

RX63T グループ

Renesas Starter Kit チュートリアルマニュアル (CubeSuite+)

ルネサス 32 ビットマイクロコンピュータ
RX ファミリー/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。

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 CubeSuite+ 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 CubeSuite+ 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。

RSKRX63T144 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	RSKRX63T144 ユーザーズマニュアル	R20UT2117JG
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	RSKRX63T144 チュートリアルマニュアル	R20UT2118JG (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	RSKRX63T144 クイックスタートガイド	R20UT2119JG
回路図	CPU ボードの回路図	RSKRX63T144 CPU ボード回路図	R20UT2116EG
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
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
CPU	Central Processing Unit	中央処理装置
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
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
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
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
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
TAU	Timer Array Unit	タイマアレイユニット
TFT	Thin Film Transistor	薄膜トランジスタ
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
3.1 はじめに	9
3.2 CubeSuite+の開始と E1 エミュレータの接続	9
3.3 デバッグ・ツールの設定	11
3.4 ビルド設定	12
4. チュートリアルプログラムのビルド	13
4.1 コードのビルド	13
4.2 エミュレータの接続	14
4.3 プロジェクトの保存	14
5. チュートリアルのダウンロードと実行	15
5.1 プログラムコードのダウンロード	15
5.2 コードの実行	15
6. チュートリアルレビュー	16
6.1 プログラム初期化	16
6.2 メイン関数	17
7. 追加情報	20

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 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' はデバッグ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' はデバッグ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。

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、CubeSuite+、コンパイラまたは 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

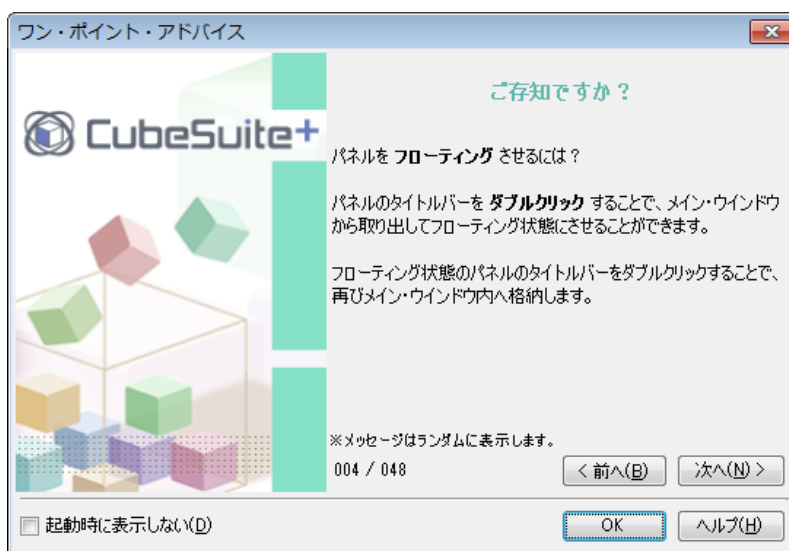
3.1 はじめに

CubeSuite+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。CubeSuite+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。

