RX ファミリ

既存の周辺機能を応用したタンパ検知方法

要旨

本アプリケーションノートは、既存の周辺機能であるデータトランスファコントローラ(DTC)やイベントリンクコントローラ(ELC) 、シリアルコミュニケーションインタフェース (SCI)等を応用することで悪意ある第三者からのタンパ検出を実現する方法を説明します。
本アプリケーションノートの対象デバイスは、以下です。本アプリケーションノートの内容を動作確認した RX66N グループ以外の RX ファミリに適応する場合は、対象マイコンの仕様にあわせて変更し十分評価してください。

対象デバイス

DTC,ELC,TMR, MTU3a,DOC,SCI および TSIP(TRNG 内蔵)を搭載する RX ファミリ

動作確認デバイス

RX66N グループ

目次

1. 概要 4
1.1 背景 4
1.2 タンパ検出機能 4
1.3 システム構成 6
1.4 メモリマップ 7
1.5 LED による状態表示 7
1.6 シーケンスの定義 8
1.6.1 MTU0.TGRA の設定 8
1.7 注意事項 9
1.7.1 実装時の注意事項 9
1.7.2 動作中の注意事項 9
1.7.3 使用モジュールの注意事項 9

2. 動作確認条件 10

3. 動作環境 11
3.1 接続方法 11
3.2 書き込み方法 12

4. システム説明 15
4.1 タンパ検出動作説明 15
4.1.1 DTC の転送動作 16
4.1.2 乱数の bit 長整合 17
4.1.3 タンパ検出時の消去 18
4.2 1 シーケンスの動作詳細 19
4.2.1 電源投入から 1 シーケンス目の転送開始時の挙動 20
4.2.2 タンパ発生時の動作詳細 20

5. ソフトウェア説明 21
5.1 構成 21
5.1.1 ファイルの構成 21
5.1.2 変数一覧 21
5.1.3 関数一覧 22
5.1.4 関数仕様 23
5.2 フローチャート 27
5.2.1 全体動作フロー 27
5.2.2 メイン関数の初期化および ActiveTAMP スタートのフロー 28
5.2.3 タンパ検出処理のフロー 29
5.2.4 タイマによる比較不一致回数が 2 回以上時のフロー 31
5.2.5 通信エラー割り込み処理 31
5.2.6 機密情報削除関数 32
5.3 フットプリント 33
5.4 注意事項 33
5.4.1 タンパ検出を停止させる場合の注意事項 33

6. 参考ドキュメント 33

改訂記録 34

1. 概要

1.1 背景

IoT 化が進む現代では、悪意のある第三者が IoT デバイスを狙った不正な改造や内部データの改ざんなどが懸念されます。
これを防止する手段の 1 つとしてハードウェア本体の改造や改ざんを検出しセキュリティ強度の向上を図るタンパ検出機能があり、RX ファミリの既存の周辺機能を応用することでタンパ検出の実現が可能です。

1.2 タンパ検出機能

一般的なレベル監視によるタンパ検出とは異なり、本サンプルプログラムでは、乱数を送信し、これを外部で折り返して受信。そしてこれを比較することでタンパの有無を監視します。また、外部の折り返し信号の断線によるフレーミング、パリティ、オーバランエラーなどの通信エラーをタンパとして検出します。タンパ検出時は、内蔵 SRAM 及びバックアップレジスタを削除し、課金情報や個人情報などの機密情報の漏洩を防ぎます。

図 1-1、図 1-2 にレベル監視と本サンプルプログラムのシステム構成図を示します。例として、筐体内にスイッチ（SW）を設置し、筐体の開閉を検出します。

図 1-3 にレベル監視によるタンパ検出方法を示します。TAMPn 信号のレベル変化を検出するとタンパ検出となります。こちらも同じく筐体の開閉を検出します。

図 1-4、図 1-5 に本サンプルプログラムによるタンパ検出方法を示します。本サンプルプログラムではレベル監視によるタンパ検出ではなく、送受信した乱数の比較で規定回数不一致時、もしくは通信エラー発生時にタンパ検出となります。

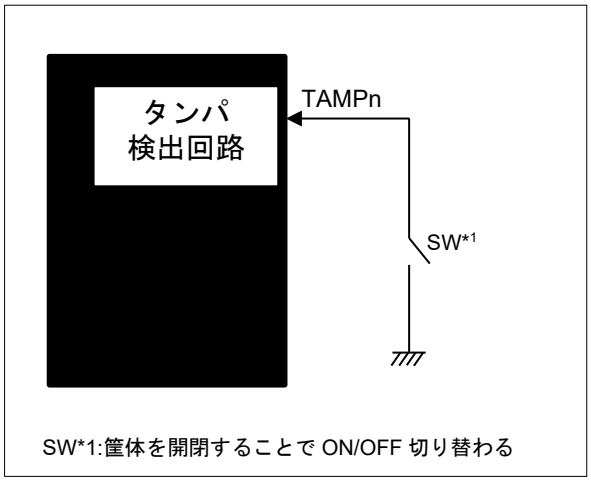


図 1-1 レベル監視システム構成図(n=0~2)

