

RL78/G13

Renesas Starter Kit ソフトウェアヘルプマニュアル (CubeSuite+)

16 ビット・シングルチップ・マイクロコントローラ
RL78 ファミリ

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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。
5. 輸出に際しては、「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。本資料に記載されている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の目的で使用しないでください。また、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に使用することができません。
6. 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、誤りが無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。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されてい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生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
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
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、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社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。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、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、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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、当社は、一切その責任を負いません。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社と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 ハードウェア概要と電気的特性をユーザに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、RSK 製品の機能概観を含みますが、組み込みプログラミングまたはハードウェア設計ガイドのためのマニュアルではありません。また、RSK および開発環境のセットアップに関するその他の詳細は、チュートリアルに記載しています。

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。

RSKRL78G13 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	RSKRL78G13 ユーザーズマニュアル	R20UT0459JG
ソフトウェアヘルプマニュアル	サンプルコードの機能説明	RSKRL78G13 ソフトウェアヘルプマニュアル	R20UT0746JG (本マニュアル)
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	RSKRL78G13 チュートリアルマニュアル	R20UT0460JG
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	RSKRL78G13 クイックスタートガイド	R20UT0461JG
回路図	CPU ボードの回路図	RSKRL78G13 CPU ボード回路図	R20UT0458EG
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	RL78/G13 ユーザーズマニュアル ハードウェア編	R01UH0146JJ

2. 略語および略称の説明

略語／略称	英語名	備考
ADC	Analogue-to-Digital Converter	A/D コンバータ
CPU	Central Processing Unit	中央処理装置
DMA	Direct Memory Access	CPU の命令を介さずに直接データ転送を行う方式
DMAC	Direct Memory Access Controller	DMA を行うコントローラ
I ² C	Phillips™ Inter-Integrated Circuit Connection Bus	-
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
LIN	Local Interconnect Network	-
LVD	Low Voltage Detection	低電圧検出
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
RSK	Renesas Starter Kit+	ルネサススタータキット
RTC	Real Time Clock	リアルタイムクロック
WDT	Watch Dog Timer	ウォッチドッグタイマ

目次

1. 概要.....	7
1.1 目的.....	7
2. イントロダクション.....	8
3. サンプルコードコンセプト.....	9
3.1 サンプルコードの構成.....	9
3.2 サンプルコードリスト.....	10
4. Tutorialサンプル.....	11
4.1 Tutorial.....	11
4.1.1 説明.....	11
4.1.2 オペレーション.....	12
4.2 Application.....	13
4.2.1 説明.....	13
5. 周辺機能サンプル.....	14
5.1 ADC_OneShot.....	14
5.1.1 説明.....	14
5.1.2 オペレーション.....	14
5.2 ADC_Repeat.....	15
5.2.1 説明.....	15
5.2.2 オペレーション.....	15
5.3 LIN.....	16
5.3.1 説明.....	16
5.3.2 オペレーション.....	16
5.4 Async_Serial.....	17
5.4.1 説明.....	17
5.4.2 オペレーション.....	17
5.5 Sync_Serial.....	17
5.5.1 説明.....	17
5.5.2 オペレーション.....	17
5.6 LVD.....	18
5.6.1 説明.....	18
5.6.2 オペレーション.....	18
5.7 IIC_Master.....	19
5.7.1 説明.....	19
5.7.2 オペレーション.....	19
5.8 IIC_Slave.....	20
5.8.1 説明.....	20
5.8.2 IICスレーブコマンド.....	20
5.8.3 オペレーション.....	20
5.9 Timer.....	21
5.9.1 説明.....	21
5.9.2 オペレーション.....	21
5.10 DMAC.....	22
5.10.1 説明.....	22
5.10.2 オペレーション.....	22
5.11 WDT.....	23
5.11.1 説明.....	23
5.11.2 オペレーション.....	23
5.12 RTC.....	23

5.12.1	説明	23
5.12.2	オペレーション	23
5.13	Snooze	24
5.13.1	説明	24
5.13.2	オペレーション	24
6	追加情報	25

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、CubeSuite+のコード生成を備えたサンプルコードの機能とその相互関係について説明します。コード生成（設計ツール）はコード生成のために使用される CubeSuite+内のツールでユーザの設定に応じて MCU の周辺機能を形成します。また、コード生成は割り込みハンドラも周辺機能のスタート/ストップ関数も生成できます。

本マニュアルはコード生成そのもののマニュアルではなく、サンプルコードでコード生成がどのように使用されているかを説明するものです。

2. イントロダクション

本マニュアルは各サンプルコードのオペレーションを観察するのに必要なサンプルコードおよび工程の機能性について説明します。

3. サンプルコードコンセプト

3.1 サンプルコードの構成

図 3-1 は RSK サンプルコードの基本的な構成を示しています。最初の関数'hdwinit'と'R_Systeminit'はメインプログラムコードが実行される前にマイクロコントローラの設定を行う関数です。

