....	138				/* Begins the static variable test */	139	FFFF8AC9	●		static_test();	140				.....																																																																																																																																																						
Line	Source Ad...	O.	S.	Source																																																																																																																																																																																																					
132				.....																																																																																																																																																																																																					
133	FFFF8AC1	●		Flash_LED();																																																																																																																																																																																																					
134				.....																																																																																																																																																																																																					
135				/* Begins the ADC-varying flash Sequence */																																																																																																																																																																																																					
136	FFFF8AC5	●		Timer_ADC();																																																																																																																																																																																																					
137				.....																																																																																																																																																																																																					
138				/* Begins the static variable test */																																																																																																																																																																																																					
139	FFFF8AC9	●		static_test();																																																																																																																																																																																																					
140				.....																																																																																																																																																																																																					
<ul style="list-style-type: none"> <li>デバッグツールバーの‘Go’をクリックし、プログラム停止後に‘ステップイン’をクリックしてください。プログラムカウンタはFlash_LED関数の先頭に移ります。</li> </ul> 	<table border="1"> <thead> <tr> <th>Line</th> <th>Source Ad...</th> <th>O.</th> <th>S.</th> <th>Source</th> </tr> </thead> <tbody> <tr><td>58</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>59</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>60</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>61</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>62</td><td></td><td></td><td></td><td>/* Function Name : Flash_LED</td></tr> <tr><td>63</td><td></td><td></td><td></td><td>*/</td></tr> <tr><td>64</td><td></td><td></td><td></td><td>/* Description : The LED flash function used at the beginning of the program</td></tr> <tr><td>65</td><td></td><td></td><td></td><td>*/</td></tr> <tr><td>66</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>67</td><td>FFFF85A0</td><td></td><td></td><td>void Flash_LED (void)</td></tr> <tr><td>68</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>69</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>70</td><td></td><td></td><td></td><td>/* Variable used to count down the number of LED flashes */</td></tr> <tr><td>71</td><td></td><td></td><td></td><td>static uint16_t flash_count = 0xC8;</td></tr> <tr><td>72</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>73</td><td></td><td></td><td></td><td>/* Declare a delay count variable */</td></tr> <tr><td>74</td><td>FFFF85A2</td><td>+</td><td></td><td>uint32_t ulLed_Delay = 0;</td></tr> <tr><td>75</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>76</td><td>FFFF85A8</td><td></td><td></td><td>/* Flash the LEDs for 200 times or until a user switch is pressed */</td></tr> <tr><td>77</td><td></td><td></td><td></td><td>while ((0 == g_switch_flag) &amp;&amp; (--flash_count &gt; 0))</td></tr> <tr><td>78</td><td>FFFF85DB</td><td></td><td></td><td>{</td></tr> <tr><td>79</td><td></td><td></td><td></td><td>for (ulLed_Delay = 0; ulLed_Delay &lt; 4000000; ++ulLed_Delay)</td></tr> <tr><td>80</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>81</td><td></td><td></td><td></td><td>/* delay */</td></tr> <tr><td>82</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>83</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>84</td><td>FFFF85FD</td><td></td><td></td><td>/* Toggles the LEDs after a specific delay. */</td></tr> <tr><td>85</td><td></td><td></td><td></td><td>Toggle_LED();</td></tr> <tr><td>86</td><td></td><td></td><td></td><td>}</td></tr> <tr><td>87</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>88</td><td>FFFF8633</td><td></td><td></td><td>/* Reset the g_switch_flag flag variable */</td></tr> <tr><td>89</td><td></td><td></td><td></td><td>g_switch_flag = 0;</td></tr> <tr><td>90</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>91</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>92</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>93</td><td></td><td></td><td></td><td>.....</td></tr> </tbody> </table>	Line	Source Ad...	O.	S.	Source	58				.....	59				.....	60				.....	61				.....	62				/* Function Name : Flash_LED	63				*/	64				/* Description : The LED flash function used at the beginning of the program	65				*/	66				.....	67	FFFF85A0			void Flash_LED (void)	68				{	69				.....	70				/* Variable used to count down the number of LED flashes */	71				static uint16_t flash_count = 0xC8;	72				.....	73				/* Declare a delay count variable */	74	FFFF85A2	+		uint32_t ulLed_Delay = 0;	75				.....	76	FFFF85A8			/* Flash the LEDs for 200 times or until a user switch is pressed */	77				while ((0 == g_switch_flag) && (--flash_count > 0))	78	FFFF85DB			{	79				for (ulLed_Delay = 0; ulLed_Delay < 4000000; ++ulLed_Delay)	80				{	81				/* delay */	82				}	83				.....	84	FFFF85FD			/* Toggles the LEDs after a specific delay. */	85				Toggle_LED();	86				}	87				.....	88	FFFF8633			/* Reset the g_switch_flag flag variable */	89				g_switch_flag = 0;	90				.....	91				.....	92				.....	93				.....															
Line	Source Ad...	O.	S.	Source																																																																																																																																																																																																					
58				.....																																																																																																																																																																																																					
59				.....																																																																																																																																																																																																					
60				.....																																																																																																																																																																																																					
61				.....																																																																																																																																																																																																					
62				/* Function Name : Flash_LED																																																																																																																																																																																																					
63				*/																																																																																																																																																																																																					
64				/* Description : The LED flash function used at the beginning of the program																																																																																																																																																																																																					
65				*/																																																																																																																																																																																																					
66				.....																																																																																																																																																																																																					
67	FFFF85A0			void Flash_LED (void)																																																																																																																																																																																																					
68				{																																																																																																																																																																																																					
69				.....																																																																																																																																																																																																					
70				/* Variable used to count down the number of LED flashes */																																																																																																																																																																																																					
71				static uint16_t flash_count = 0xC8;																																																																																																																																																																																																					
72				.....																																																																																																																																																																																																					
73				/* Declare a delay count variable */																																																																																																																																																																																																					
74	FFFF85A2	+		uint32_t ulLed_Delay = 0;																																																																																																																																																																																																					
75				.....																																																																																																																																																																																																					
76	FFFF85A8			/* Flash the LEDs for 200 times or until a user switch is pressed */																																																																																																																																																																																																					
77				while ((0 == g_switch_flag) && (--flash_count > 0))																																																																																																																																																																																																					
78	FFFF85DB			{																																																																																																																																																																																																					
79				for (ulLed_Delay = 0; ulLed_Delay < 4000000; ++ulLed_Delay)																																																																																																																																																																																																					
80				{																																																																																																																																																																																																					
81				/* delay */																																																																																																																																																																																																					
82				}																																																																																																																																																																																																					
83				.....																																																																																																																																																																																																					
84	FFFF85FD			/* Toggles the LEDs after a specific delay. */																																																																																																																																																																																																					
85				Toggle_LED();																																																																																																																																																																																																					
86				}																																																																																																																																																																																																					
87				.....																																																																																																																																																																																																					
88	FFFF8633			/* Reset the g_switch_flag flag variable */																																																																																																																																																																																																					
89				g_switch_flag = 0;																																																																																																																																																																																																					
90				.....																																																																																																																																																																																																					
91				.....																																																																																																																																																																																																					
92				.....																																																																																																																																																																																																					
93				.....																																																																																																																																																																																																					
<ul style="list-style-type: none"> <li>Flash_LED関数は周期的にLEDをトグル出力します。</li> <li>while文でg_switch_flag変数（スイッチ押下検出用）とflash_count変数（LEDの点滅カウント用）の値をチェックします。スイッチが押されるか、またはLED点滅カウントが0になると、関数はwhileループを抜けます。</li> </ul>																																																																																																																																																																																																									