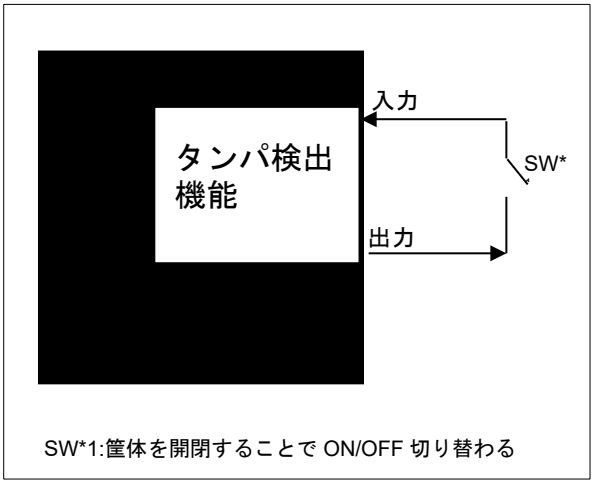


図 1-2 本サンプルプログラムシステム構成図

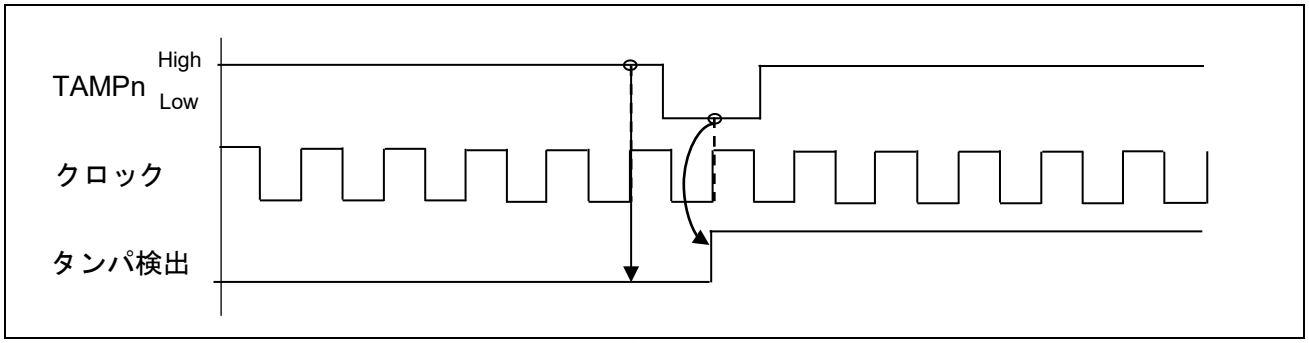


図 1-3 レベル監視によるタンパ検出

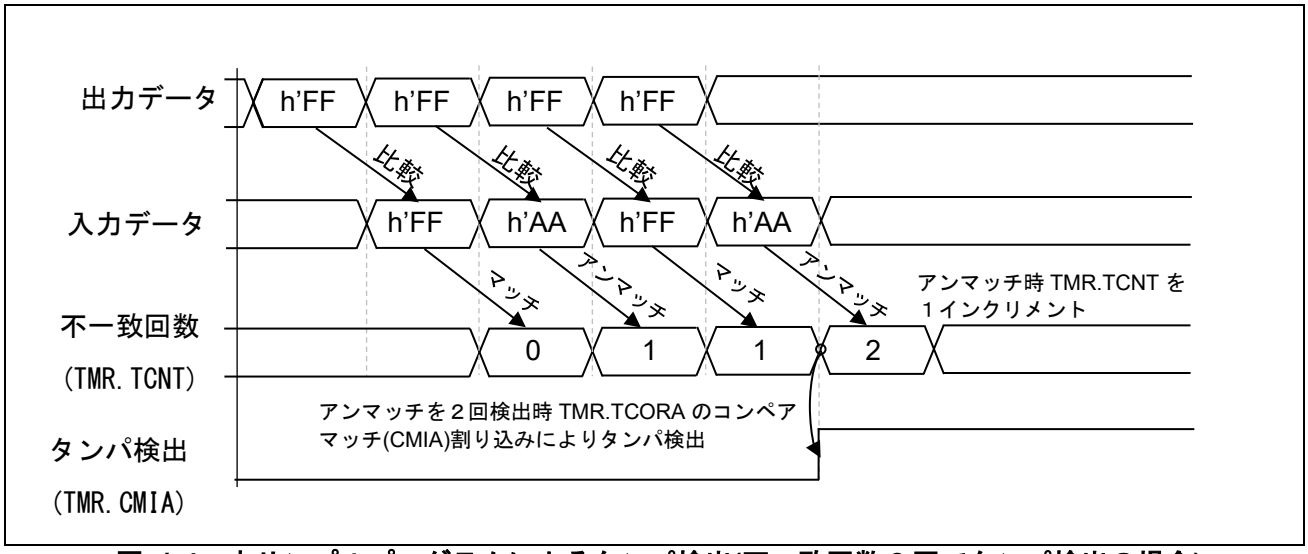


図 1-4 本サンプルプログラムによるタンパ検出(不一致回数 2 回でタンパ検出の場合)

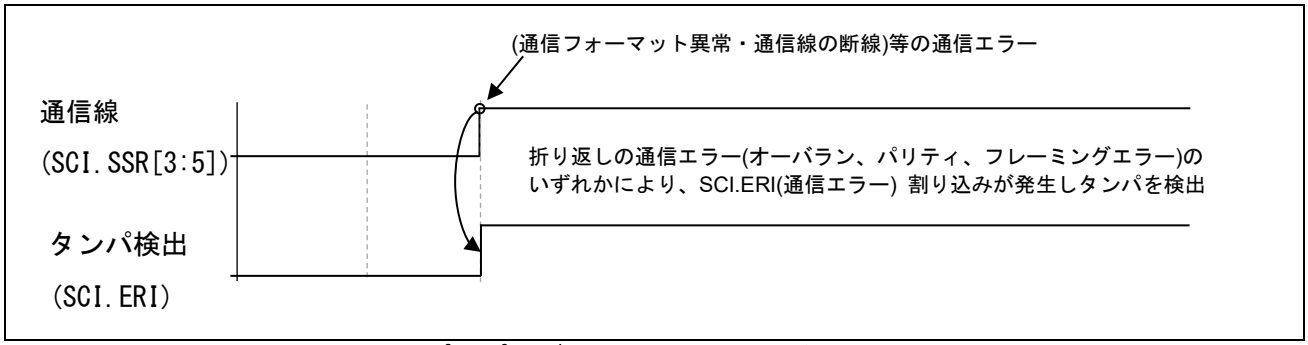


図 1-5 本サンプルプログラムによるタンパ検出(通信エラーの場合)

表 1-1 に本サンプルコードで検出可能なタンパをまとめます。

表 1-1 検出可能なタンパまとめ

タンパ	検出要因
• 筐体の開閉 • 基板上の通信線の破断	• 通信エラー • 送信/受信回数の不一致
• プローブによる通信線のデータ改ざん	• 送受信データ比較による不一致回数が 2 回以上 • 送信/受信回数の不一致

1.3 システム構成

図 1-6 に RX66N を用いたタンパ検出のシステム構成図、表 1-2 に使用する周辺機能と用途を示します。

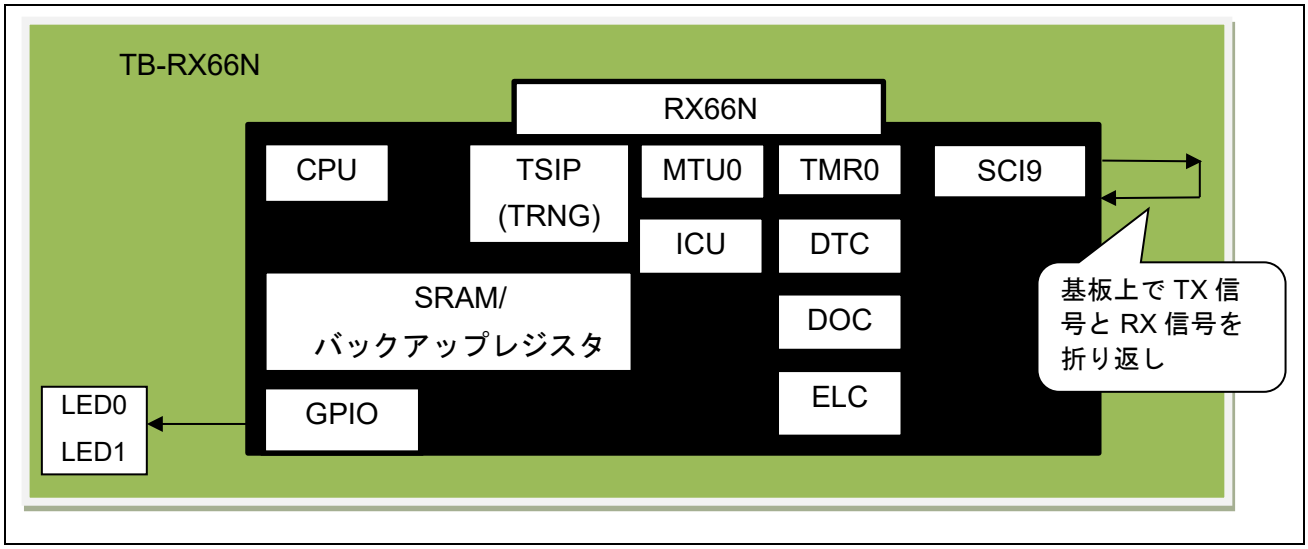


図 1-6 システム構成図

表 1-2 使用する周辺機能と用途

周辺機能	用途
MTU0	後述のシーケンスを管理します。
ICU	周辺モジュールからのさまざまな割り込み要求を管理し、CPU への割り込み要求、または DTC への転送要求を生成します
TSIP 内 TRNG	乱数を生成します。 乱数は CPU で SRAM に書き込みます。
SCI9	乱数を送信し、外部折り返しで受信します。
DTC	TRNG で生成した乱数を SRAM→SCI9 へ転送します。 SCI9 で受信した乱数を DOC へ転送します。 乱数の残り転送回数をカウントします。
DOC	送信した乱数と受信した乱数を比較し、一致/不一致を判定します。
ELC	DOC の不一致信号を TMR0 にイベントリンクします。
TMR0	ELC を介して DOC の不一致信号の回数をカウントします。
GPIO	LED0,1 の点灯/滅灯制御