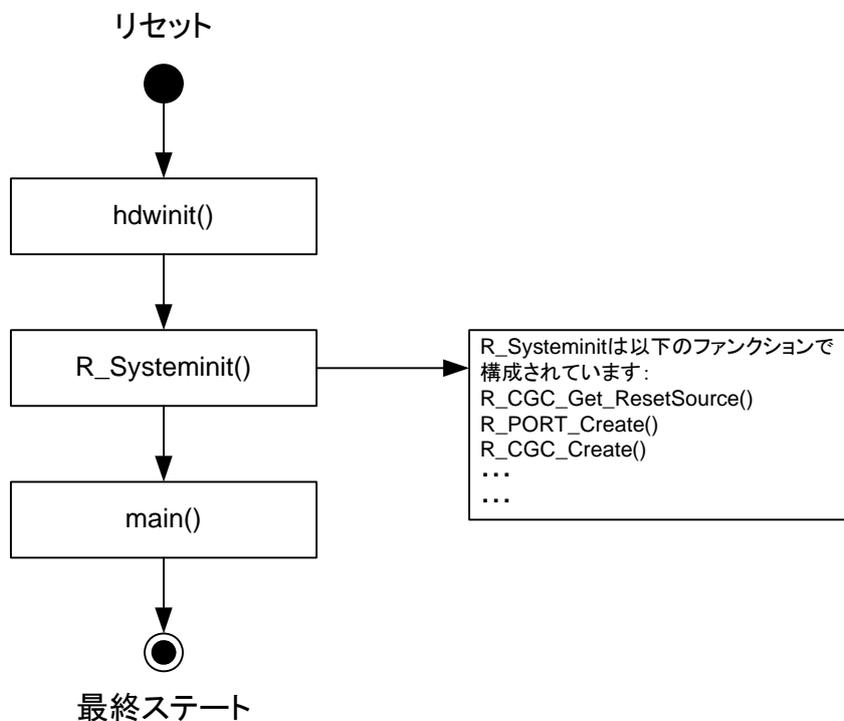


図 3-1: サンプルコードの基本構成

すべてのサンプルコードプロジェクトは CPU リセット後にメイン関数位置まで実行するようデバッグ設定されています。プログラムコードはマイクロコントローラにコードをプログラムした後、自動的に実行され、メイン関数内の第 1 命令で止められます。したがって、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。CubeSuite+はプログラムのエントリポイントをメイン関数以外に変更することができます。詳細はチュートリアルマニュアルを参照してください。

本マニュアルでのサンプルコードの説明は、メイン関数内のプログラム実行に着目します。

3.2 サンプルコードリスト

RSKRL78G13のサンプルコードリストを表 3-1 に示します。

サンプルコード	内容
Tutorial	デバッガおよび RSK ハードウェアの基礎的な使用法を学ぶためのデモコード
Application	ユーザ開発用のブランクプロジェクト
ADC_OneShot	10bit ADC モジュール (ワンショットモード) のデモコード
ADC_Repeat	10bit ADC モジュール (リピートモード) のデモコード
Async_Serial	シリアルアレイユニット (UART) のデモコード
DMAC	DMAC モジュールのデモコード
IIC	I ² C ユニット (マスターモード/スレーブモード) のデモコード
LIN	LIN モジュール (マスターモード/スレーブモード) のデモコード
LVD	電圧検出回路のデモコード
Sync_Serial	シリアルアレイユニット (CSI) のデモコード
Timer	タイマモジュール (4 種のモード) のデモコード
Watchdog	ウォッチドッグタイマのデモコード
Snooze	SNOOZE モードのデモコード