<ul style="list-style-type: none"> <li>• 'Go'をクリックしてプログラムを再開し、ボード上の任意のスイッチを押してください。プログラムは Timer_ADC 関数上のブレークポイントで停止します。</li> </ul>	<table border="1"> <thead> <tr> <th>Line</th> <th>Source Ad...</th> <th>O.</th> <th>S.</th> <th>Source</th> </tr> </thead> <tbody> <tr><td>106</td><td>FFFF942A</td><td></td><td></td><td>static void start_timer (void)</td></tr> <tr><td>107</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>108</td><td></td><td></td><td></td><td>/* Protection off */</td></tr> <tr><td>109</td><td>FFFF942A</td><td>→</td><td></td><td>SYSTEM.PRCR.WORD = 0xA503;</td></tr> <tr><td>110</td><td></td><td></td><td></td><td></td></tr> <tr><td>111</td><td></td><td></td><td></td><td>/* Cancel the CMT1 module clock stop mode */</td></tr> <tr><td>112</td><td>FFFF9438</td><td></td><td></td><td>MSTP_CMT1 = 0;</td></tr> <tr><td>113</td><td></td><td></td><td></td><td></td></tr> <tr><td>114</td><td></td><td></td><td></td><td>/* Cancel the CMT2 module clock stop mode */</td></tr> <tr><td>115</td><td>FFFF9454</td><td></td><td></td><td>MSTP_CMT2 = 0;</td></tr> <tr><td>116</td><td></td><td></td><td></td><td></td></tr> <tr><td>117</td><td></td><td></td><td></td><td>/* Protection on */</td></tr> <tr><td>118</td><td>FFFF946F</td><td></td><td></td><td>SYSTEM.PRCR.WORD = 0xA500;</td></tr> <tr><td>119</td><td></td><td></td><td></td><td></td></tr> <tr><td>120</td><td></td><td></td><td></td><td>/* Set CMT1 interrupt priority level to 10 */</td></tr> <tr><td>121</td><td>FFFF947D</td><td></td><td></td><td>IPR(CMT1,CMI1) = 0xA;</td></tr> <tr><td>122</td><td></td><td></td><td></td><td></td></tr> <tr><td>123</td><td></td><td></td><td></td><td>/* Enable CMT1 interrupts */</td></tr> <tr><td>124</td><td>FFFF9499</td><td></td><td></td><td>IEN(CMT1,CMI1) = 0x1;</td></tr> <tr><td>125</td><td></td><td></td><td></td><td></td></tr> <tr><td>126</td><td></td><td></td><td></td><td>/* Clear CMT1 interrupt flag */</td></tr> <tr><td>127</td><td>FFFF94A9</td><td></td><td></td><td>IR(CMT1,CMI1) = 0x0;</td></tr> <tr><td>128</td><td></td><td></td><td></td><td></td></tr> <tr><td>129</td><td></td><td></td><td></td><td>/* Set CMT2 interrupt priority level to 5 */</td></tr> <tr><td>130</td><td>FFFF94B8</td><td></td><td></td><td>IPR(CMT2,CMI2) = 0x5;</td></tr> </tbody> </table>	Line	Source Ad...	O.	S.	Source	106	FFFF942A			static void start_timer (void)	107				{	108				/* Protection off */	109	FFFF942A	→		SYSTEM.PRCR.WORD = 0xA503;	110					111				/* Cancel the CMT1 module clock stop mode */	112	FFFF9438			MSTP_CMT1 = 0;	113					114				/* Cancel the CMT2 module clock stop mode */	115	FFFF9454			MSTP_CMT2 = 0;	116					117				/* Protection on */	118	FFFF946F			SYSTEM.PRCR.WORD = 0xA500;	119					120				/* Set CMT1 interrupt priority level to 10 */	121	FFFF947D			IPR(CMT1,CMI1) = 0xA;	122					123				/* Enable CMT1 interrupts */	124	FFFF9499			IEN(CMT1,CMI1) = 0x1;	125					126				/* Clear CMT1 interrupt flag */	127	FFFF94A9			IR(CMT1,CMI1) = 0x0;	128					129				/* Set CMT2 interrupt priority level to 5 */	130	FFFF94B8			IPR(CMT2,CMI2) = 0x5;																																																		
Line	Source Ad...	O.	S.	Source																																																																																																																																																																																	
106	FFFF942A			static void start_timer (void)																																																																																																																																																																																	
107				{																																																																																																																																																																																	
108				/* Protection off */																																																																																																																																																																																	
109	FFFF942A	→		SYSTEM.PRCR.WORD = 0xA503;																																																																																																																																																																																	
110																																																																																																																																																																																					
111				/* Cancel the CMT1 module clock stop mode */																																																																																																																																																																																	
112	FFFF9438			MSTP_CMT1 = 0;																																																																																																																																																																																	
113																																																																																																																																																																																					
114				/* Cancel the CMT2 module clock stop mode */																																																																																																																																																																																	
115	FFFF9454			MSTP_CMT2 = 0;																																																																																																																																																																																	
116																																																																																																																																																																																					
117				/* Protection on */																																																																																																																																																																																	
118	FFFF946F			SYSTEM.PRCR.WORD = 0xA500;																																																																																																																																																																																	
119																																																																																																																																																																																					
120				/* Set CMT1 interrupt priority level to 10 */																																																																																																																																																																																	
121	FFFF947D			IPR(CMT1,CMI1) = 0xA;																																																																																																																																																																																	
122																																																																																																																																																																																					
123				/* Enable CMT1 interrupts */																																																																																																																																																																																	