1.4 メモリマップ

図 1-7 に機密情報の格納された空間や乱数の送受信に使用する空間を表すメモリマップを示します。

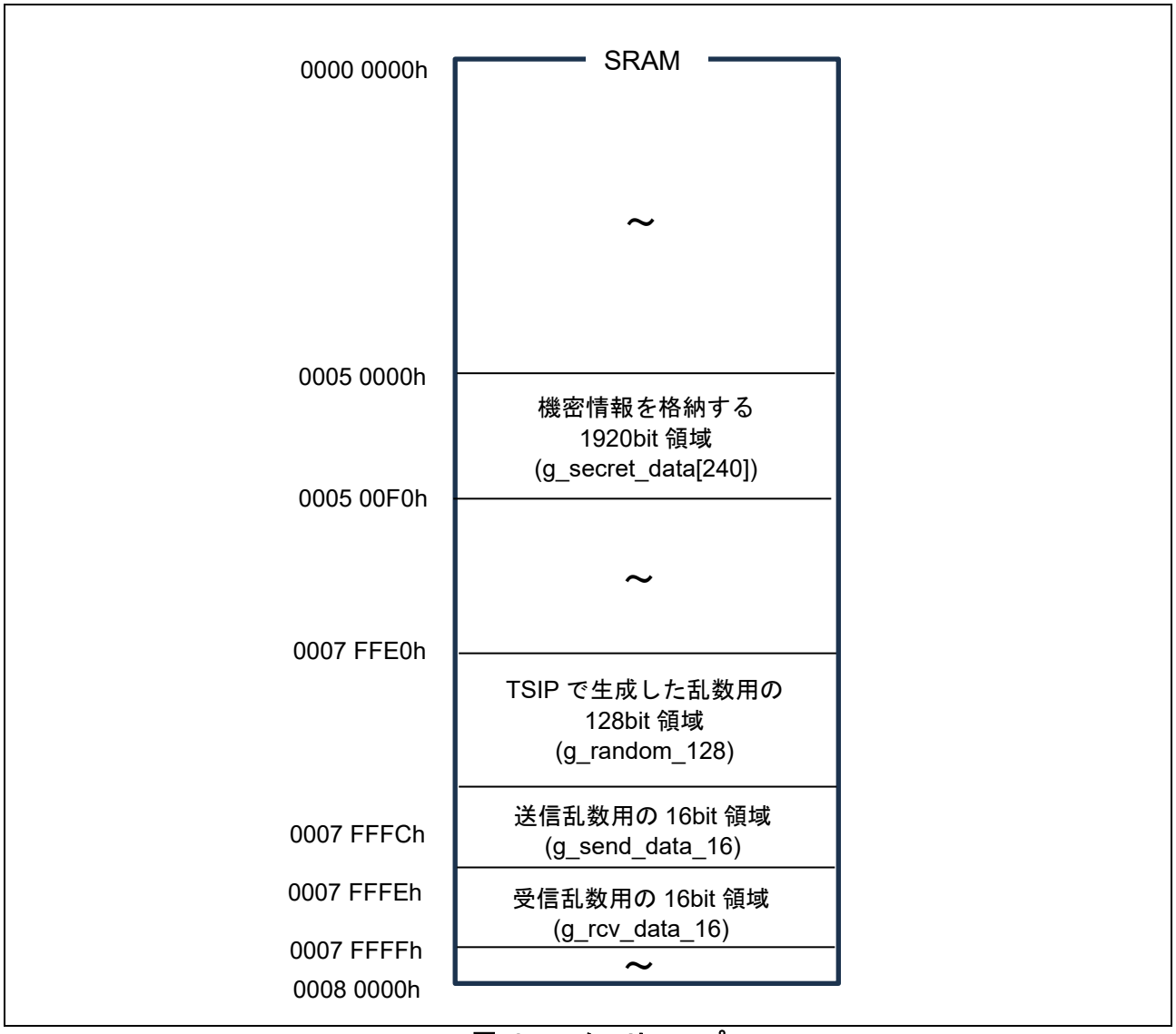


図 1-7 メモリマップ

1.5 LED による状態表示

表 1-3 にタンパ検出を示す LED0、LED1 の状態を示します。

表 1-3 LED によるタンパ検出の状態

0	LED1	状態
OFF	OFF	正常（タンパ検出なし）
OFF	ON	タンパ検出あり 送受信データ比較結果の不一致回数が 2 回以上 (データの改ざん)
ON	OFF	タンパ検出あり 断線(筐体破壊 or 基板改造)
ON	ON	設定なし

1.6 シーケンスの定義

最初の乱数の生成から次の転送開始までをシーケンスとします。シーケンスはタンパ検出時間とアイドル時間で構成され、MTU0.TGRA のコンペアマッチで次のシーケンスが開始されます。

これにより、設定された周期でのシーケンス動作が可能です。

タンパ検出動作の詳細は4.1タンパ検出動作説明を参照ください。

アイドル時間に関しては1.6.1MTU0.TGRAの設定を参照ください。

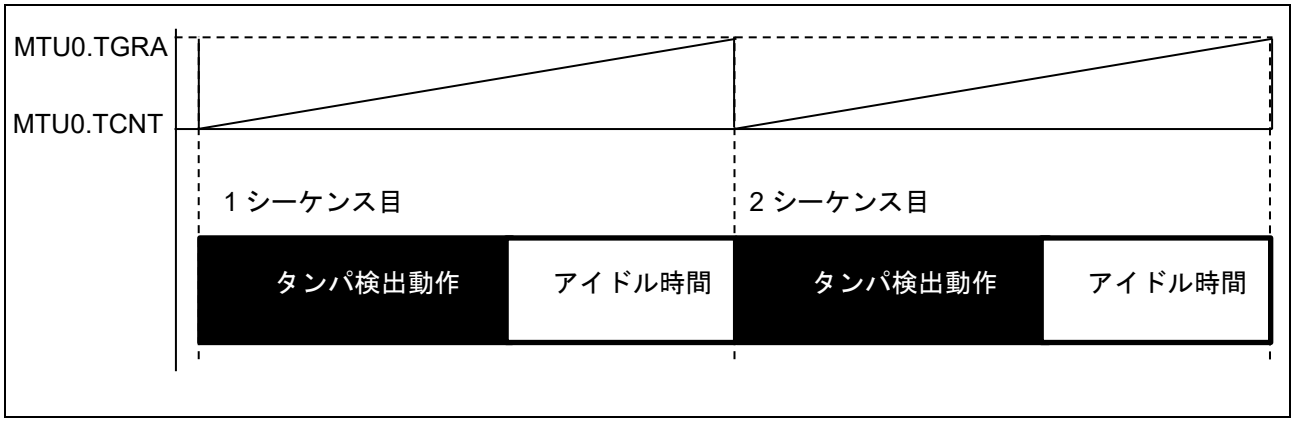


図 1-8 シーケンスの定義

1.6.1 MTU0.TGRA の設定

シーケンスはタンパ検出動作時間とアイドル時間から成ります。

タンパ検出動作時間は、5.6[us]+(1000000/ボーレート[bps])*10*4 [us]で算出できます。

アイドル時間は、ユーザで任意の時間を設定してください。

よって、シーケンスはタンパ検出時間を算出後、ユーザで任意の時間を加算した合計時間を MTU0.TGRA に設定してください。

例:ボーレート 9600bps 時、アイドル時間を 100nS としたい場合、
タンパ検出動作時間は上記式より 4172.26[us]、アイドル時間を 100[ns]とするため、
MTU0.TGRA には 2 つを加算した 4272.36[us]より大きくなるように設定してください。

本計算式には受信完了にかかる時間は考慮されていません。
そのため実際には、受信にかかる時間を考慮の上調整ください。

1.7 注意事項

1.7.1 実装時の注意事項

本サンプルコードでは、外部折り返しが断線すると、いかなる場合でも機密情報を削除する仕様となっています。正規のメンテナンスのために、筐体の開閉を行う場合には、開閉を行う前に機密情報をクラウドに上げる、特定の操作でタンパ検出を停止させるなどの対応をお願いします。

タンパ検出を停止する方法は、**5.4.1 タンパ検出を停止させる場合の注意事項**を参照ください。

1.7.2 動作中の注意事項

本サンプルプログラム動作中は通信線と電源、GND との短絡また、通信線のノイズにご注意ください。タンパとして検出される可能性があります。

1.7.3 使用モジュールの注意事項

本サンプルプログラムを使用する場合、DOC は 1 チャネルしか搭載されていないためユーザプログラム内では使用することができません。その他のモジュールは、本サンプルプログラム内で使用していないチャネルを使用することができます。

2. 動作確認条件

本アプリケーションノートのサンプルプログラムは、以下に示す条件で動作を確認しています。

表 2-1 動作確認環境

項目	内容
使用 MCU	R5F566NNHDFP (Target Board for RX66N 搭載)
動作周波数	メインクロック：停止(HOCO を使用) PLL：240MHz（HOCO x 1/1 x 15） HOCO：16MHz LOCO：停止 システムクロック（ICLK）：120MHz（PLL x 1/2） 周辺モジュールクロック A（PCLKA）: 120MHz（PLL x 1/2） 周辺モジュールクロック B（PCLKB）: 60MHz（PLL x 1/4） 周辺モジュールクロック C（PCLKC）: 60MHz（PLL x 1/4） 周辺モジュールクロック D（PCLKD）: 60MHz（PLL x 1/4） Flash IF クロック（FCLK）：60MHz（PLL x 1/4）
動作電圧	3.3V
総合開発環境	ルネサスエレクトロニクス e² studio Version 2024-07.0
C コンパイラ注	ルネサスエレクトロニクス C/C++ Compiler Package for RX Family V3.06.00
RX スマート・コンフィグレータ	V2.22.0
ボードサポートパッケージ(r_bsp)	V7.50
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルコードバージョン	V1.00
使用ボード	Target Board for RX66N （型名：RTK5RX66N0C00000BJ）
エミュレータ	E2-Lite