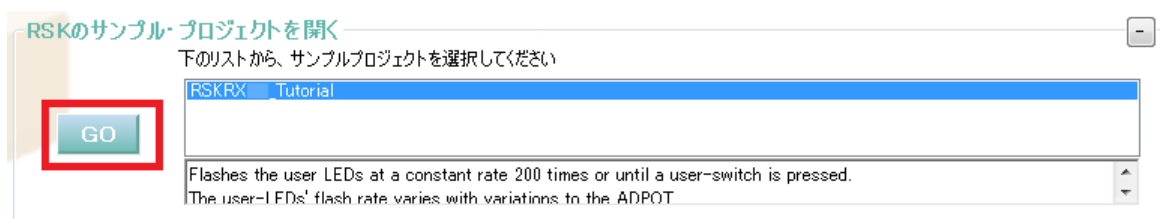
3.2 CubeSuite+の開始と E1 エミュレータの接続

まず、Windows のスタートメニューから CubeSuite+を起動してください。

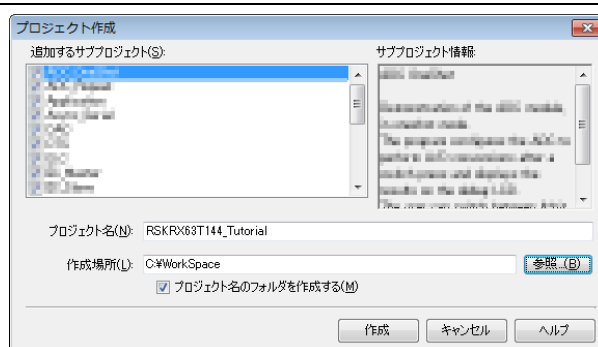
CubeSuite+を初めて使用する場合、ワンポイントアドバイスのダイアログが表示されます。



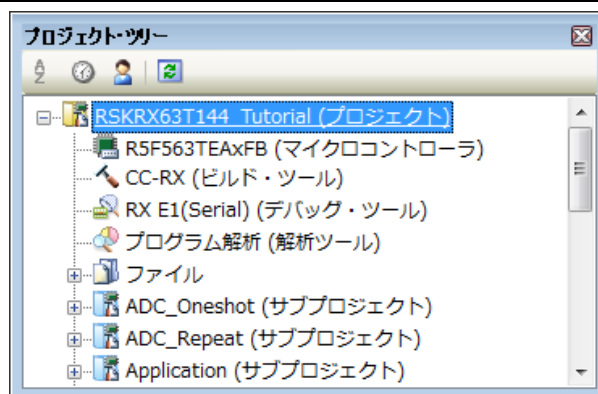
<OK>をクリックし、ダイアログを閉じてください。その後、スタートパネルが現れます。'RSK のサンプル・プロジェクトを開く'から RSKRX63T144_Tutorial を選択し、<GO>をクリックしてください。この操作によって、RSKRX63T144_Tutorial プロジェクトのコピーを保存します。



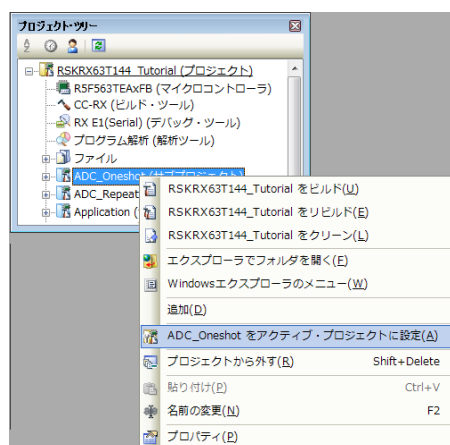
- CubeSuite+はプロジェクト作成ダイアログを表示しません。
- 各サブプロジェクト名のチェックボックスをチェックし、サブプロジェクトをすべて追加してください。各サブプロジェクトの情報はダイアログ上のサブプロジェクト情報の下に表示されます。
- プロジェクト名を入力し、作成場所を指定して<作成>をクリックしてください。
- コピーされたファイルを参照するには、プロジェクトツリーにリスト化されたファイルをダブルクリックしてください。新しいウィンドウが開きます。



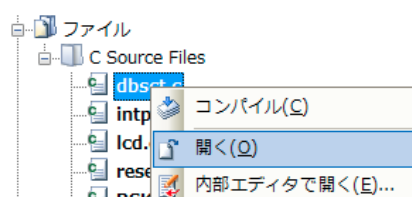
- RSKR63T144_Tutorial はマスタープロジェクトで、サブプロジェクトとして各サンプルを含んでいます。
- スクリーンショットのファイルフォルダはマスタープロジェクト RSKR63T144_Tutorial に属します。
- このフォルダは個別のフォルダ構造で用意されたテキストファイルを含むプロジェクトソースおよびヘッダファイルを含んでおりリストします。
- ファイルフォルダの下にサブプロジェクトがリストされます。
- 各サブプロジェクトフォルダを展開すると、マスタープロジェクトと同様にツールとフォルダ構成になっています。
- 現在アクティブなプロジェクトはプロジェクト名に下線が含まれます。
- 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 ADC_OneShot 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。



