

ブレイクポイントの使用法

本資料は英語版を翻訳した参考資料です。内容に相違がある場合には英語版を優先します。資料によっては英語版のバージョンが更新され、内容が変わっている場合があります。日本語版は、参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントを参照ください。

要旨 (Introduction)

ブレイクポイントの概念は、指定された命令 (specified instruction) の前で、プログラムの実行を中断することです。ソフトウェアまたはハードウェアのどちらでも実装が可能ですが、本書では実装方法については説明していません。ブレイクポイントはシンプルな機能ですが、組み合わせて使用することでバグを容易に解決することができます。実際に、ソフトウェアおよびファームウェア開発者にとって、ブレイクポイントは欠かせないものです。では、どのようにして活用するのでしょうか？ 本書では、ブレイクポイントの機能を活用して、ユーザーアプリケーションを速く、効率的にデバッグする方法を説明しています。

目次

1. ブレイクポイントの使用 (Use of Breakpoints)	2
1.1 コードブレイクポイント (Code Breakpoints)	2
1.2 条件付きコードブレイクポイント (Conditionnal Code Breakpoints)	3
1.3 リード/ライトアクセスによるデータブレイクポイント (Data Breakpoints with Read and Write Access)	5
1.4 ログブレイクポイント (Log Breakpoints)	6
1.5 電源ブレイクポイント (Power Breakpoints)	8

1. ブレークポイントの使用 (Use of Breakpoints)

IAR Embedded Workbench® for Renesas Synergy™ (IAR EW for Synergy) では、以下のブレークポイントを使用することが可能です。

1.1 コードブレークポイント (Code Breakpoints)

これは、ブレークポイントの最もシンプルな使用方法です。Cソースプログラムのいずれかの行 (line) を選択するか、逆アセンブルウィンドウ (disassembly window) でASM命令を選択し、ブレークポイント (breakpoint) をトグル (toggle) すればよいだけです。ブレークポイントがヒットすると、アプリケーションは停止します。この時点で、変数 (variable) 、フラグ (flag) 、およびレジスタ (register) の値を確認できます。