注 本プロジェクトで指定するツールチェーン(C コンパイラ）と同一のバージョンがインストールされていない場合は、ツールチェーンが選択されない状態になり、エラーが発生します。
エラーが発生した場合は、プロジェクトの設定画面でツールチェーンの選択状態を確認してください。

設定方法は、FAQ 3000404 を参照してください。

FAQ 3000404 :インポートしたプロジェクトをビルドすると「PATH でプログラム"make"が見つかりません」エラーになる(e² studio)

3. 動作環境

本アプリケーションのサンプルプログラムのプロジェクトは、Target Board for RX66N と開発用 PC を USB で接続し、書き込む必要があります。

ここでは、Target Board for RX66N 搭載のオンチップデバッギングエミュレータを使用します。

表 3-1 にサンプルプログラムの書き込み環境を、**図 3-1** にその接続方法を示します。

表 3-1 サンプルプログラムの書き込み環境

項目	内容
開発環境	・開発用 Windows PC ・e^2 studio Version 2024-07.0
デバッグツール	・E2 Lite(RX)
ボード	・Target Board for RX66N （型名：RTK5RX66N0C00000BJ）
ケーブル	・USB ケーブル(type-A ⇔ type micro B) ・ジャンパケーブル(オス ⇔ オス)

3.1 接続方法

a. Pmod コネクタ(CN1 の 9 番ピンと 10 番ピン)にジャンパケーブル(オス ⇔ オス)を接続します。

b. USB ファンクションケーブルの type micro B 側を Target Board for RX66N の USB コネクタに接続します。

c. この USB ケーブルの type-A 側を開発用 PC の USB ポートに接続します。これでオンチップデバッギングエミュレータに接続されます。