表 3-1: サンプルコードリスト

4. Tutorial サンプル

4.1 Tutorial

サンプルコード“Tutorial”はデバッガおよび RSK ハードウェア基礎的な使用方を学ぶためのサンプルコードです。

4.1.1 説明

Tutorial はポートピン制御、割り込み設定、C 変数初期化を行うために 3 つの関数をコールします。これらの関数を図 4-1 に示します。

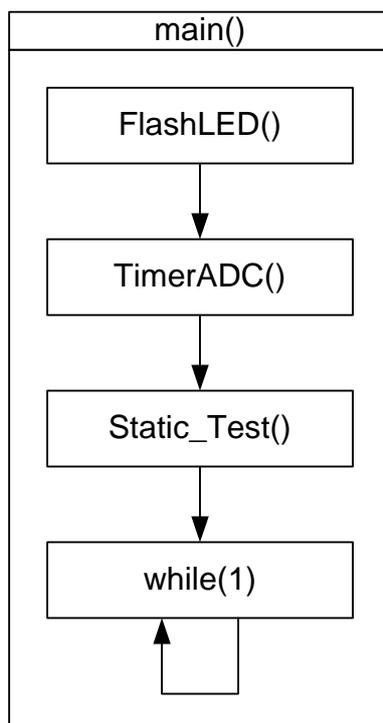


図 4-1: Tutorial フロー

4.1.2 オペレーション

- ① すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ② LCD モジュールを初期化し、LCD の 1 行目に'Renesas'、2 行目にマイクロコントローラ名を表示します。
- ③ Flash_LED 関数をコールします。この関数は繰り返し LED をトグル出力し、スイッチが押されるか LED が 200 回トグル出力されるまで LED のトグル出力が繰り返されます。
- ④ その後、周期的に A/D 変換を起動するために ADC チャンネルおよびタイマチャンネルを形成する TimerADC 関数をコールします。
- ⑤ タイマチャンネルの周期が経過すると LED がトグル出力されます。このとき A/D 変換結果をフェッチしてタイマ周期を更新するために使用されます。
- ⑥ TimerADC をコールし、タイマおよび ADC 割り込みのセットアップ後、Statics_Test 関数をコールします。
- ⑦ Statcis_Test 関数は LCD の 2 行目に'STATIC'を表示し、ストリング定数'TESTTEST'に表示内容を置き換えます。表示内容の置き換えが完了すると、表示内容は初期表示に戻ります。その後、コードは無限ループに入ります。周期的なタイマが動作しているので LED はトグル出力を繰り返します。ボード上のポテンショメータの調整により、A/D 変換結果を変えることでタイマ周期を変えることができます。

4.2 Application

4.2.1 説明

Application はユーザ自身でコード作成するために用意されたサンプルです。メイン関数が実行される前に実行されるハードウェア初期化および設定コードを含みますがメイン関数にはコードがありません。

5. 周辺機能サンプル

本セクション中のサンプルコードでは、初期化の例およびいくつかの周辺モジュールの使用法について説明します。また、周辺機能をデバッグする方法についても説明します。

5.1 ADC_OneShot

5.1.1 説明

本サンプルコードはワンショットモード（ワンショット変換）による A/D 変換のデモコードです。ボード上のスイッチを押すと、ポテンシオメータ RV1 の入力を A/D 変換します。

ポテンシオメータは簡易的にマイクロコントローラに可変アナログ入力供給をするために備え付けられています。A/D 変換の精度は保証できませんので、予めご了承ください。

5.1.2 オペレーション

- ① すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。ボード上のポテンシオメータに接続された A/D チャンネルは最初 10bit 分解能に設定されます。
- ② LCD モジュールを初期化し、LCD にインストラクションを表示します。
- ③ ボード上のユーザスイッチに接続された外部割り込みを許可します。
- ④ コードは無限ループに入り、スイッチが押されるのを待ちます。
- ⑤ スイッチ SW1 が押される場合、対応する外部割り込みハンドラがコールされます。A/D 変換を開始し、変換が完了するのを待ちます。変換完了後に、割り込みハンドラは分解能モードステータスをチェックします。結果は A/D 値をバッファアレイ中の LCD 表示用に整えられたストリングに変換する前に分解能に合わせて保存されます。その後、ストリングは割り込みハンドラから復帰する前に LCD に表示されます。また、プログラムはメイン関数に戻ります。
- ⑥ スイッチ SW2 を押すと、対応する外部割り込みハンドラがコールされます。A/D 変換の分解能を 8bit に設定し、分解能モードステータスを'2'に設定します。割り込みハンドラはメイン関数に戻ります。
- ⑦ スイッチ SW3 を押すと、対応する外部割り込みハンドラがコールされます。A/D 変換の分解能を 10bit に設定し、分解能モードステータスを'3'に設定します。

5.2 ADC_Repeat

5.2.1 説明

本サンプルコードはリピートモード（連続変換モード）による A/D 変換のデモコードです。コードはボード上のポテンショメータ RV1 の入力を繰り返し A/D 変換します。また、周期的なタイマ割り込みによって A/D 変換値を更新して LCD モジュールに表示します。