124	FFFF9499			IEN(CMT1,CMI1) = 0x1;																																																																																																																																																																																	
125																																																																																																																																																																																					
126				/* Clear CMT1 interrupt flag */																																																																																																																																																																																	
127	FFFF94A9			IR(CMT1,CMI1) = 0x0;																																																																																																																																																																																	
128																																																																																																																																																																																					
129				/* Set CMT2 interrupt priority level to 5 */																																																																																																																																																																																	
130	FFFF94B8			IPR(CMT2,CMI2) = 0x5;																																																																																																																																																																																	
<ul style="list-style-type: none"> <li>• 'ステップイン'をクリックして start_timer 関数にエントリします。</li> </ul> 	<ul style="list-style-type: none"> <li>• start_timer 関数はボード上のポテンシオメータ RV1 による LED 点滅間隔の調整を可能にするために、タイマと A/D コンバータの両方を設定します。</li> </ul>																																																																																																																																																																																				
<ul style="list-style-type: none"> <li>• 'ステップアウト'をクリックして start_timer 関数を抜けます。</li> </ul> 	<table border="1"> <thead> <tr> <th>Line</th> <th>Source Ad...</th> <th>O.</th> <th>S.</th> <th>Source</th> </tr> </thead> <tbody> <tr><td>154</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>155</td><td></td><td></td><td></td><td>/* Function Name : start_adc</td></tr> <tr><td>156</td><td></td><td></td><td></td><td>* Description : Initialises the S12AD ADC module's channel AN000 for</td></tr> <tr><td>157</td><td></td><td></td><td></td><td>* continuous scan operations, then starts the conversions.</td></tr> <tr><td>158</td><td></td><td></td><td></td><td>* Argument : none</td></tr> <tr><td>159</td><td></td><td></td><td></td><td>* Return value : none</td></tr> <tr><td>160</td><td></td><td></td><td></td><td>.....</td></tr> <tr><td>161</td><td>FFFF955B</td><td></td><td></td><td>static void start_adc (void)</td></tr> <tr><td>162</td><td></td><td></td><td></td><td>{</td></tr> <tr><td>163</td><td></td><td></td><td></td><td>/* Protection off */</td></tr> <tr><td>164</td><td>FFFF955B</td><td>→</td><td></td><td>SYSTEM.PRCR.WORD = 0xA503;</td></tr> <tr><td>165</td><td></td><td></td><td></td><td></td></tr> <tr><td>166</td><td></td><td></td><td></td><td>/* Cancel the S12AD module clock stop mode */</td></tr> <tr><td>167</td><td>FFFF9569</td><td></td><td></td><td>MSTP_S12AD = 0;</td></tr> <tr><td>168</td><td></td><td></td><td></td><td></td></tr> <tr><td>169</td><td></td><td></td><td></td><td>/* Protection on */</td></tr> <tr><td>170</td><td>FFFF9585</td><td></td><td></td><td>SYSTEM.PRCR.WORD = 0xA500;</td></tr> <tr><td>171</td><td></td><td></td><td></td><td></td></tr> <tr><td>172</td><td></td><td></td><td></td><td>/* Clear the S12AD interrupt flag */</td></tr> <tr><td>173</td><td>FFFF9593</td><td></td><td></td><td>IR(S12AD, S12ADI) = 0x0;</td></tr> <tr><td>174</td><td></td><td></td><td></td><td></td></tr> <tr><td>175</td><td></td><td></td><td></td><td>/* Set the S12AD interrupt level to 5 */</td></tr> <tr><td>176</td><td>FFFF95A2</td><td></td><td></td><td>IPR(S12AD, S12ADI) = 0x5;</td></tr> <tr><td>177</td><td></td><td></td><td></td><td></td></tr> <tr><td>178</td><td></td><td></td><td></td><td>/* Enable S12AD interrupt requests */</td></tr> <tr><td>179</td><td>FFFF95BE</td><td></td><td></td><td>IEN(S12AD, S12ADI) = 0x1;</td></tr> <tr><td>180</td><td></td><td></td><td></td><td></td></tr> <tr><td>181</td><td></td><td></td><td></td><td>/* Use the AN000 (Potentiometer) pin, P4.0, P(56)</td></tr> <tr><td>182</td><td></td><td></td><td></td><td>as an I/O for peripheral functions */</td></tr> <tr><td>183</td><td></td><td></td><td></td><td>/* Enable Group B scan end Interrupt */</td></tr> <tr><td>184</td><td>FFFF95CC</td><td></td><td></td><td>S12AD.ADCSR.WORD  = 0x40;</td></tr> <tr><td>185</td><td></td><td></td><td></td><td></td></tr> <tr><td>186</td><td></td><td></td><td></td><td>/* Selects AN000 */</td></tr> <tr><td>187</td><td>FFFF95DA</td><td></td><td></td><td>S12AD.ADANSA.WORD = 0x0001;</td></tr> <tr><td>188</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	Line	Source Ad...	O.	S.	Source	154				.....	155				/* Function Name : start_adc	156				* Description : Initialises the S12AD ADC module's channel AN000 for	157				* continuous scan operations, then starts the conversions.	158				* Argument : none	159				* Return value : none	160				.....	161	FFFF955B			static void start_adc (void)	162				{	163				/* Protection off */	164	FFFF955B	→		SYSTEM.PRCR.WORD = 0xA503;	165					166				/* Cancel the S12AD module clock stop mode */	167	FFFF9569			MSTP_S12AD = 0;	168					169				/* Protection on */	170	FFFF9585			SYSTEM.PRCR.WORD = 0xA500;	171					172				/* Clear the S12AD interrupt flag */	173	FFFF9593			IR(S12AD, S12ADI) = 0x0;	174					175				/* Set the S12AD interrupt level to 5 */	176	FFFF95A2			IPR(S12AD, S12ADI) = 0x5;	177					178				/* Enable S12AD interrupt requests */	179	FFFF95BE			IEN(S12AD, S12ADI) = 0x1;	180					181				/* Use the AN000 (Potentiometer) pin, P4.0, P(56)	182				as an I/O for peripheral functions */	183				/* Enable Group B scan end Interrupt */	184	FFFF95CC			S12AD.ADCSR.WORD  = 0x40;	185					186				/* Selects AN000 */	187	FFFF95DA			S12AD.ADANSA.WORD = 0x0001;	188				
Line	Source Ad...	O.	S.	Source																																																																																																																																																																																	
154				.....																																																																																																																																																																																	
155				/* Function Name : start_adc																																																																																																																																																																																	
156				* Description : Initialises the S12AD ADC module's channel AN000 for																																																																																																																																																																																	