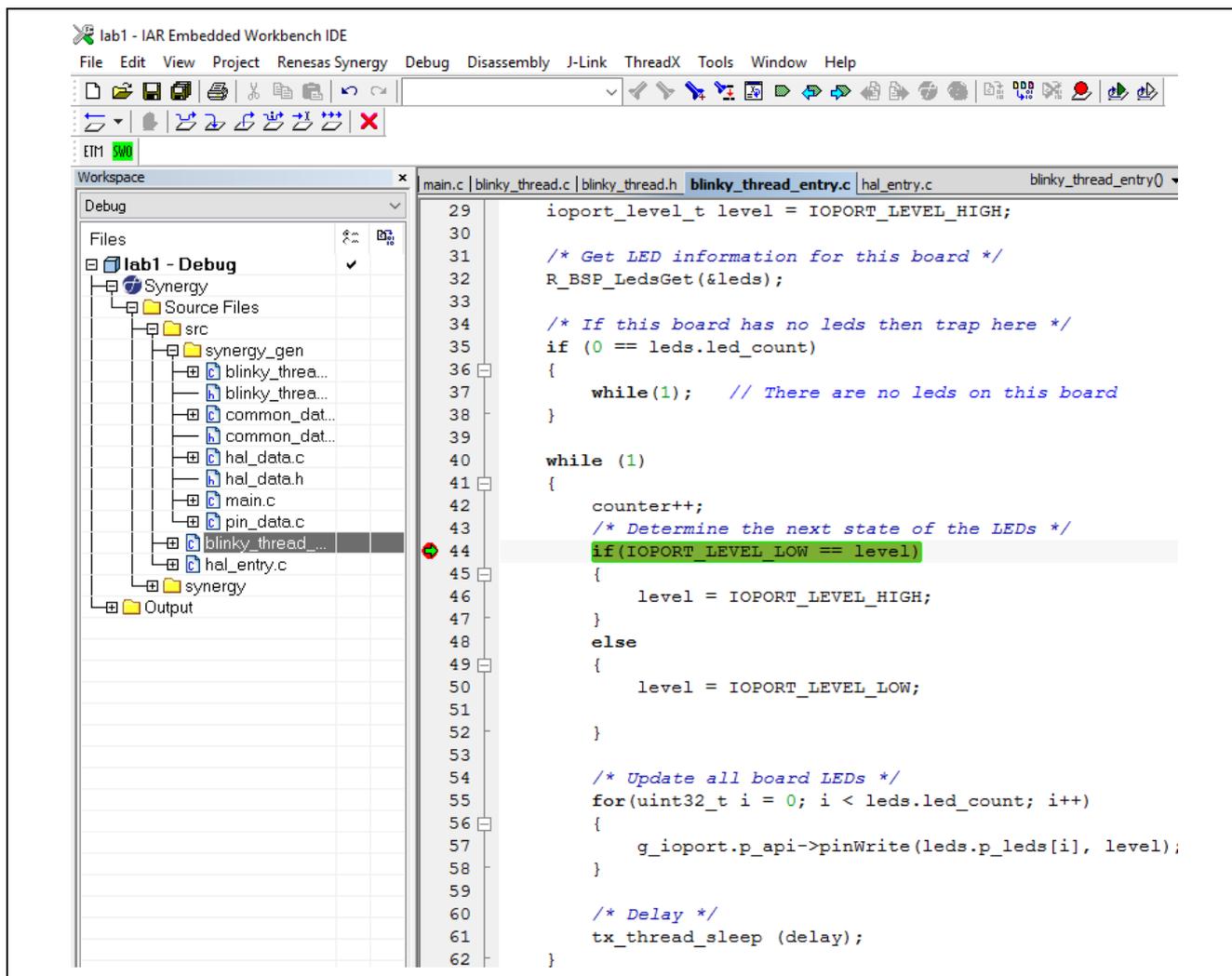


Figure 1. コードブレークポイント

コードブレークポイント (code breakpoint) の数は、ハードウェアブレークポイント (hardware breakpoint) の数に制限されます。ただし、ソフトウェアブレークポイント (software breakpoint) を利用するか、またはRAM上でアプリケーションを実行すれば、ブレークポイント数の制限はなくなります。たとえば、6個のブレークポイントを持つRenesas Synergyデバイスの場合、ブレークポイントの数が制限されていても、ブレークポイントの位置情報 (location) を保存し、必要に応じてブレークポイントを有効/無効 (disable/enable) にすることが可能です。「View」→「Breakpoints」ウィンドウを選択するだけで、ボックスの**設定/クリア (set/clear)** が可能になります。これは、ブレークポイントの**有効化/無効化 (enabling/desabling)** を意味します。

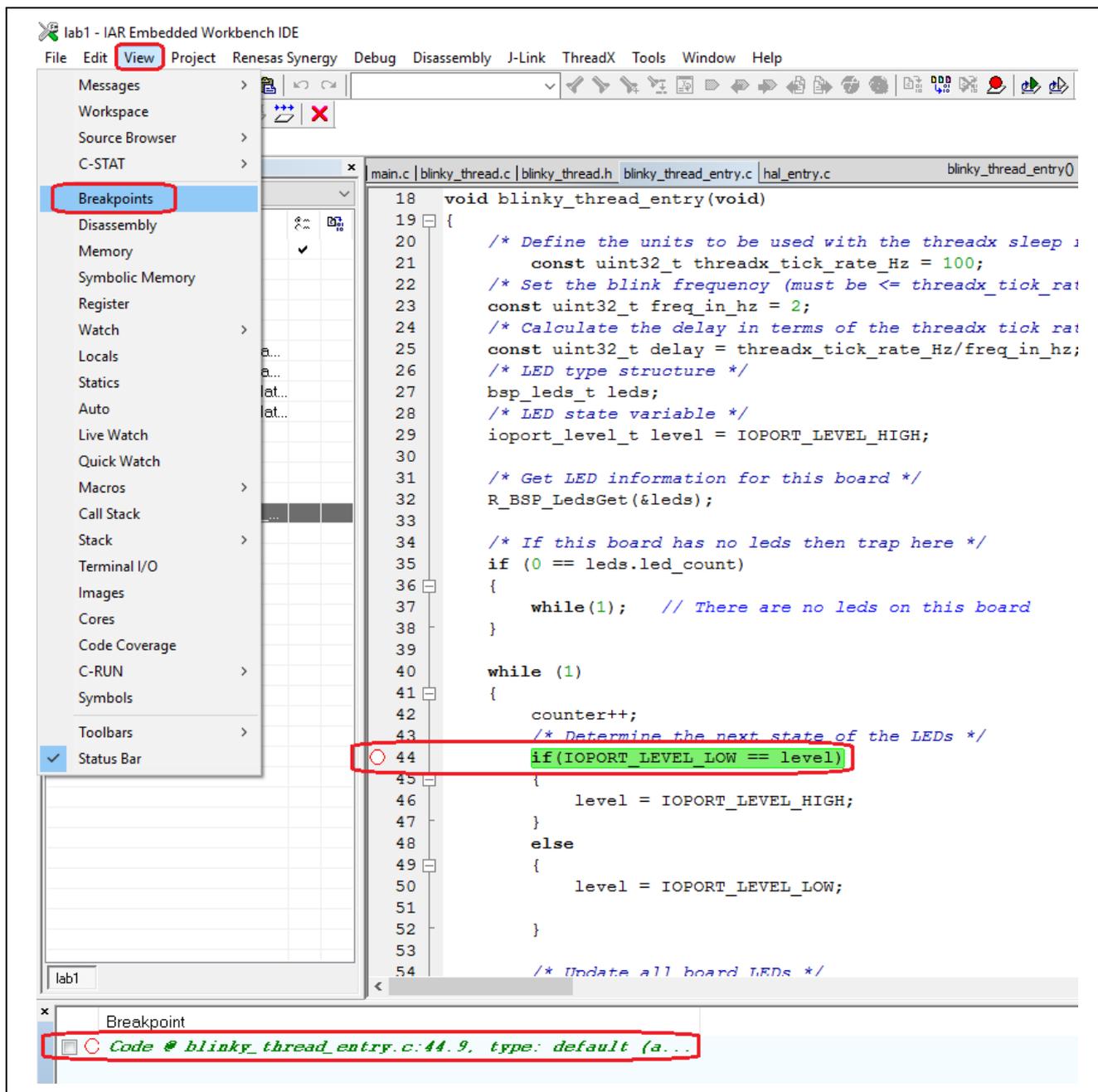


Figure 2. ブレイクポイントの表示

この場合、6個を超えるブレイクポイントを保持できますが、すべてが同時にアクティブ (active) になるわけではありません。J-Link™およびI-Jet™もソフトウェアブレイクポイントの機能を提供しています。これを使用するとフラッシュのブレイクポイントの数は無制限になりますが、いくつかのメモリブロックを再プログラミングするコストがかかり、デバッグが遅くなるという欠点があります。