ポテンショメータは簡易的にマイクロコントローラに可変アナログ入力供給をするために備え付けられています。AD 変換の精度は保証できませんので、予めご了承ください。

5.2.2 オペレーション

- ① すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。ボード上のポテンショメータに接続された A/D チャンネルは最初 10bit 分解能に設定されます。
- ② LCD モジュールを初期化し、LCD にサンプル名を表示します。
- ③ ボード上のユーザスイッチ SW1、SW3 に接続された外部割り込みを許可します。
- ④ コードは ADC と周期的なタイマを起動します。
- ⑤ コードは無限ループに入り、割り込みを待ちます。
- ⑥ A/D 変換が完了すると A/D 変換割り込みハンドラがコールされます。どのユーザスイッチが最後に押されたかをチェックし、結果は 10bit/8bit 分解能に合った形式で保存されます。
- ⑦ スイッチ SW1 が押される場合、対応する外部割り込みハンドラがコールされます。A/D 変換の分解能を 8bit に設定し、SW1 が押されたことをフラグに知らせます。
- ⑧ スイッチ SW3 が押される場合、対応する外部割り込みハンドラがコールされます。A/D 変換の分解能を 10bit に設定し、SW3 が押されたことをフラグに知らせます。
- ⑨ タイマチャンネルの割り込みハンドラはタイマのオーバーフロー毎にコールされます。LCD に表示される前に A/D 変換結果は LCD 表示用に変換されます。

5.3 LIN

5.3.1 説明

本サンプルコードは LIN のデモコードです。サンプルコード中の以下`#define` 設定の有無によってマスターモードまたはスレーブモードを選択できます。

```
#define MASTER_LIN
```

注：スレーブモードはオプションリンク抵抗の変更が必要になります。抵抗の変更に関しては `Description.txt` ファイルを参照してください。

5.3.2 オペレーション

- ① ボード上の LIN コネクタを使用して、マスター/スレーブ装置を接続してください。
- ② 5V 電源を接続し、電源を投入してください。
- ③ すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ④ LCD モジュールを初期化し、選択された動作モードに依存して LCD に”LIN Mstr”または”LIN Slav”を表示します。
- ⑤ その後、外部割り込み `INTP0` を許可します。
- ⑥ ボード上の LIN トランシーバは許可状態になります。また、ボード上の LED1 が消灯します。
- ⑦ マスターモードにおいて、スレーブからのデータ受信が検出されるまで、サンプルは周期的に LIN ヘッダを送信します。検出は割り込みを使用して取り扱われます。対応する割り込みハンドラは、指定されたアドレス範囲へ受信データを格納します。スレーブデータ受信の検出において、`LIN_Master_Receive` 関数もコールされます（タイマ動作を止めて連続動作を禁止にします）。チェックサムは受信チェックサムと比較される前に受信データを使用して計算されます。チェックサムが一致する場合、受信データはストリングに変換され、LCD に表示されます。プログラム実行が止められるまで、コードは無限ループ内で待機します。
- ⑧ スレーブモードにおいて、サンプルコードはマスターによって送信された LIN ヘッダを周期的に待ちます。受信スレーブアドレスが `0x10` と一致する場合、`LIN_Slave_Transmit` 関数がコールされます。関数は A/D 変換を開始し、変換が完了するのを待ちます。結果は LCD 表示用のストリングに変換され、データを送信します。関数はメイン関数に戻ります。次のヘッダが受け取られる前に、受信スレーブアドレスの中身はクリアされます。プログラム実行が止められるまで、コードは無限ループ内で待機します。

5.4 Async_Serial

5.4.1 説明

本サンプルコードは非同期モードによるシリアルレイユニット (SAU) のデモコードです。RS232 ケーブルを介して PC 上のターミナルと通信します。

5.4.2 オペレーション

サンプルコードを実行する前に、RS232 ケーブルを経由して PC とボードを接続し、ターミナルソフトを起動します (サンプルコードのヘッダ欄でインストラクションを確認できます)。

- ① サンプルコードを実行する前に、RS232 ケーブルを経由して PC とボードを接続し、ターミナルソフトを起動します。サンプルコードのヘッダ欄でインストラクション、ターミナル設定を確認できます。
- ② すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ③ LCD モジュールを初期化し、LCD にサンプル名を表示します。
- ④ コードは PC との通信用に SAU0 ユニットのチャンネル 0 を設定し、インターバルタイマを起動します。
- ⑤ コードは割り込みを待ちます。
- ⑥ インターバルタイマは対応する割り込みハンドラによって 100ms 間隔の周期的な割り込みを生成します。割り込みハンドラはターミナル上で入力された文字をチェックし、文字が「z」でない場合、ターミナルヘインクリメントする ASCII 番号 (0~9、9 の後は 0 へ戻る) を送信します。送信が進行中であることを示すために LED0 を点灯させます。文字が「z」の場合、送信は中止され、LED0 が消灯します。

5.5 Sync_Serial

5.5.1 説明

本サンプルコードは同期モードによるシリアルレイユニット (SAU) のデモコードです。2 つの SAU チャンネル間をループバックさせ通信します。

以下の接続を行ってください：

TxD0 [JA2.Pin6]	↔	RxD1 [JA6.Pin7]
RxD0 [JA2.Pin8]	↔	TxD1 [JA6.Pin8]
SCK00 [JA2.Pin10]	↔	SCK10 [JA6.Pin10]