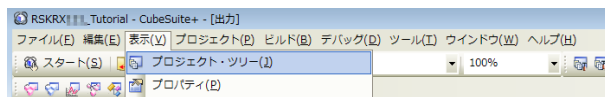
- ファイルフォルダは3つのサブフォルダを含んでいます。この構造はUSB関連プロジェクトを除き共通です。
- ファイルを参照するには、参照したいファイルを右クリックし、"開く"を選択します。ファイルをダブルクリックしても参照できます。



3.3 デバッグ・ツールの設定

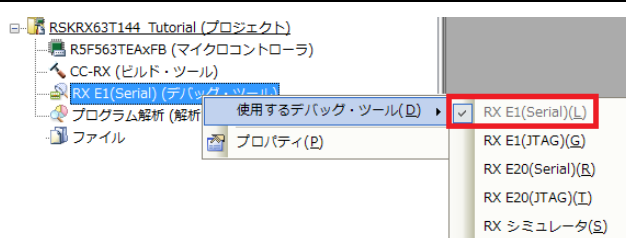
注：Tutorial プロジェクトは予めデバッグ・ツールの設定がされています。このセクションは新しいプロジェクトを作成するための説明です。

- プロジェクトツリーは CubeSuite+ の左側ウィンドウに表示されます。
- これはメニューバーから起動することができます。
(表示 -> プロジェクト・ツリー)

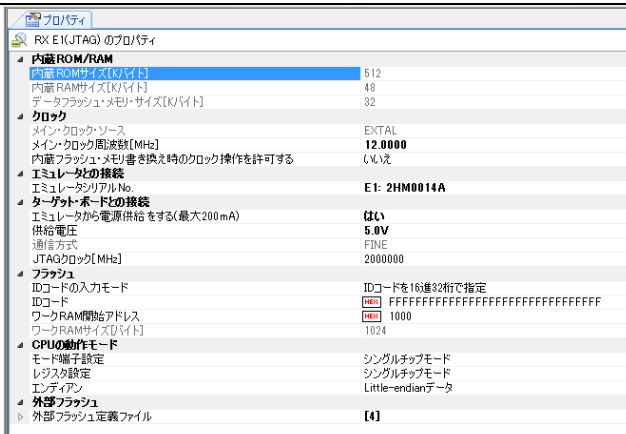


このリストはソースコードをリストするのと同様にデバイスをプログラムしデバッグするための IDE を形成するのに使用される多くのツールを含んでいます。既に設定された内容を確認するために、次の指示に従ってください：

- RX xxx (デバッグ・ツール) を右クリックし、RX E1 (Serial) を選択してください。スクリーンショットは予め RX E1 (Serial) が選択されています。



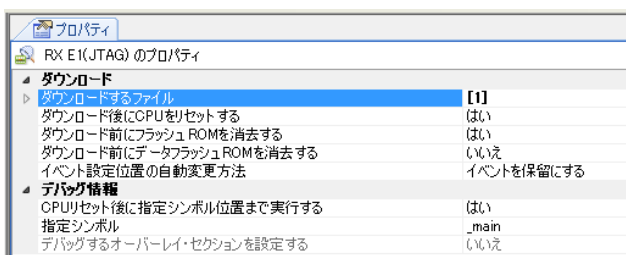
- RX E1 (Serial) (デバッグ・ツール) を右クリックし、プロパティを選択してください。
- 接続用設定タブをクリックしてください。
- 設定内容がスクリーンショットと同じであることを確認してください。
特に、メイン・クロックソースが"EXTAL"、メイン・クロック周波数 [MHz] が "12.000"、供給電圧が "5.0V" であることを確認してください。