1.2 条件付きコードブレイクポイント (Conditionnal Code Breakpoints)

条件付きブレイクポイント (conditional breakpoint) は、条件として何らかのフラグ (flag) または変数 (variable) を含むコードブレイクポイント (code breakpoint) を組み合わせたものです。ブレイクポイントを設定したら、「View」→「Breakpoints」ウィンドウ (window) を使用してブレイクポイントをすべて確認できるだけでなく、右クリックして「Edit」オプション (option) を選択することで、追加のパラメータを設定することも可能です。

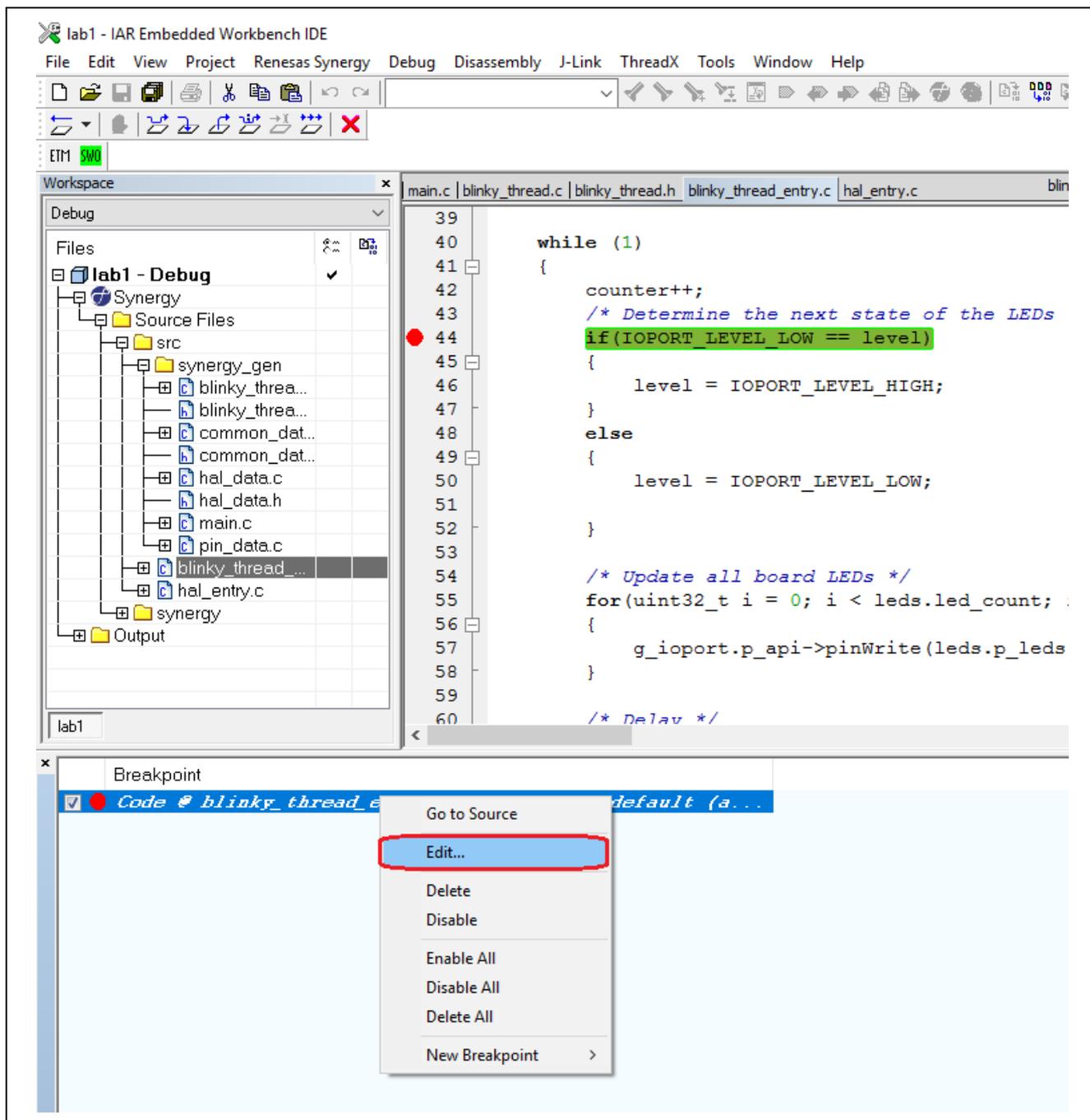


Figure 3. 条件付きコードブレイクポイント

使用される構文 (syntax) はC構文 (syntax) に似ており、`==`、`>=`、および`<=`の条件を指定することが可能です。たとえば、カウンタ (counter) が10になったときにブレイクポイント (breakpoint) でアプリケーションを停止したい場合は、`counter==10`を使用します。