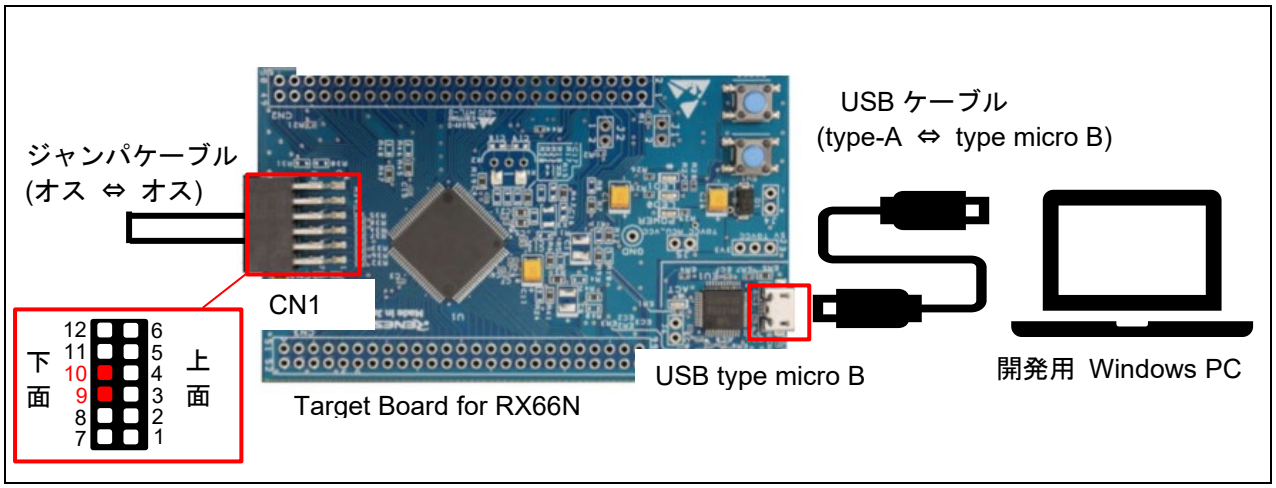


図 3-1 Target Board for RX66N 接続図

3.2 書き込み方法

接続が完了したら、以下手順に従いプロジェクトの書き込みを行います。

a. プロジェクトをインポートします。

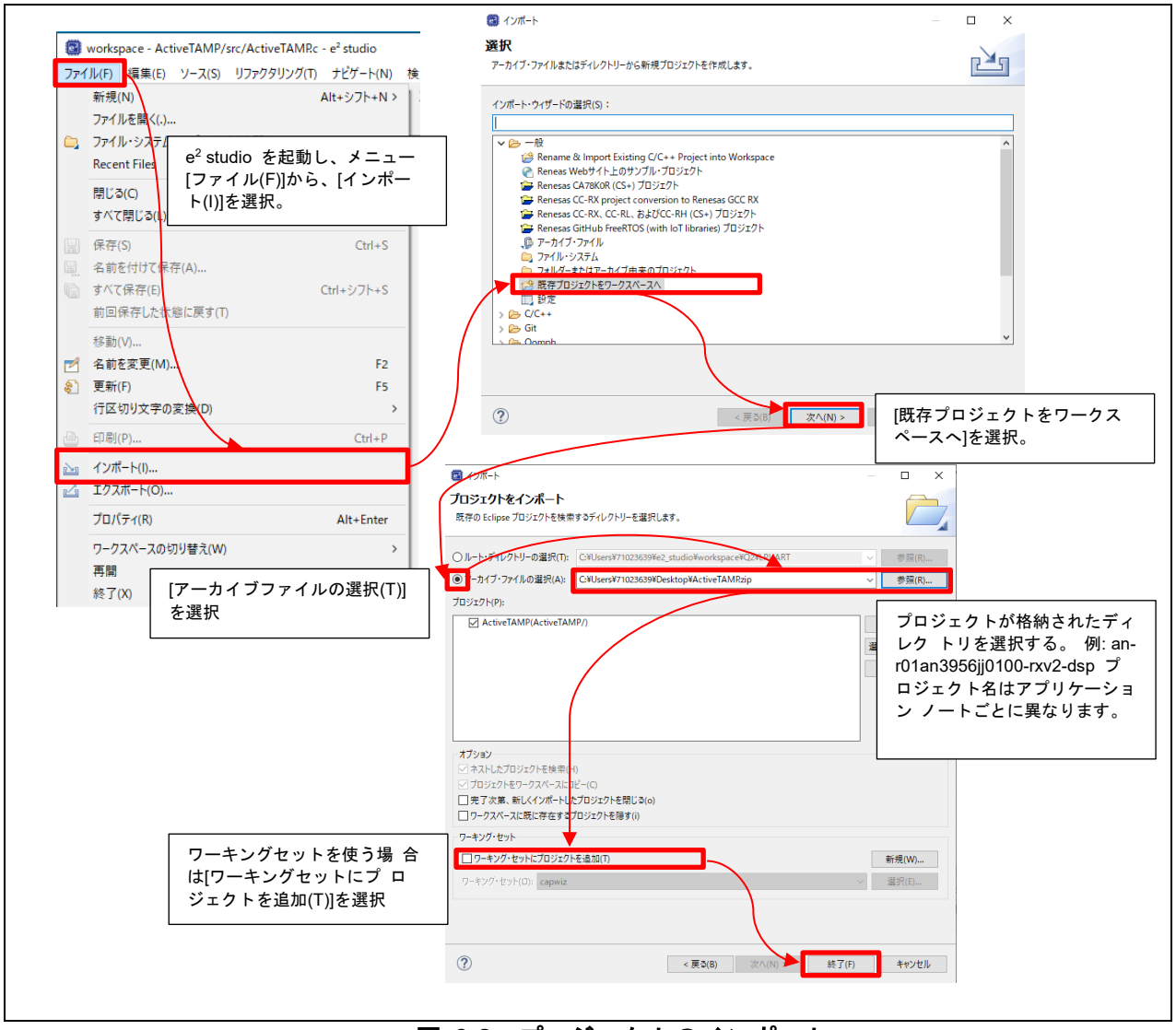


図 3-2 プロジェクトのインポート

b. プロジェクトをビルド後、e² studio のルールとしてデバッグボタンを押します。これで書き込みが開始します。

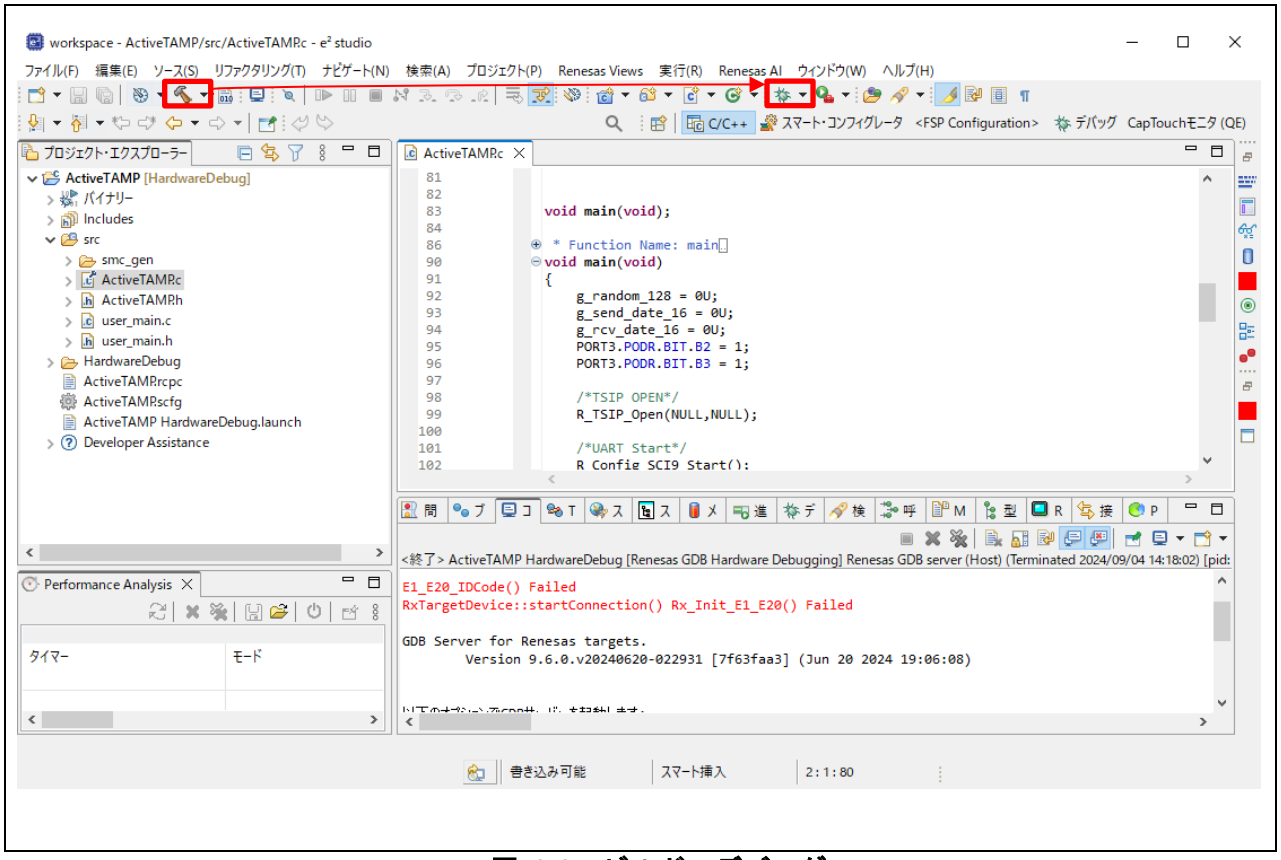


図 3-3 ビルド、デバッグ

c. b の書き込み後、自動的に FINE ブートモードで起動するため終了を押下して、e² studio とオンチップデバッギングエミュレータを切断します。

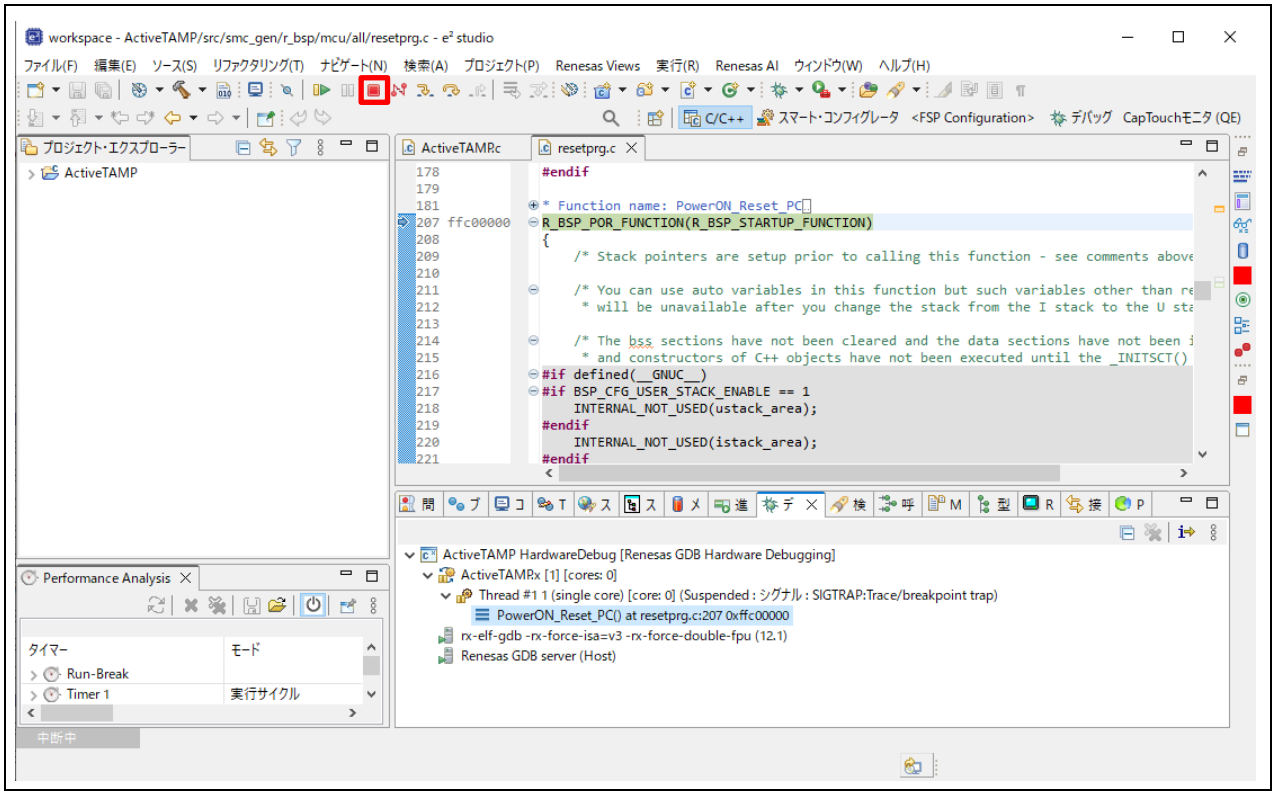
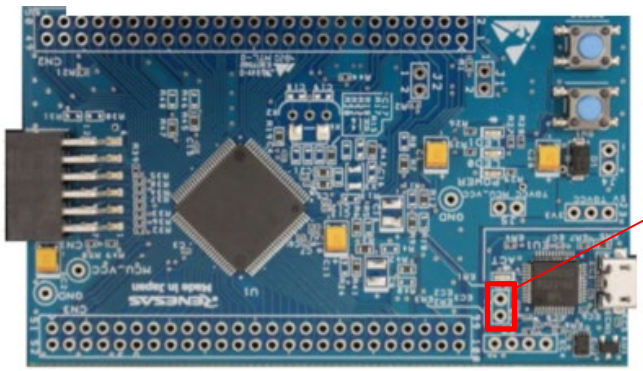


図 3-4 エミュレータ切断

d. エミュレータリセットヘッダ(J6)を接続するとデバッガを必要としないシングルチップモードとして起動し、書き込まれたプロジェクトを実行します。



Target Board for RX66N

エミュレータリセット端子(J6)を接続することにより、シングルチップモードで動作を開始します。

図 3-5 動作モード切り替え

4. システム説明

4.1 タンパ検出動作説明

図 4-1 に本サンプルプログラムのデータ(乱数)の流れ、表 4-1 にタンパ未発生時、表 4-2 に送受信データ改ざんによるタンパ発生時、表 4-3 に通信エラーによるタンパ検出時の動作詳細を示します。