5.5.2 オペレーション

- ① すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ② LCD モジュールを初期化し、LCD にサンプル名を表示します。
- ③ コードは Sync_Transfer 関数をコールし、SAU データバッファをクリアします。SAU チャンネル CSI00 および CSI10 は 2 チャンネル間のデータ転送で利用可能になります。チャンネル CSI00 はマスターとして形成され、スレーブとして形成されるチャンネル CSI10 にクロック信号を出力します。チャンネルのデータ送信および受信の両方は割り込みが使用されます。ユーザが結果をチェックすることができる前に転送が完了することを可能にするためにタイマチャンネルを使用して、遅延が作られます。データ中にエラーが生じる場合、「Failure」メッセージが LCD の 2 行目に表示されます。エラーが生じない場合、2 チャンネル間のデータ転送は転送の終わりに比較され、LCD の 2 行目に「Success」を表示します。データが一致しなかった場合、代わりに「Failure」が表示されます。
- ④ コードは無限ループに入ります。

5.6 LVD

5.6.1 説明

本サンプルコードは LVD（電圧検出）回路のデモコードです。電源が検出レベル以下に落ちる時に割り込みを生成します。検出レベルは約 4.06V です。

5.6.2 オペレーション

- ① すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ② LCD モジュールを初期化し、LCD にサンプル名を表示します。
- ③ コードは無限ループに入る前に LVD オペレーションを許可するほかに周期タイマを起動します。
- ④ コードは割り込みを待ちます。
- ⑤ 電源が 4.06V 以上である限り、周期タイマはユーザ LED すべてを点滅させる割り込みを生成します。タイマの割り込みハンドラは、ユーザ LED の同期を示すユーザ定義のフラグステータスをチェックします。フラグがセットされている場合、LED0～LED2 は LED3 に同期してトグル出力されます。すべての LED がトグル出力される前に LCD の 2 行目がクリアされます。
- ⑥ 電圧検出が起きると、LVD 割り込みハンドラがコールされます。タイマを停止し、LVD フラグがセットされている場合に LCD の 2 行目に「Detected」を表示し、LED0～LED2 が消灯します。LED3 は点灯のままです（LED の同期を示すフラグはクリアされます）。タイマは再度開始されます。

5.7 IIC_Master

5.7.1 説明

本サンプルコードはマスターモードにおいて IIC ユニットより EEPROM メモリまたはスレーブモードで動作している他の RSKRL78G13 CPU ボードへのリード/ライトオペレーションをデモします。サンプルコードは 8-bit word アドレスの EEPROM デバイスで動作するように設定されています。合計 8 データ・バイトがスレーブメモリ（メモリ位置 0 から開始）へ転送されます。EEPROM メモリにライトする場合、第 2 データは各書き込み（0~9）後にインクリメントされます。サンプルコードは EEPROM メモリとの通信のために形成されます。スレーブとして使用する場合は、「r_cg_typedefine.h」ファイル中の以下のラインを有効にしてください：

```
//#define SLAVE_RL78G13
```

RSKRL78G13 CPU ボードとスレーブ装置の接続は以下の通りです。（注：CPU ボード上のデータ、クロックラインに 4.7KΩ のプルアップ抵抗が接続されています）

SDAA0 [JA1.Pin25]	↔	相手側 SDA
SCLA0 [JA1.Pin26]	↔	相手側 SCL
CON_5V [JA1.Pin1]	↔	相手側 VCC
0V [JA1. Pin2]	↔	相手側側 VSS

5.7.2 オペレーション

- ① 5V 電源を接続し、電源を投入してください。
- ② すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ③ LCD モジュールを初期化し、LCD にサンプル名を表示します。
- ④ コードは Master_EEPROM 関数をコールする前にユーザスイッチ用の割り込みを許可します。
- ⑤ Master_EEPROM 関数内で、第 2 バッファ配列位置は ASCII 文字「0」を表す 0x30 にセットされます。
- ⑥ コードは無限ループに入ります。データ転送は送信または受信割り込み（スイッチ SW1 および SW3 で切り替え）を使用して行われます。ライトオペレーションはスレーブ装置に転送されたデータを確認するためにリードオペレーションの前に行ってください。リードオペレーションがライトオペレーションの前に行われる場合、データはスレーブ装置から読まれますが、LCD の 2 行目に表示されません。代わりに、1 行目に「IIC Push」、2 行目に「SW1->SW3」が表示されます。
- ⑦ スイッチ SW2 を押すと、スレーブ装置へ送られた最後のデータを表示することができます。
- ⑧ ライトおよびリードオペレーションが行われた後、送信データと受信データは比較されます。データが一致しない場合、LCD に「W/R Diff」が表示されます。