注：外部電源を使用する場合は、エミュレータから電源供給をする(最大 200mA)を"いいえ"に設定してください。

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。

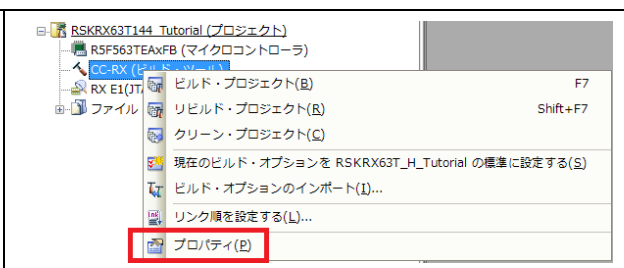
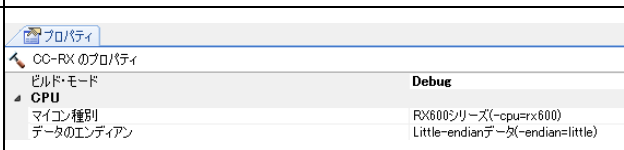
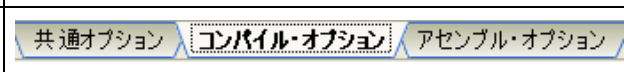
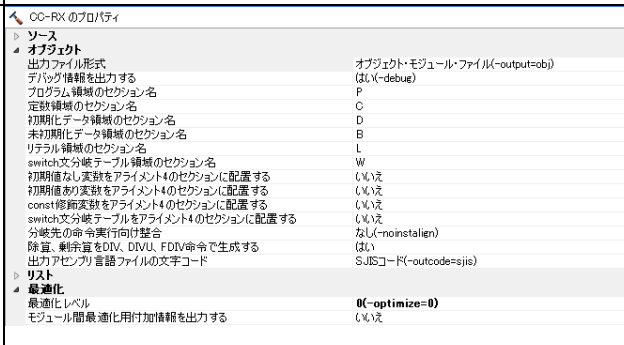


注：割り込みハンドラをエントリポイントとして指定しないでください。

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 はデバグ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グ設定です。最適化が有効の場合、デバグ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、デバグをスムーズに処理する為には、デバ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。

<ul style="list-style-type: none"> 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。 	
<ul style="list-style-type: none"> 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。 	
<ul style="list-style-type: none"> コンパイル・オプションタブを選択してください。 	
<ul style="list-style-type: none"> デバ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。

<ul style="list-style-type: none"> 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。

<ul style="list-style-type: none"> 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。 	
<ul style="list-style-type: none"> 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。 	
<ul style="list-style-type: none"> 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。 	
<ul style="list-style-type: none"> 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。 	
<ul style="list-style-type: none"> 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

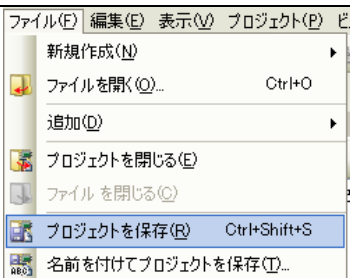
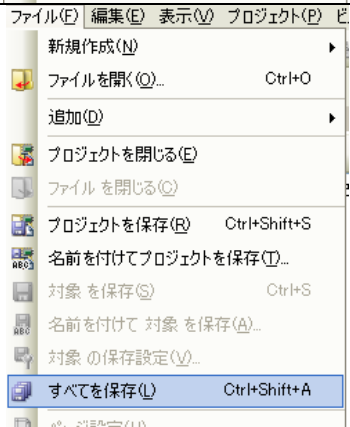

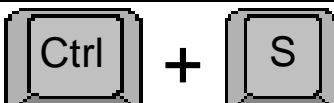
本チュートリアルでは、外部から CPU ボードに電源を供給する必要はありません。電源は USB ポートから供給されます。USB ポートに多くのデバイスが接続している場合、Windows がシャットダウンするかもしれないので注意してください。この問題が発生した場合、一部のデバイスを削除して、もう一度やり直してください。外部電源を供給する際、極性および電源電圧が適切であることを必ず確認してください。

LVD（電圧検出）回路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 RSKRX63T144 ユーザー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下さい。
- E1 をご使用のコンピュータの USB ポートに接続して下さい。
- 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下さい。
- 外部電源を CPU ボードに供給する場合、セクション 3.3 を参照し、エミュレータから電源供給するオプションを解除して下さい。


4.3 プロジェクトの保存

<p>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。</p> <ul style="list-style-type: none"> • 'ファイル' 'プロジェクトを保存' を選択します。 	
<p>CubeSuite+中のファイルを変更した場合、次の操作で保存することができます。</p> <ul style="list-style-type: none"> • 'ファイル' 'すべてを保存' を選択します。 	
<p>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することもできます。</p>	
<p>また、キーボードでファイルを保存することもできます。</p>	