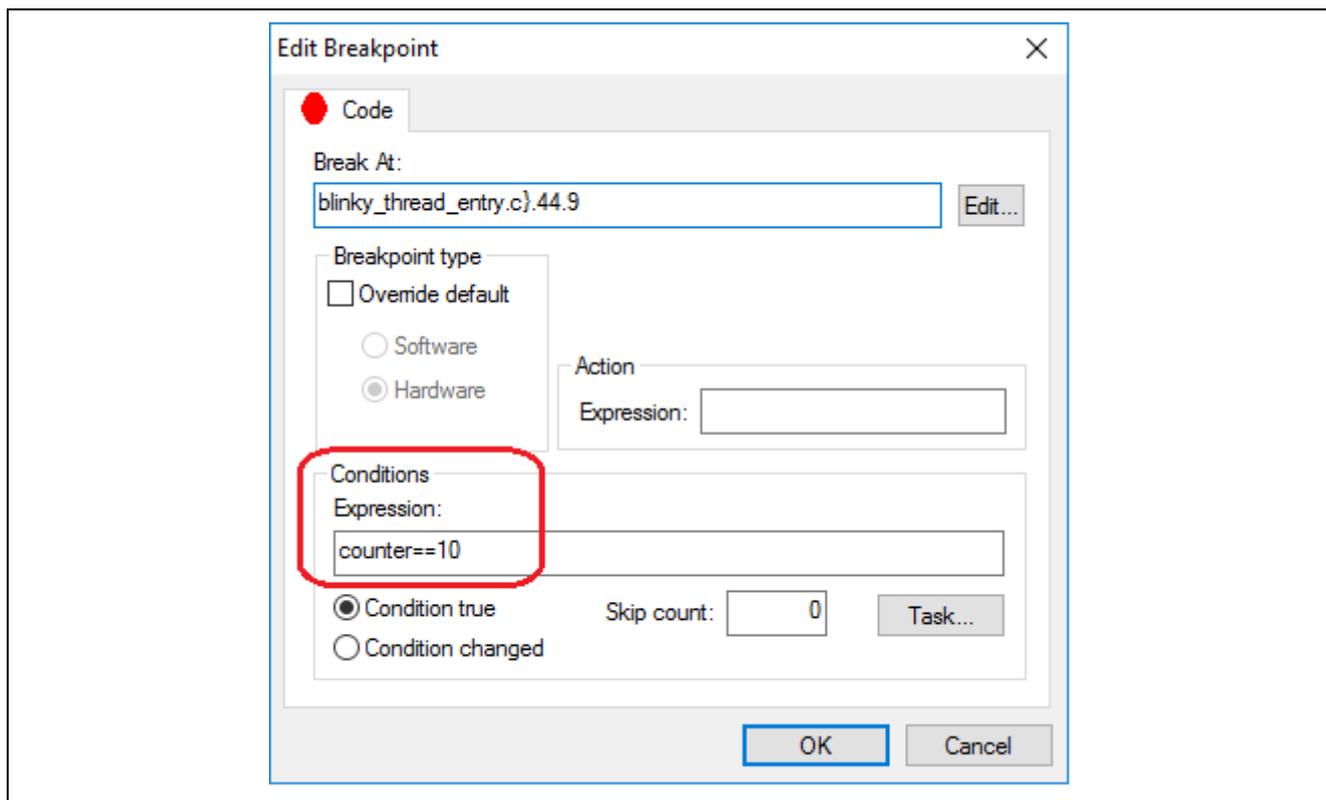


Figure 4. コードブレークポイントの編集

これは、割り込みルーチン (interrupt routine) 内部でブレークポイント (breakpoint) が必要な場合、とても役立ちます。条件の指定がなければ、アプリケーションは常に停止するので、アプリケーションのデバッグが不可能になります。フラグ (flag) または変数 (variable) を条件として使用すると、かなり簡単に処理できます。また、スキップカウンタ (skip counter) および条件チェック (condition check) (true または changed) を使用するとさらに便利です。

1.3 リード/ライトアクセスによるデータブレークポイント (Data Breakpoints with Read and Write Access)

データブレークポイントは、特定のメモリアドレス (specific memory address)、フラグ (flag)、変数 (variable)、またはレジスタ (register) へのリード/ライトアクセスを監視します。データブレークポイントを使用するのはとても簡単で、フラグ (flag) または変数 (variable) を右クリックし、「Set Data Breakpoint」オプション (option) を選択するだけです。リード/ライトアクセスは、デフォルトで監視されます。何らかの設定を追加したい場合は、「View」->「Breakpoints」ウィンドウ (window) および「Edit」オプション (option) で追加することが可能です。アクセスだけではなく、データが一致 (data match) するかどうか監視できます。これは、データが一致する場合に限り、リード/ライトアクセスは停止 (halt) にトリガ (trigger) をかけます。「Edit」ボタンを選択すると、絶対アドレス (absolute address) またはソース行 (source line) も選択できる追加のウィンドウが開きます。変数またはフラグの場合、自動サイズ調整 (auto size) の使用が推奨されますが、より広い範囲を監視する必要がある場合は、目的のサイズに合わせて手動サイズ調整 (manual size) を使用してください。