157				* continuous scan operations, then starts the conversions.																																																																																																																																																																																	
158				* Argument : none																																																																																																																																																																																	
159				* Return value : none																																																																																																																																																																																	
160				.....																																																																																																																																																																																	
161	FFFF955B			static void start_adc (void)																																																																																																																																																																																	
162				{																																																																																																																																																																																	
163				/* Protection off */																																																																																																																																																																																	
164	FFFF955B	→		SYSTEM.PRCR.WORD = 0xA503;																																																																																																																																																																																	
165																																																																																																																																																																																					
166				/* Cancel the S12AD module clock stop mode */																																																																																																																																																																																	
167	FFFF9569			MSTP_S12AD = 0;																																																																																																																																																																																	
168																																																																																																																																																																																					
169				/* Protection on */																																																																																																																																																																																	
170	FFFF9585			SYSTEM.PRCR.WORD = 0xA500;																																																																																																																																																																																	
171																																																																																																																																																																																					
172				/* Clear the S12AD interrupt flag */																																																																																																																																																																																	
173	FFFF9593			IR(S12AD, S12ADI) = 0x0;																																																																																																																																																																																	
174																																																																																																																																																																																					
175				/* Set the S12AD interrupt level to 5 */																																																																																																																																																																																	
176	FFFF95A2			IPR(S12AD, S12ADI) = 0x5;																																																																																																																																																																																	
177																																																																																																																																																																																					
178				/* Enable S12AD interrupt requests */																																																																																																																																																																																	
179	FFFF95BE			IEN(S12AD, S12ADI) = 0x1;																																																																																																																																																																																	
180																																																																																																																																																																																					
181				/* Use the AN000 (Potentiometer) pin, P4.0, P(56)																																																																																																																																																																																	
182				as an I/O for peripheral functions */																																																																																																																																																																																	
183				/* Enable Group B scan end Interrupt */																																																																																																																																																																																	
184	FFFF95CC			S12AD.ADCSR.WORD  = 0x40;																																																																																																																																																																																	
185																																																																																																																																																																																					
186				/* Selects AN000 */																																																																																																																																																																																	
187	FFFF95DA			S12AD.ADANSA.WORD = 0x0001;																																																																																																																																																																																	
188																																																																																																																																																																																					
<ul style="list-style-type: none"> <li>• その後、'ステップイン'をクリックするとプログラムは start_adc 関数に到達します。</li> </ul> 	<ul style="list-style-type: none"> <li>• start_adc 関数はボード上のポテンシオメータ RV1 の入力を繰り返し変換が行えるように ADC ユニットを設定します。</li> </ul>																																																																																																																																																																																				