5.8 IIC_Slave

5.8.1 説明

本サンプルコードはスレーブモードにおける IIC ユニットより単純な 8 バイトの疑似 EEPROM メモリへのオペレーションをデモします。

RSKRL78G13 CPU ボードとマスター装置の接続は以下の通りです。（注：CPU ボード上のデータ、クロックラインに 4.7KΩ のプルアップ抵抗が接続されています）

SDAA0 [JA1.Pin25]	↔	相手側 SDA
SCLA0 [JA1.Pin26]	↔	相手側 SCL
CON_5V [JA1.Pin1]	↔	相手側 VCC
0V [JA1. Pin2]	↔	相手側側 VSS

5.8.2 IICスレーブコマンド

(1) ライトオペレーション

- 疑似 EEPROM にライトするために、マスターはスタートコンディションの後に EEPROM デバイスアドレス（デフォルトアドレス：0x10）を送り、スレーブからの ACK 信号を待ちます。
- マスターは 8 ビットの EEPROM メモリアドレス（未使用）を含むデータを 8 バイト送ることにより進行し、次にスレーブからの ACK 応答を待ちます。スレーブは各 8 バイトの転送の同じアドレスから開始によって受信バイトを省くので、メモリアドレスバイトが無視されます。
- マスターは各データバイトの後の ACK 応答を待ちます。スレーブは形成され 8 バイトのデータ受信を待ちます。
- 最終バイトが送られたら、マスターは処理を終了するストップ信号を送ります。
- 疑似 EEPROM の内部アドレスポインタはバイトライトごとに自動的にインクリメントされます。



(2) リードオペレーション

- アドレスが最大値に達してストップコンディションが検出されるまで、リードオペレーションは常に現在の内部疑似 EEPROM メモリポインタから開始し、次のバイトへ自動的にインクリメントされます。
- その後、EEPROM スレーブは ACK 信号で返答し、現在のメモリ位置にあるデータを送り、次のバイト位置へ内部ポインタを自動的にインクリメントします。
- 別のバイトを読むために、マスターは ACK 信号を送ります。8 バイトリードされるまで、マスターはこの操作が繰り返され、ストップコンディションを含むオペレーションを終了します。



5.8.3 オペレーション

- ① すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ② LCD モジュールを初期化し、LCD にサンプル名を表示します。
- ③ コードは装置に送られ、かつマスター装置から受け取られるデータ先頭番地を指定するために Slave_EEPROM 関数をコールします。コードはメイン関数に戻ります。
- ④ マスター装置から受け取られたデータは、8 バイトのバッファ配列 'Master_Data' に格納されます。スレーブリードがマスターによって要求される場合、同じデータもマスターへ送られます。

5.9 Timer

5.9.1 説明

本サンプルコードは 4 つのアプリケーションのうちの 1 つを実行するためにタイマレイユニット (TAU) を形成します。合計 4 本の TAU チャンネルがサンプルコード中で使用されます。使用される 4 つの TAU チャンネルはチャンネル 0、1、3、5 です。4 つのアプリケーションは方形波出力、パルス間隔測定、イベントカウントおよび PWM 出力です。初期プロジェクトは PWM アプリケーションを実行するために形成されます。アプリケーションを変更する場合、「r_cg_typedefine.h」ファイル中の定義命令に希望のアプリケーション定義番号に対応する値を入力してください：

```
#define __Current_TMR_APP    1
```