5. チュートリアル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

CubeSuite+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。

<ul style="list-style-type: none"> 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。 	
<ul style="list-style-type: none"> 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。 	<pre> /***** * Function Name: main * Description : This function implements main function. * Arguments : None * Return Value : None *****/ void main(void) { R_MAIN_UserInit(); /* Start user code. Do not edit comment generated here */ /* Display Splash Screen */ Display_LCD(0, (uint8_t *)"Renesas"); Display_LCD(1, (uint8_t *)NICKNAME); /* Begins the initial LED flash sequence */ Flash_LED(); /* Start the timer_adc function */ timer_adc(); /* static_test function */ static_test(); while (1U) { /* Wait */ } /* End user code. Do not edit comment generated here */ } /***** </pre>

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、プログラムを実行することができます。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、CubeSuite+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認します。



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'_HardwareSetup'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。

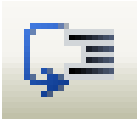





<ul style="list-style-type: none"> コード表示をメニューバーの'逆アセンブル'ボタン、'混合'ボタンで表示を切り替えることができます。 	<pre> 269: while (counter--) ffff87de 471e CMP R1,R14 ffff87e0 2103 BNE.B _DisplayDelay+BH 273: } ffff87e2 02 RTS 271: nop(); ffff87e3 03 NOP ffff87e4 621e ADD #1H,R14 ffff87e6 2ef8 BRA.B _DisplayDelay+6H 127: Init_LCD(); main: ➔ffff87e8 05cefeff BSR.A _Init_LCD 130: Display_LCD(LCD_LINE1, "Renesas"); ffff87ec fb22f48efff MOV.L #-0000710CH,R2 ffff87f2 6601 MOV.L #0H,R1 ffff87f4 0542ffff BSR.A _Display_LCD 131: Display_LCD(LCD_LINE2, NICKNAME); ffff87f8 fb22fc8efff MOV.L #-00007104H,R2 ffff87fe 754110 MOV.L #10H,R1 ffff8801 0535ffff BSR.A _Display_LCD 134: Flash_LED(); ffff8805 058bfdff BSR.A _Flash_LED 137: Timer_ADC(); ffff8809 05250400 BSR.A _Timer_ADC 140: static_test(); </pre>
 ← 逆アセンブルボタン  ← 混合ボタン	
<ul style="list-style-type: none"> ツールバーに逆アセンブルボタンを表示させるには、ツールバー上で右クリックして、パネル表示を選択してください。 	

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。

<ul style="list-style-type: none"> Flash_LED 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。Display_LCD 関数はストリングデータ'Renesas'および'RX63T144'をLCDの1行目および2行目にライトしませず（表示します）。 	<pre> ***** * Function Name : main * Description : The main program function. Displays the Renesas splash screen * onto the LCD display, then calls the 'flashLED' and 'TimerADC' * functions. The function then calls the statics test routine, * before waiting in an infinite while loop. * Argument : none * Return value : none ***** void main (void) { /* Initialise the debug LCD */ Init_LCD(); /* Displays the Renesas splash screen */ Display_LCD(LCD_LINE1, "Renesas"); Display_LCD(LCD_LINE2, NICKNAME); /* Begins the initial LED flash sequence */ Flash_LED(); /* Begins the ADC-varying flash Sequence */ Timer_ADC(); </pre>
<ul style="list-style-type: none"> Timer_ADC 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）。 	<pre> /* Begins the initial LED flash sequence */ Flash_LED(); /* Begins the ADC-varying flash Sequence */ Timer_ADC(); /* Begins the static variable test */ static_test(); </pre>
<ul style="list-style-type: none"> '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 Flash_LED 関数にエントリします。 	 
<ul style="list-style-type: none"> プログラムカウンタは Flash_LED 関数に移動します。この関数はユーザスイッチに接続されている外部割り込みを許可し、ユーザ LED を点滅させます。LED は 200 回点滅するかユーザスイッチが押されるまで点滅を繰り返します。 '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。 	<pre> void Flash_LED (void) { /* Variable used to count down the number of LED flashes */ static uint16_t flash_count = 0xC8; /* Declare a delay count variable */ uint32_t ulLed_Delay = 0; /* Flash the LEDs for 200 times or until a user switch is pressed */ while ((0 == g_switch_flag) && (--flash_count > 0)) { for (ulLed_Delay = 0; ulLed_Delay < 4000000; ++ulLed_Delay) { /* delay */ } /* Toggles the LEDs after a specific delay. */ Toggle_LED(); } /* Reset the g_switch_flag flag variable */ g_switch_flag = 0; } </pre>