<ul style="list-style-type: none"> <li>• 'F5'を押してプログラムを再開させます。プログラムは statics_test 関数上のブレークポイントで停止します。</li> </ul> 	<pre> 152 Source Ad... O: S Source 153 /***** 154 * Function Name : static_test 155 * Description : Static variable test routine. The function replaces the 156 * contents of the string 'cStr' with that of 'ucReplace', one 157 * element at a time. Right-click the variable 'cStr', and 158 * select 'instant watch' - click add in the subsequent dialog. 159 * If you step through the function, you can watch the string 160 * elements being overwritten with the new data. 161 * Argument : none 162 * Return value : none 163 *****/ 164 static void static_test (void) 165 { 166     /* Declare loop count variable */ 167     uint8_t uicount = 0; 168     /* Write cStr variable, "STATIC" to LCD */ 169     Display_LCD(LCD_LINE2, cStr); 170 171     /* Begin for loop which writes one letter of ucReplace to the LCD at a time 172     The nested while loops generate the delay between each letter change */ 173     for (uicount = 0; uicount &lt; 8; uicount++) 174     { 175         /* Start a one-shot timer to create a delay between each loop 176         iteration */ 177         Timer_Delay(40, MILLISECS); 178 179         /* Replace letter number 'uicount' of 'ucStr' from 'ucReplace' */ 180         cStr[uicount] = ucReplace[uicount]; 181         Display_LCD(LCD_LINE2, cStr); 182     } 183 184     /* Clear LCD Display */ 185     cStr[uicount] = '\0'; 186 187     /* Add a delay before overwriting the string */ 188     Timer_Delay(40, MILLISECS); 189 </pre>
<ul style="list-style-type: none"> <li>• 'F11'を押して関数の先頭へ移ります。</li> </ul> 	
<ul style="list-style-type: none"> <li>• statics_test 関数は静的変数の文字ストリングを初期化します。初期化後、別の文字ストリングに置き換えます。</li> <li>• '実行'をクリックするか'F5'を押してプログラムを再開させます。LCD の 2 行目の文字が「STATIC」から一文字ごとに「TESTTEST」に置き換わることが確認できます。その後、プログラムは 2 行目の文字を「RX63T-L」に戻します。</li> </ul>	