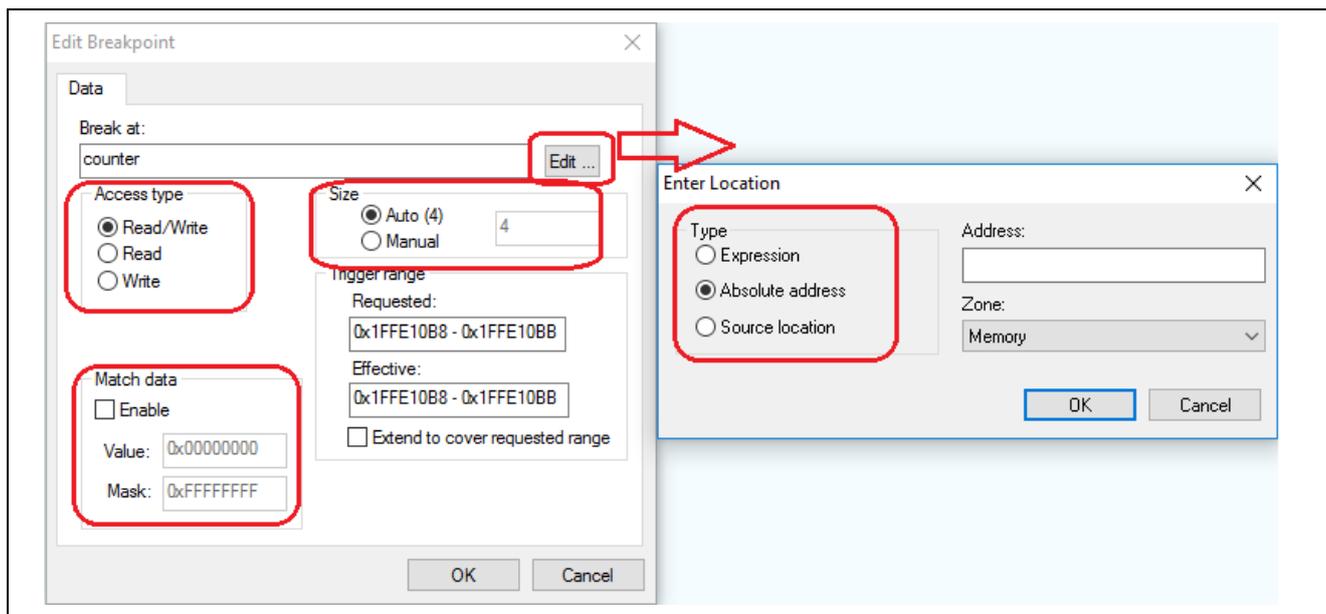


Figure 5. データブレークポイント

データブレークポイントは、アプリケーションによって破損 (corrupted) されているフラグと変数をデバッグする上でとても有効です。何らかのアクセスがあると、アプリケーションは停止します。もう1つの方法は、スタックオーバーフローの調査 (stack overflow investigation) です。スタックサイズの80~90%にデータブレークポイントを設定するだけで、オーバーフローに近づいたときにアプリケーションを停止し、問題の原因を絞り込みが可能です。

1.4 ログブレークポイント (Log Breakpoints)

コードブレークポイントおよびデータブレークポイントのほかに、ログブレークポイント (log breakpoint) があります。これは、アプリケーションを一時的に停止してメッセージのみ出力するだけの特別なブレークポイントです。ブレークポイントがヒット (hit) すると、選択されたメッセージを表示するだけです。