5.9.2 オペレーション

- ① すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ② LCD モジュールを初期化します。
- ③ 選択されたアプリケーションによって、以下のいずれかのオペレーションになります。
- ④ PWM が選択されている場合、アプリケーションヘッダ JA5.Pin9 にオシロスコープを接続してください。
- ⑤ LCD に「Tr.PWM」を表示し、PWM 波形を出力する TAU チャンネル 0 を開始する前にスイッチ SW1 および SW2 の外部割り込みを有効にします。その後、無限ループに入り、割り込みを待ちます。TAU チャンネル 0 は 1ms の周期中でデューティサイクル 0%~90%間で変化するように設定されます。波形はアプリケーションヘッダ JA5.Pin9 (TO01 ピン) 上で観測できます。スイッチ SW1 を押すと、デューティサイクルを固定し、LCD の 2 行目にデューティサイクルを表示します。スイッチ SW2 を押すと、LCD の 2 行目をクリアし、デューティサイクルの更新 (変更) を再開します。
- ⑥ 方形波出力が選択されている場合、アプリケーションヘッダ JA2.Pin20 にオシロスコープを接続してください。
- ⑦ LCD に「Tr.SqOut」、「JA2-20」を表示し、1KHz の方形波がアプリケーションヘッダ JA2.Pin20 (TO05) 上に出力されます。
- ⑧ パルス間隔測定が選択されている場合、LCD に「Tr.Intvl」、「Push SW1」を表示します。コードが無限ループに入る前に、スイッチ SW1 に接続している外部割り込みピンが有効になります。TAU チャンネル 5 は周期的に 1ms 間隔で割り込みを生成するために形成されます。SW1 を押すと、対応する割り込みハンドラによって割り込みを生成します。割り込みハンドラはチャンネル 5 のカウント開始/停止前にエッジ検出を確認します。検出が立ち下がりの場合、割り込みカウント変数 gTimerTick はクリアされ、エッジ検出が立ち上がりエッジに変更されます。その後、チャンネル 5 のカウントは割り込みハンドラがメイン関数へ戻る前に開始されます。カウントは SW1 が押されている間、継続します。TAU チャンネル 5 の割り込みハンドラはこの間コールされ、gTimer_Tick をインクリメントします。SW1 が開放されると直ちに対応する割り込みハンドラはコールされ、TAU チャンネル 5 のカウントを停止し、カウント値が 10000 未満かどうかチェックします。10000 未満の場合、HEX カウント値は BCD ストリングに変換され、LCD に ms 表示されます。10000 を越える場合、計測期間が 10 秒以上であることを示すために LCD に「> 10s」を表示します。信号検出は割り込みハンドラを出て、メイン関数に戻り再び SW1 が押される前に、立ち下りエッジ検出に設定を変更します。

- ⑨ イベントカウントが選択されている場合、LCDに「Tr.Event」、「Count:」を表示します。TAUチャンネル3のカウントが開始される前に、カウント変数 `gEventCount` がクリアされます。コードは無限ループに入ります。TAUチャンネル3は入力信号ピン `TI03` を外部クロック信号として計測されるよう形成されます。このピンはスイッチ `SW3` (`INTP4`) とマルチプレクスされています。外部割り込みは無効になり、`SW3` の立ち下りエッジ信号をカウントします。割り込みリクエストは各立ち下がりエッジ信号によって生成され、対応する割り込みハンドラによって処理されます。HEX カウント値はストリング変換されて、LCDに表示されます。

5.10 DMAC

5.10.1 説明

本サンプルコードはDMACのデモコードです。スイッチ `SW1` が押されると、配列へDMA転送を行います。

- メニューバーからメモリウィンドウを開いてください。
- アドレスレンジ `0xFFA00~0xFFBFF` にスクロールしてください。
- 任意のデータが指定アドレスに格納されていることを確認してください。

5.10.2 オペレーション

- ① すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ② LCDモジュールを初期化し、LCDにサンプル名を表示します。そして、すべてのユーザLEDが点灯します。
- ③ スイッチ `SW1` の割り込みを有効にし、`SW1` が押されるのを待ちます。
- ④ シリアルアレイユニット `SAU0` のチャンネル0とDMACを形成する `DMAC_DataTransfer` 関数をコールします。`SAU0` のチャンネル0は、ループ内部のDMACによってメモリに転送された各送信データを備えた送信のために形成されます。ループから出る前に、トータル512バイトの転送が行われます。
- ⑤ 転送が完了すると、すべてのユーザLEDが消灯し、LCDの2行目に「END」が表示されます。
- ⑥ コードは無限ループに入ります。
- ⑦ ここでプログラム実行を停止させて、メモリ内容を確認してください。
- ⑧ アドレスレンジ `0xFFA00~0xFFBFF` にストリング「Renesas RL78G13」が格納されています。

5.11 WDT

5.11.1 説明

本サンプルコードは WDT（ウォッチドッグタイマ）のデモコードです。WDT はタイマのカウンタ周期が 75%に達する場合に、割り込みを生成します。TAU チャネルはボード上のポテンショメータの調整位置（A/D 変換値）に応じて周期が変わる周期的な割り込みを生成し、WDT のカウンタをリスタートします。プログラムを実行する前に、ポテンショメータを反時計回り一杯に回してください。

5.11.2 オペレーション

- ① すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ② LCD モジュールを初期化し、LCD に「WDT」、「Running」を表示します。
- ③ コードは無限ループに入る前に WDT、ADC および TAU チャネルを起動します。
- ④ ADC はポテンショメータによって調整された電圧を繰り返し変換します。結果は ADC の割り込みハンドラ内のグローバル変数 `gWDT_Period_Updater` に格納されます。TAU チャネルの割り込みハンドラはカウンタのオーバーフローによってコールされます。チャネルのカウンタを止め、周期が 75%未満かチェックして WDT のカウンタをリスタートします。TAU チャネルの周期は A/D 変換結果を使用して更新されます。ユーザ LED はトグル出力されて TAU チャネルのカウンタがリスタートされます。ポテンショメータを時計回りに回すと TAU チャネルの周期とユーザ LED の点滅周期が長くなります。TAU チャネル周期が WDT 周期より長くなると割り込みが発生します。WDT 割り込みハンドラは ADC と TAU のオペレーションを停止し、無限ループに入る前に LCD の 2 行目に「75%」を表示させます。
- ⑤ テストを最初から再開する場合は、CPU ボードをリセットしてください。