## 8. 追加情報

### サポート

High-performance Embedded Workshop の詳細情報は、CD またはウェブサイトに掲載のマニュアルを参照してください。

RX63T グループ マイクロコントローラに関する詳細情報は、RX63T グループユーザ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、情報等は以下のウェブサイトより入手可能です：

<http://japan.renesas.com/rskrx63t> (日本サイト)  
<http://www.renesas.com/rskrx63t> (グローバルサイト)

### オンライン技術サポート

技術関連の問合せは、以下を通じてお願いいたします。

日本：[csc@renesas.com](mailto:csc@renesas.com)  
グローバル：[csc@renesas.com](mailto:csc@renesas.com)

ルネサスのマイクロコントローラ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)

### 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、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。

© 2013 (2014) Renesas Electronics Europe Limited. All rights reserved.  
© 2013 (2014) Renesas Electronics Corporation. All rights reserved.  
© 2013 (2014) Renesas Solutions Corp. All rights reserved.



改訂記録	RSKRX63T 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	2013.01.31	－	初版発行
2.00	2014.04.14	－	全面改訂

---

RSKRX63T チュートリアルマニュアル

発行年月日 2014年4月14日 Rev.2.00

発行 株式会社ルネサスソリューションズ  
〒532-0003 大阪府大阪市淀川区宮原 4-1-6

---



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/>

RX63T グループ