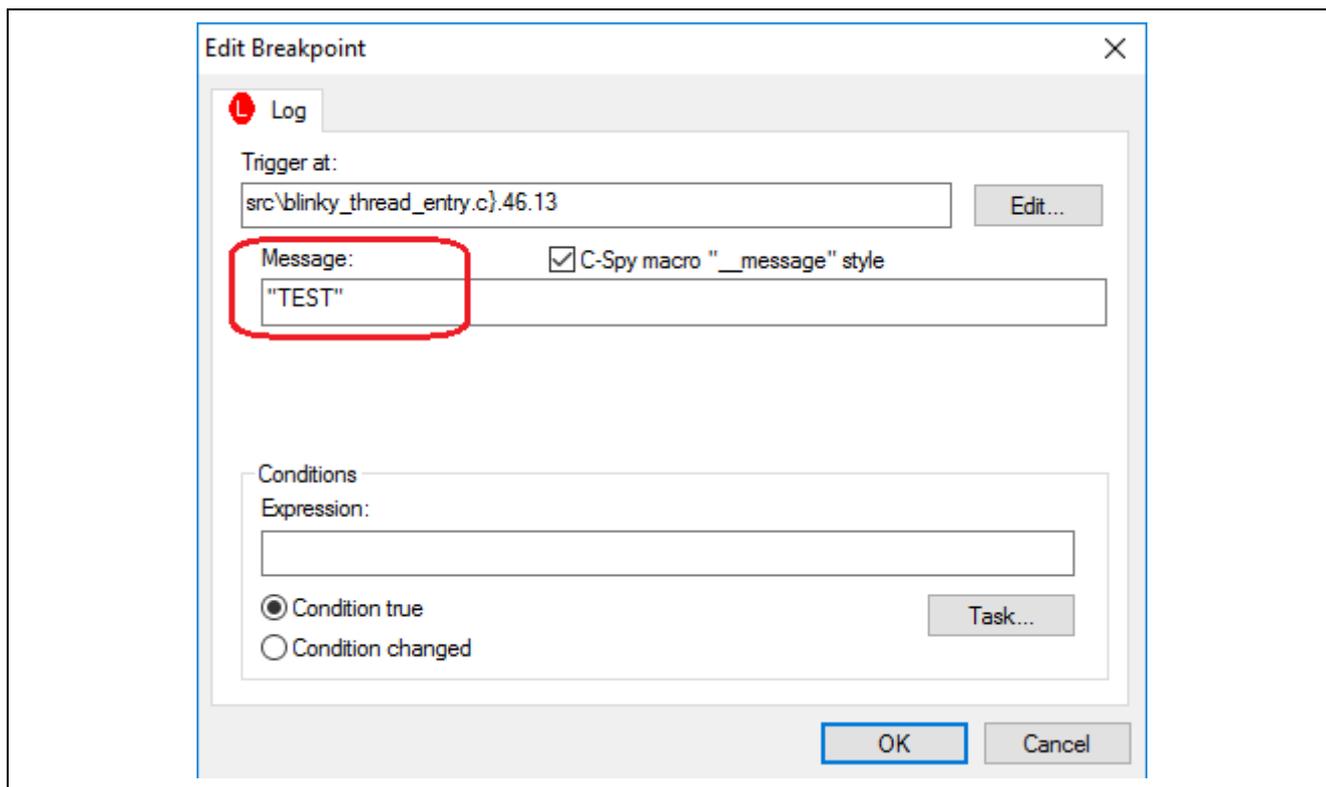


Figure 6. ログブレークポイント

ブレークポイントがヒットするたびに、デバッグログウィンドウ (debug log window) にメッセージが表示されます。追加されたカウンタにより、アプリケーションがソースコードの該当箇所を通過した回数を確認することが可能です。