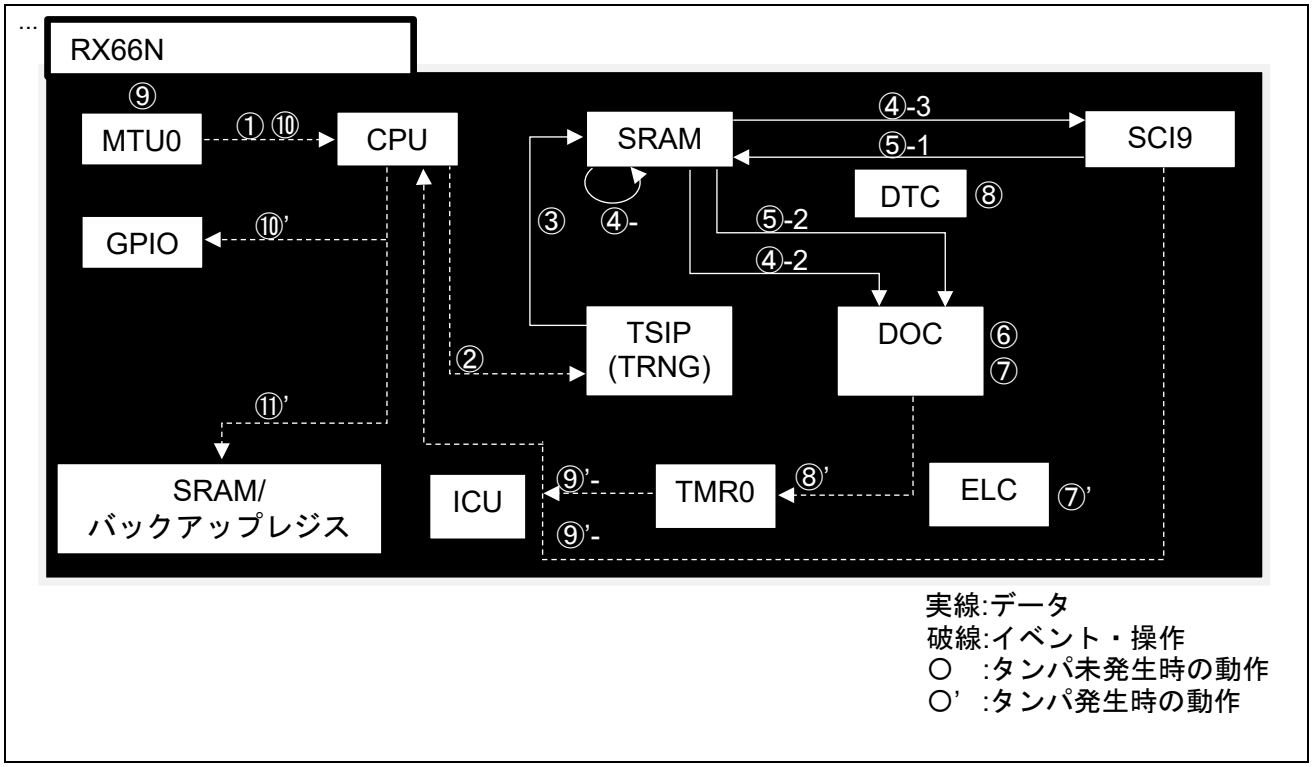


図 4-1 タンパ検出動作

表 4-1 タンパ検出動作詳細(タンパ未発生時)

動作	説明
①シーケンスの開始	MTU0.TGRA コンペアマッチ割込みにて、シーケンスを開始します
②乱数生成	TSIP 内 TRNG に乱数生成命令を行います
③乱数を格納	②によって生成された乱数を CPU によって SRAM 上に格納します
④-1DTC 転送(送信データ)	③で格納した乱数の一部を DTC のノーマル転送で、SRAM⇒SRAM へ転送します
④-2DTC 転送(送信データ)	④-1 で転送した乱数を④-1 のチェーン転送で、SRAM⇒DOC.DODSR へ転送します
④-3DTC 転送(送信データ)	④-1 で転送した乱数の一部を④-2 のチェーン転送で、SRAM⇒SCI9.TDR へ転送します
⑤-1DTC 転送(受信データ)	受信した乱数を DTC のノーマル転送で、SCI9.RDR⇒SRAM へ転送します
⑤-2DTC 転送(送信データ)	⑤-1 で転送した乱数を⑤-1 のチェーン転送で、SRAM⇒DOC.DODIR へ転送します
⑥送受信した乱数の比較	④-2 で転送した乱数と、⑤-2 で転送した乱数を DOC によって比較します
⑦一致時	次の乱数の転送を開始します
⑧繰り返し	④～⑦-1 を計 4 回繰り返します
⑨アイドル時間	アイドル時間分待機します
⑩シーケンスの終了	MTU0.TGRA コンペアマッチ割り込みで 1 シーケンスを終了します (①に戻る)

表 4-2 タンパ検出動作詳細(送受信データ改ざんによるタンパ検出時)

動作	説明
①～⑥タンパ未発生時と同じ	-
⑦'不一致時	⑥の比較結果が不一致であった場合、イベントリンク信号を出力し、次の乱数の転送を開始します。
⑧'不一致回数のカウント	⑦'のイベント信号により、TMR0.TCNT(不一致回数)をインクリメントします。
⑨'-1 ２回不一致時、 TMR0 割り込み発生	TMR0.TCNT の値が 2 となる時、TMR0 は CPU に対し TCORA コンペアマッチ割り込みを上げます。
⑩'LED1 の点灯	⑨-2 の割り込みが発生時、CPU は LED1 を点灯します。
⑪'機密情報の削除	⑩の後、CPU は SRAM 内の機密情報ならび、バックアップレジスタの内容を削除(0 クリア)し、エラー時のユーザプログラムをコールします。

表 4-3 タンパ検出動作詳細(通信エラーによるタンパ検出時)

動作	説明
①～⑥タンパ未発生時と同じ	-
⑨'-2 通信エラー 割り込み発生	通信エラー検出時、SCI9 は CPU に対し通信エラー割り込みを上げます
⑩'LED0 の点灯	⑨-1 の割り込みが発生時、CPU は LED0 を点灯します。
⑪'機密情報の削除	⑩の後、CPU は SRAM 内の機密情報ならび、バックアップレジスタの内容を削除(0 クリア)し、エラー時のユーザプログラムをコールします。

4.1.1 DTC の転送動作

本サンプルプログラムでは、ソフトウェア要因、送信データエンプティ要因、受信データフル要因により DTC 転送を行います。

以降、起動要因ごとの DTC を (ソフト)DTC、(送信)DTC、(受信)DTC と示します。

表 4-4 に起動要因ごとの DTC の動作概要を示します。表中の番号は図 4-1 内の番号とリンクしています。

また、転送する乱数の詳細は 4.1.2 **乱数の bit 長整合**を参照ください

表 4-4 起動要因と DTC 転送動作

動作概要	名称	起動要因
④-1 128bit 乱数の内 8bit を SRAM⇒SRAM へ転送します	(ソフト)DTC	ソフトウェア
④-2 上記のチェーン転送にて上記で転送した乱数を SRAM⇒DOC へ転送します	(ソフト)DTC	チェーン転送につき、なし
④-3 上記のチェーン転送にて SRAM⇒SCI9.TDR へ転送します。	(ソフト)DTC	同上
④-3 SRAM⇒SCI9.TDR へ転送を行います。	(送信)DTC	送信データエンプティ
⑤-1 SCI.RDR 内の乱数を SRAM へ転送します。	(受信)DTC	受信データフル
⑤-2 上記のチェーン転送で、SRAM⇒DODIR へ転送します。	(受信)DTC	チェーン転送につき、なし

4.1.2 乱数の bit 長整合

TRNG で生成される乱数は 128bit 長、SCI で転送できる乱数は 8bit 長、DOC で比較できる乱数は 16bit 長と同一でないため、乱数のやり取りには bit 長を整合する必要があります。