5.12 RTC

5.12.1 説明

本サンプルコードは RTC（リアルタイムクロック）のデモコードです。RTC は 1Hz の周期的な割り込みを生成し、RTC1HZ ピンに 1Hz の信号を出力するために形成されます。出力はボード上の J2.Pin16 上で観測できます。

5.12.2 オペレーション

- ① すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ② LCD モジュールを初期化し、LCD に「RTC」、「00:00:00」を表示します。
- ③ RTC モジュールを有効にして無限ループに入る前に RTC の 1Hz 出力を有効にします。
- ④ コードは割り込みを待ちます。
- ⑤ 1Hz 周期で割り込みを生成します。割り込みハンドラは `RTC_ConstPeriodInterruptCallback` コールバック関数を実行します。関数は RTC レジスタを読んで、LCD 表示用に変換した値を LCD に表示させます。

5.13 Snooze

5.13.1 説明

本サンプルコードは消費電流を低減するための SNOOZE モードのデモコードです。インターバルタイマの割り込みは CPU と無関係の A/D 変換を引き起こす前に STOP モードから解除するために使用されます。レジスタ、フラグ、データ・メモリのすべてはスタンバイ・モードに設定される直前の内容が保持されます。また、入出力ポートの出力ラッチ、出力バッファの状態も保持されます。

本サンプルコードの初期設定は LCD を使用しません。LCD は A/D 変換結果を確認するために使用することができます。 「r_cg_userdefine.h」ファイル中の以下に示す定義命令のコメントをオフにすると LCD を使用できます。

```
//#define USE_LCD
```

5.13.2 オペレーション

- ① ボード上の抵抗 R22 を外してください。ジャンパ J6 を実装してジャンパ J6 に電流計を接続してください。
- ② すべての周辺装置およびモジュールは、メイン関数をコールする前に形成されます。
- ③ 「r_cg_userdefine.h」ファイル中の”//#define USE_LCD”のコメントをオフにしている場合、LCD モジュールを初期化し、LCD に「Snooze」、「PressSW1」を表示します。
- ④ 無限ループに入る前に SW1 と SW2 の割り込みを有効にします。
- ⑤ SW1 を押すと CPU が通常動作モードで A/D 変換を開始する割り込みを生成します。再び A/D 変換を行う場合は SW1 を押してください。また、この時の消費電流に注意してください。
- ⑥ STOP 命令を実行し、CPU 動作を停止するために SW2 を押してください。メインクロック（高速オンチップオシレータ）は STOP モード中に切られます。消費電流がステップ⑤よりも低減されることに注意してください。
- ⑦ インターバルタイマを起動するために SW3 を押してください。インターバルタイマの期間終了は CPU を STOP モードから解除し、CPU を動作せずに A/D 変換を引き起こします。LCD の使用を有効にしている場合、LCD に変換結果が表示されます。一旦 CPU が復帰すると、ADC は SNOOZE モードを無効にします。

6 追加情報

サポート

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。

RL78/G13 マイクロコントローラに関する詳細情報は、RL78/G13 ユーザーズマニュアルハードウェア編を参照してください。

アセンブリ言語に関する詳細情報は、RL78 ファミリユーザーズマニュアルソフトウェア編を参照してください。

オンラインの技術サポート、情報等は以下のウェブサイトより入手可能です：

<http://japan.renesas.com/rskrl78g13> (日本サイト)
<http://www.renesas.com/rskrl78g13> (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ます。ルネサス エレクトロニクス株式会社の書面での承諾無しに、本書の一部または全てを複製することを禁じます。

© 2011 Renesas Electronics Corporation. All rights reserved.
© 2011 Renesas Electronics Europe Limited. All rights reserved.
© 2011 Renesas Solutions Corp. All rights reserved.

改訂記録	RSKRL78G13 ソフトウェアヘルプマニュアル(CubeSuite+)
------	---------------------------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2011.08.31	－	初版発行

RSKRL78G13 ソフトウェアヘルプマニュアル(CubeSuite+)

発行年月日 2011年8月31日 Rev.1.00

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・電話番号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 エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2 (日本ビル)

(03)5201-5307

■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/inquiry>

RL78/G13