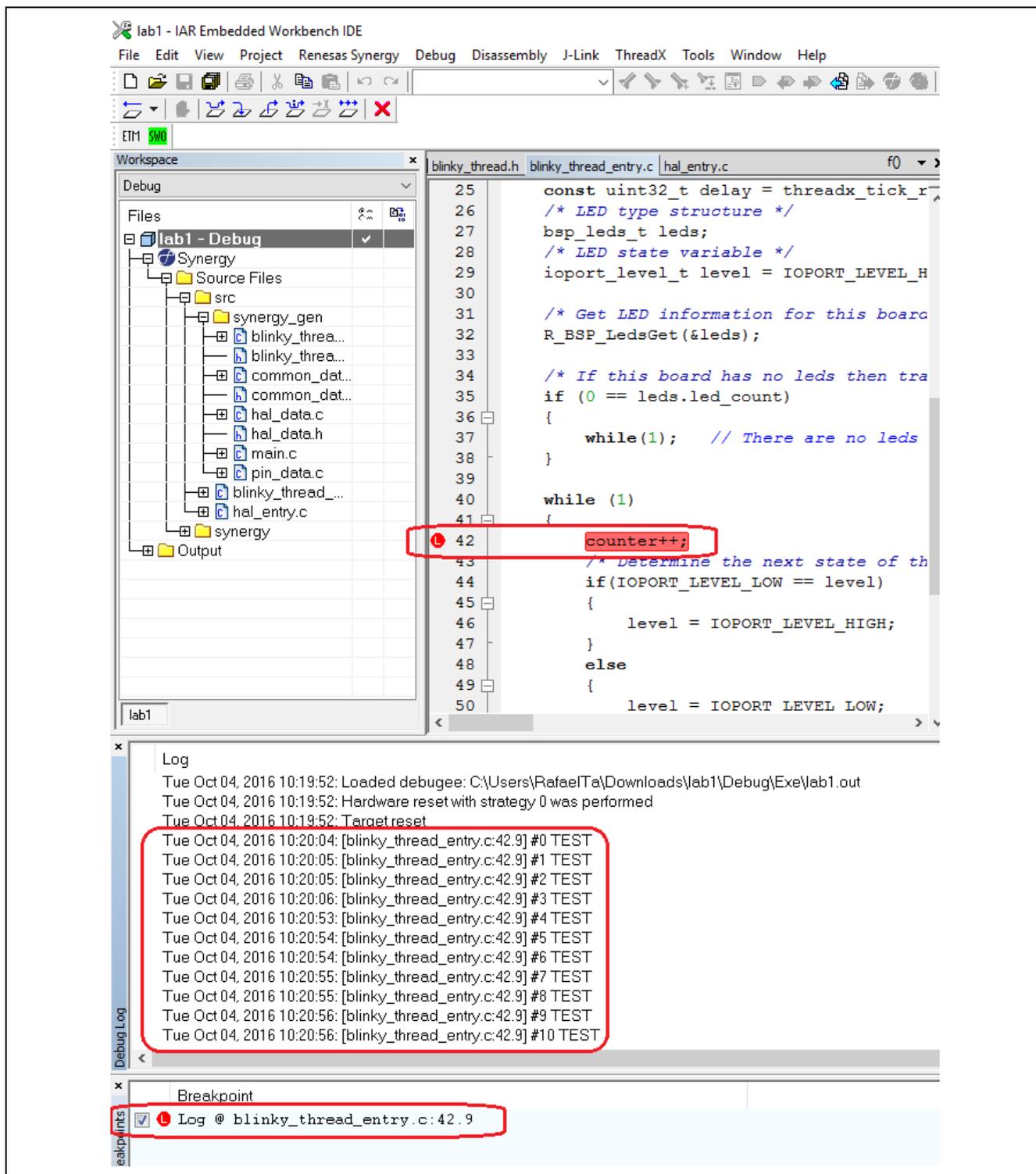


Figure 7. デバッグログウィンドウ

1.5 電源ブレイクポイント (Power Breakpoints)

IAR EW for Synergyなどには、エネルギー消費量 (energy consumption) を監視してソースコードに関連付けることができる**電源デバッグ (Power Debugging)** の概念が導入されています。これにより、アプリケーション全体のエネルギー消費量を確認することが可能になります。この概念によって**電源ブレイクポイント (power breakpoint)** の追加が可能になります。例えば、215mAなどのしきい値 (threshold) を設定することも可能です。エネルギーがこの値を上回ると、デバッガは停止します。

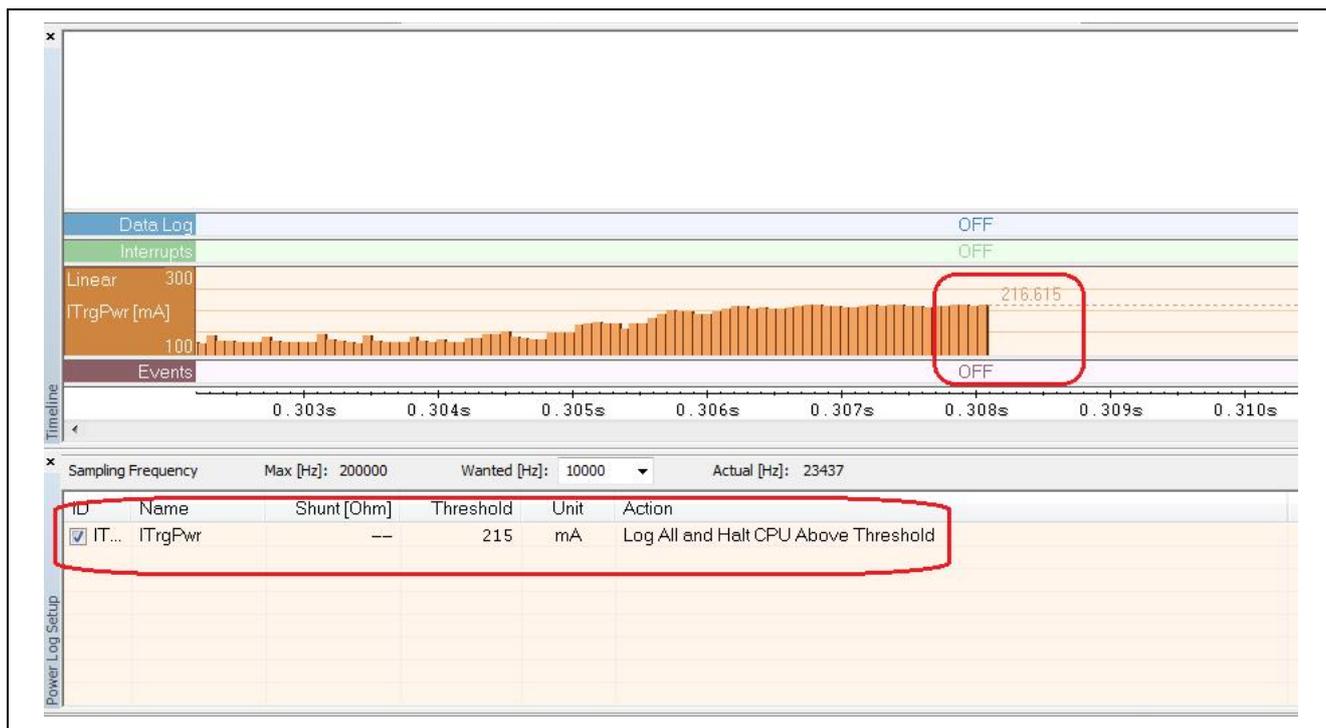


Figure 8. 電源ブレイクポイント

しきい値の設定はとても簡単です。「J-Link」または「I-JET/JTAGJet」->「Power Log Setup」ウィンドウ (window) を開いて、値 (value) とオプション (値を上回るか下回るかなど) を設定するだけです。この機能は、J-Linkボードでは使用できないことに注意してください (J-Linkボードは、Synergy SKおよびDKキットの大部分で利用可能)。ターゲットへのプローブ供給 (probe supply) が450mAに制限されるため、外部電源 (external power supply) にはI-Jetと連携するI-Scopeが推奨されます。