本サンプルプログラムでは、DOC で比較可能な 16bit 長に整合します。

図 4-2 に乱数の bit 長の整合方法の概要、表 4-5 に乱数 bit 長の整合方法の詳細を示します。

図内の番号は図 4-1 内の番号とリンクしています。

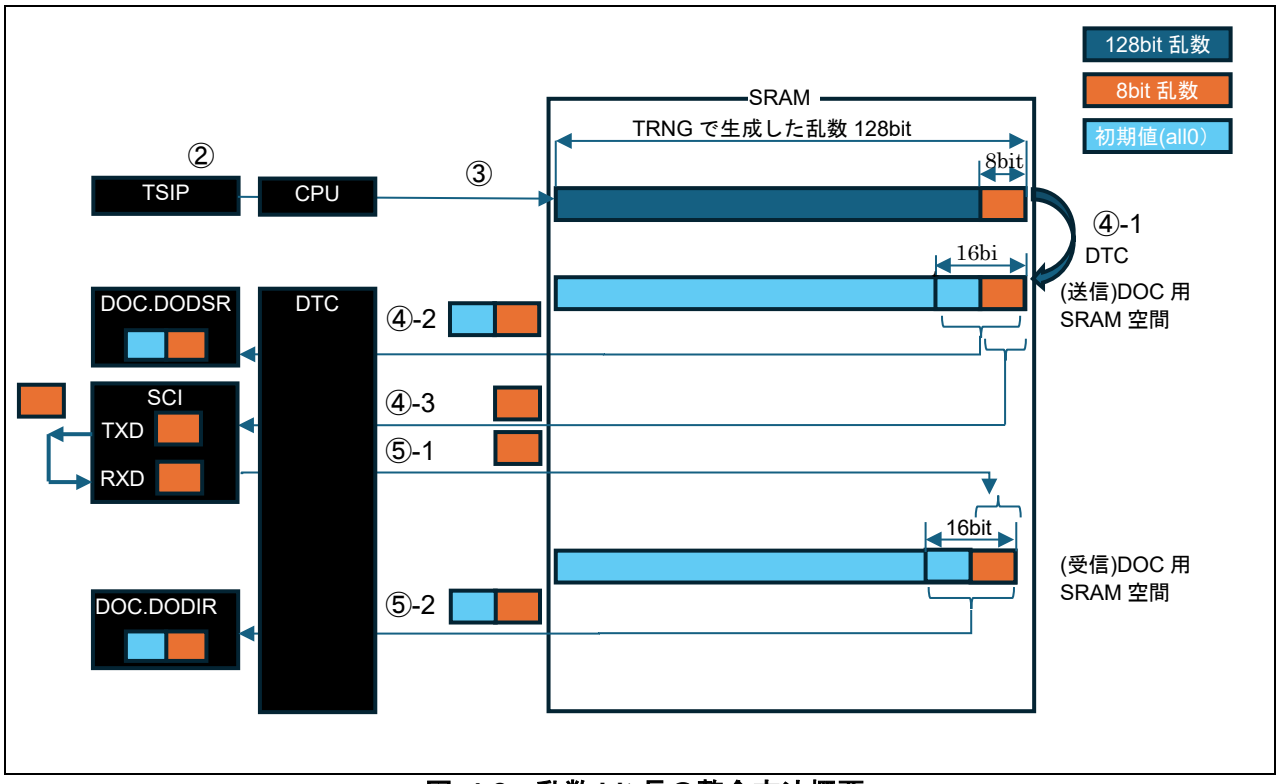


図 4-2 乱数 bit 長の整合方法概要

表 4-5 乱数 bit 長の整合方法詳細

動作	説明
②乱数生成	TSIP 内の TRNG により乱数(128bit 長)を生成します。
③乱数格納	②で生成した 128bit 長の乱数を CPU により SRAM へ格納します。
④-1 乱数の 16bit 化	③で格納された 128bit 長の乱数の最下位 8bit を(送信)DOC 用 SRAM 空間の下位へ転送し、初期値と合わせて 16bit 長の乱数に整合します。
④-2 送信乱数を DOC へ格納	④-1 で整合した 16bit 長の乱数を DTC を介して DOC.DODSR へ転送します。
④-3 送信乱数の転送	④-1 で作成した 16bit 長の乱数の下位 8bit を DTC を介して SCI9.TDR に転送します。
⑤-1 受信乱数の転送	受信した 8bit 長の乱数を(受信)DOC 用 SRAM 空間の下位へ転送し初期値と合わせて 16bit 長に整合します。
⑤-2 受信乱数を DOC へ格納	⑤-1 で整合した 16bit 長の乱数を DTC を介して DOC.DODIR へ転送します。

4.1.3 タンパ検出時の消去

タンパ検出により、SRAM の 0005 0000h～0005 00F0h とバックアップレジスタ(0008 C2A0h~0008 C2BFh)にある機密情報を消去します。

消去後、本サンプルプログラムは無限ループに入ります。なお、ユーザは、通信エラー（オーバラン、フレーミング、パリティエラー）割り込みもしくは、TMR0.TCORA コンペアマッチ割り込みでタンパ検出により、機密情報が消去されたことを知ることができます。

4.2 1シーケンスの動作詳細

本サンプルプログラムはユーザプログラムへの負荷を最低限となるよう極力 CPU 処理を少なくしています。これにより、バックグラウンドでタンパ検出が可能です。本サンプルプログラムの電源投入から 1 シーケンス終了までの動作を以下に示します。

1. 各周辺機能(TSIP、SCI9、TMR0、ELC、DOC、MTU0、DTC)を初期化します。
2. MTU0.TGRA の値を MTU0.TCNT に設定後、各周辺機能(TSIP、SCI9、TMR0、ELC、DOC、MTU0)の順に起動します。
3. MTU0.TSTRA.CST0 ビットに 1b を設定すると、MTU0 がカウントを開始します。
4. MTU0.TCNT レジスタの値が MTU0.TGRA レジスタの値と一致すると MTU0.TCNT レジスタの値がクリアされます。同時に MTU0.TGRA コンペアマッチ割り込みが発生します。
5. MTU0.TGRA コンペアマッチ割り込み処理で乱数生成、DTC の転送回数の確認/初期化/転送許可を行います。その後 SWINTR.SWINT に 1b を設定しソフトウェア要因の DTC 転送をスタートします。
6. ソフトウェア要因の DTC 転送により、SRAM(A)*から SRAM(B)*へ転送を行い、チェーン転送にて、SRAM(B)*を DOC.DODIR へ転送します。さらにチェーン転送で SRAM(B)*の一部を SCI9.TDR へ転送します。これにより送信データエンプティ要因の DTC 転送が開始され、4 回の SCI9 送信がスタートします。
7. SCI9 送信データの自己折り返しによる SCI9 の受信データにより、受信データフル要因の DTC 転送がスタートし、受信データ(SCI9.RDR)を SRAM(C)*へ転送します。
8. (7)のチェーン転送にて受信データを DOC.DODSR へ転送し送受信データを比較します。