<ul style="list-style-type: none"> プログラムカウンタは Timer_ADC 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。   <p>Timer_ADC 関数は連続的な A/D 変換および周期タイマ（周期は A/D 変換結果によって更新されます）を動作させます。</p> <p>周期タイマによってユーザ LED の点滅間隔が制御されます。</p>	<pre> /* Begins the initial LED flash sequence */ Flash_LED(); /* Begins the ADC-varying flash Sequence */ Timer_ADC(); /* Begins the static variable test */ static_test(); </pre>
<ul style="list-style-type: none"> 'timeradc.c'ファイルを開いてください。Excep_CMT1_CMI1 割り込みハンドラ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。 '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。 	
<ul style="list-style-type: none"> プログラムはタイマの周期経過によってハードウェア・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。 	<pre> #pragma interrupt (Excep_CMT1_CMI1(vect=VECT_CMT1_CMI1)) void Excep_CMT1_CMI1 (void) { /* Toggle LEDs */ Toggle_LED(); /* Update the timer */ CMT1.CMCOR = (uint16_t)((g_adc_result / 50) * 200) + 300; } </pre>

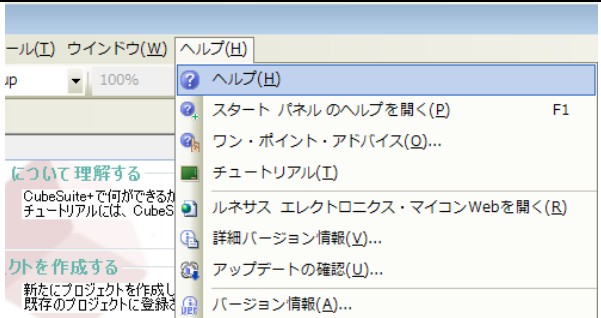
<ul style="list-style-type: none"> • 'F5'ボタンを押し、プログラム実行を再開してください。static_test 関数が実行されることで、LCD の 2 行目のstring表示が「STATIC」から「TESTTEST」に置き換わることを確認してください。 • string表示が置き換わった後、2 行目の表示は初期表示「RX63T144」に戻ります。 	<pre>static void static_test (void) { /* Declare loop count variable */ uint8_t uicount = 0; /* Write cStr variable, "STATIC" to LCD */ Display_LCD(LCD_LINE2, cStr); /* Begin for loop which writes one letter of ucReplace to the LCD at a time The nested while loops generate the delay between each letter change */ for (uicount = 0; uicount < 8; uicount++) { /* Start a one-shot timer to create a delay between each loop iteration */ Timer_Delay(40, MILLISECS); /* Replace letter number 'uiCount' of 'ucStr' from 'ucReplace' */ cStr[uicount] = cReplace[uicount]; Display_LCD(LCD_LINE2, cStr); } /* Clear LCD Display */ cStr[uicount] = '\0'; /* Add a delay before overwriting the string */ Timer_Delay(40, MILLISECS); /* Write MCU nickname to LCD again */ Display_LCD(LCD_LINE2, NICKNAME); }</pre>
<ul style="list-style-type: none"> • '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、RSKRX63T144 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。

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機能を持っています。E1 エミュレータの詳細情報は、E1/E20 エミュレータ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

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

RX63T(144-pin)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/rskrx63t144> (日本サイト)
<http://www.renesas.com/rskrx63t144> (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
 グローバル：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

RSKRX63T144 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	2013.11.11	—	初版発行
1.01	2014.03.19	—	「2. 略語および略称の説明」を更新
		8	ビルドコンフィグレーション”Debug”、”Release”に最適化レベル情報を追加
		10-20	一部の説明文、図に枠線を追加
		10	ページ最後の説明文を更新 “_____をアクティブ・プロジェクトに設定” ↓ ”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”
		16	逆アセンブルボタン、混合ボタンを区別化

RSKRX63T144 チュートリアルマニュアル

発行年月日 2014年03月19日 Rev.1.01

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