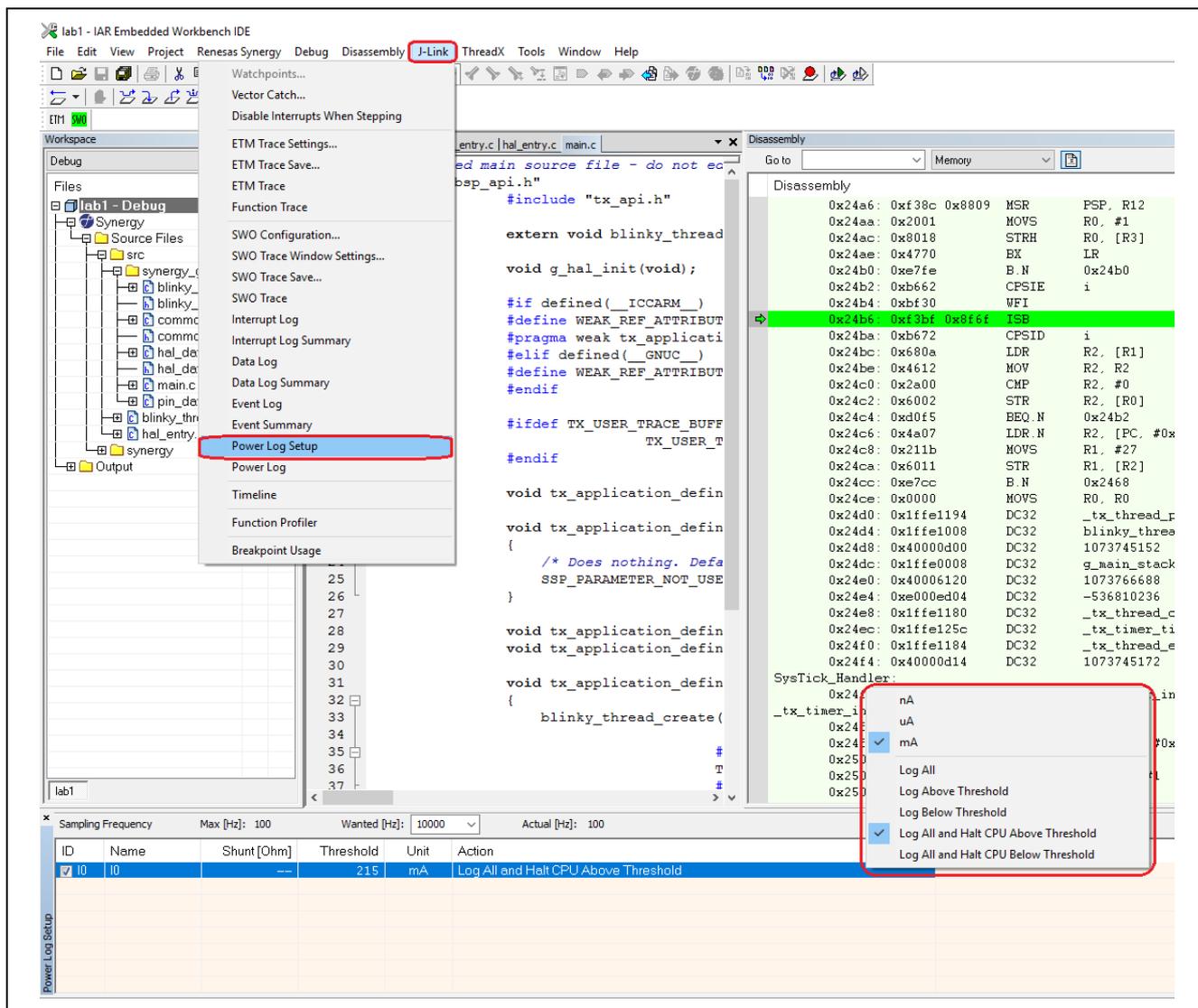


Figure 9. 電源ログ設定

この機能は、電力消費量のピークにならないこと、または指定より高い値にならないことを確実にする上で役立ちます。また解析によって、バッテリーの持続時間が長くなることを確実にする上でも役立ちます。アプリケーションを長期間にわたって実行することが可能になります。タイムライン (timeline) は必要ありませんが、タイムラインがあれば、消費されているエネルギーの完全な情報が得られます。

J-LinkおよびI-JetのIAR EW for Synergyでは、さまざまなタイプのブレイクポイント (breakpoint) が利用可能です。ここまで、ブレイクポイントのさまざまな組み合わせや使用方法をご説明しました。ブレイクポイントを最大限に活用しながら、アプリケーションをデバッグすることが可能になります。

ホームページとサポート窓口

サポート : <https://synergygallery.renesas.com/support>

テクニカルサポート :

- アメリカ : https://renesas.zendesk.com/anonymous_requests/new
- ヨーロッパ : <https://www.renesas.com/en-eu/support/contact.html>
- 日本 : <https://www.renesas.com/ja-jp/support/contact.html>

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂履歴

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2018.06.15	-	初版 英文版（資料番号 r11an0062eu0100-synergy-uses-of-breakpo ints、リビジョンRev1.00、発効日2017年7月10日）を翻訳

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
 4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
 5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、
金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。
 6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
 7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
 9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
 11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
 12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.4-0-1 2017.11)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

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

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

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<https://www.renesas.com/contact/>