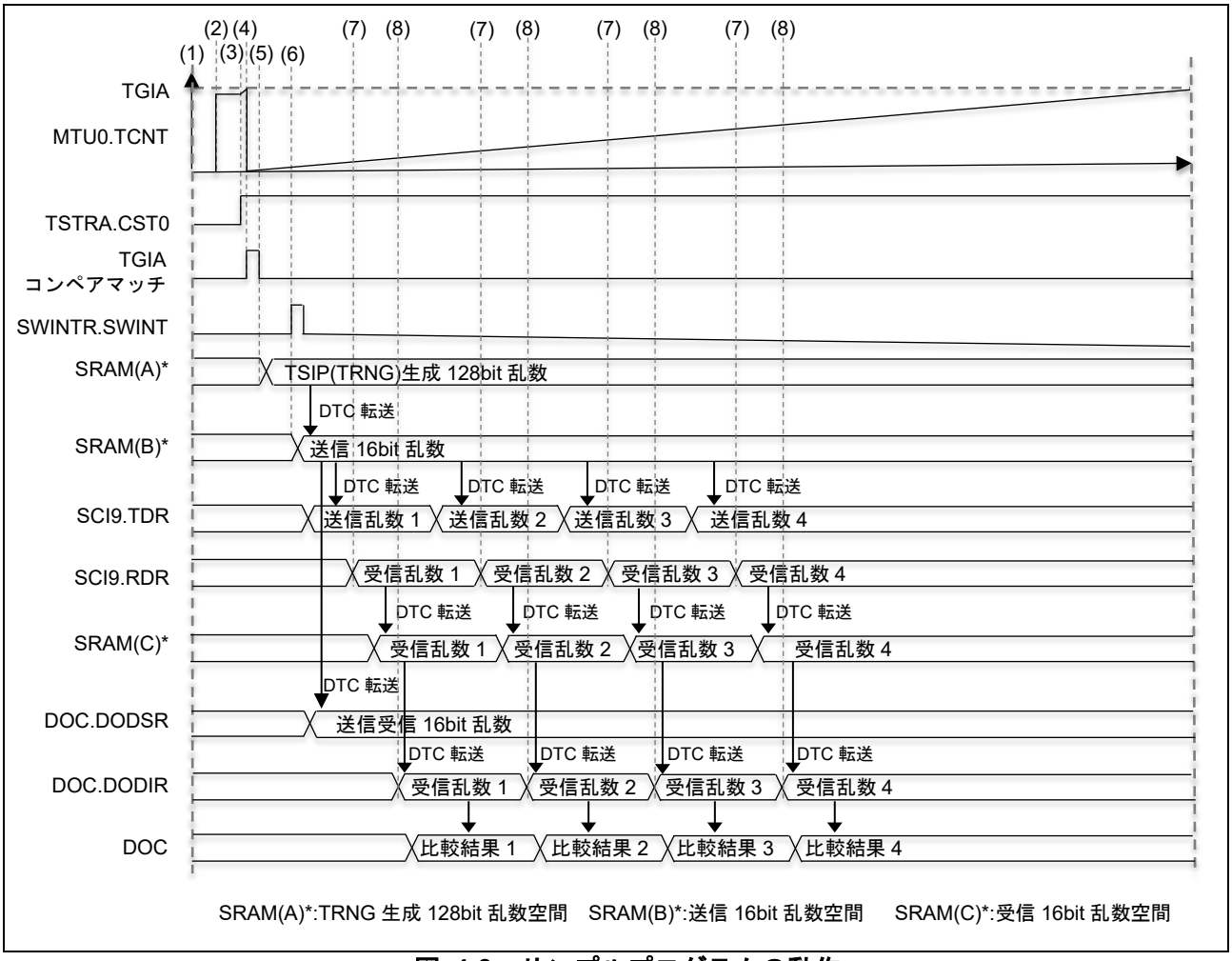


図 4-3 サンプルプログラムの動作

4.2.1 電源投入から 1 シーケンス目の転送開始時の挙動

本サンプルプログラムでは、電源投入後のタンパ検出ができない時間を極力抑えるため、MTU0 のソースクロックで 1 クロック目の立ち上がりでタンパ検出を開始します。

1 シーケンス目後は MTU0.TGRA に設定した周期でシーケンスを開始します。

4.2.2 タンパ発生時の動作詳細

シーケンスの動作中にタンパが発生した場合、即座に以下の動作を行います。

1. 発生したタンパの原因(乱数改ざん、通信エラー)に対応する LED を点灯します。
2. タンパ検出に使用している周辺機能をすべて停止します。
3. RAM とバックアップレジスタ内の機密情報をすべて削除(0 クリア)します。
4. ユーザのエラー時の処理関数をコールします。

5. ソフトウェア説明

5.1 構成

5.1.1 ファイルの構成

表 5-1 に本サンプルプログラムで使用するファイルを示します。なお、スマート・コンフィグレータのコード生成機能によって生成されたソースコードをそのまま使用しているファイルは除きます。

表 5-1 サンプルコードで使用するファイル

ファイル名	概要	備考
ActiveTAMP.c	メイン処理	本サンプルコードでは、タンパ検出処理の初期設定、スタート処理を行う。
ActiveTAMP.h	ActiveTAMP .c のヘッダファイル	-
user_main.c	ユーザのメイン、エラー時の処理	ActiveTAMP.c から呼び出す。 本サンプルプログラム内では何も処理を行わない。
user_main.h	user_main.c のヘッダファイル	-
Config_MTU0_user.c	タイマ割り込み処理	-
Config_DTC_soft.c	DTC 転送設定処理	ソフトウェア要因
Config_DTC_rcv_user.c	DTC 転送設定処理	ソフトウェア要因
Config_DTC_send.c	DTC 転送設定処理	送信データエンプティ要因
Config_DTC_send_user.c	DTC 転送設定処理	送信データエンプティ要因
Config_DTC_rcv.c	DTC 転送設定処理	受信データフル要因
Config_DTC_soft_user.c	DTC 転送設定処理	受信データフル要因

5.1.2 変数一覧

表 5-2 にグローバル変数を示します。

表 5-2 変数一覧

型	変数名	内容	使用関数
uint32_t	g_random_128	128bit の乱数を格納する SRAM 領域	main r_Config_MTU0_tgia0_interrupt
uint16_t	g_send_data_16	16bit 長の送信乱数を格納する SRAM 領域	main r_Config_MTU0_tgia0_interrupt
uint16_t	g_rcv_data_16	16bit 長の受信乱数を格納	main
uint8_t	g_secret_data[240]	機密情報を模したデータを格納する SRAM 領域	main
uint8_t	g_backup[32]	機密情報を模したデータを格納するバックアップレジスタ領域	main

5.1.3 関数一覧

表 5-3 に関数一覧を示します。なお、スマート・コンフィグレータのコード生成機能によって生成されたソースコードをそのまま使用している関数については省略しています。

表 5-3 関数一覧

関数名	概要
main	メイン処理、タンパ検出動作の初期設定、開始処理
erase_secret_data	機密情報の削除処理
user_main_function	ユーザのメイン処理
user_err_function	ユーザのエラー処理
r_Config_MTU0_tgia0_interrupt	タンパ検出シーケンスの開始処理
R_Config_DTC_rcv_Remaining	受信データフル要因の DTC 残り転送回数の取得処理
R_Config_DTC_rcv_reset	受信データフル要因の DTC 転送回数の再設定処理
R_Config_DTC_rcv_Create_UserInit	受信データフル要因の DTC 転送回数の初期設定処理
R_Config_DTC_send_reset	送信データエンプティ要因の DTC 転送回数の再設定処理
R_Config_DTC_send_Create_UserInit	送信データエンプティ要因の DTC 転送回数の初期設定処理
R_Config_DTC_soft_reset	ソフトウェア要因の DTC 転送回数の再設定処理
R_Config_DTC_soft_Create_UserInit	ソフトウェア要因の DTC 転送回数の初期設定処理

5.1.4 関数仕様

サンプルコードの関数仕様を示します。

main	
概　要	メイン処理、タンパ検出動作の初期設定、開始処理
ヘッダ	なし
宣　言	void main(void)
説　明	初期設定後、タンパ検出動作の開始、ユーザメイン処理のコールを行います。
引　数	なし
リターン値	なし
備　考	-

erase_secret_data	
概　要	機密情報の削除処理
ヘッダ	ActiveTAMP.h
宣　言	void erase_secret_data(void)
説　明	SRAM 内やバックアップレジスタ内の機密情報を削除(0 クリア)し、 ユーザのエラー処理をコールします。
引　数	なし
リターン値	なし
備　考	処理後は無限ループに入り、他の処理を行いません。

user_main_function	
概　要	ユーザのメイン処理
ヘッダ	user_main.h
宣　言	void user_main_function(void)
説　明	ユーザの定義したメイン処理を書き込むことができます。 本処理はタンパ検出と並行して動作します。
引　数	なし
リターン値	なし
備　考	-

user_err_function	
概　要	ユーザのエラー処理
ヘッダ	user_main.h
宣　言	void user_err_function(void)
説　明	タンパ発生時のユーザの定義したエラー処理を書き込むことができます。 本処理は機密情報の削除後に 1 度だけコールされます。
引　数	なし
リターン値	なし
備　考	erase_secret_data 関数によりコールされます。

r_Config_MTU0_tgia0_interrupt	
概　要	タイマ割り込み処理
ヘッダ	r_Config_MTU0.h
宣　言	static void r_Config_MTU0_tgia0_interrupt(void)
説　明	MTU0.TGIA のコンペアマッチ割り込み処理により、タンパ検出のシーケンスの開始処理を行います。
引　数	なし
リターン値	なし
備　考	-

R_Config_DTC_rcv_Remaining	
概　要	受信データフル要因の DTC 残り転送回数の取得処理
ヘッダ	R_Config_DTC_rcv.h
宣　言	int R_Config_DTC_rcv_Remaining(void)
説　明	SCI9 の受信データフル要因の残り転送回数をリターンします。
引　数	なし
リターン値	int dtc_transferdata_vector102[0].cra_crb
備　考	-

R_Config_DTC_rcv_reset	
概　要	受信データフル要因の DTC 残り転送回数の再設定処理
ヘッダ	R_Config_DTC_rcv.h
宣　言	void R_Config_DTC_rcv_reset(void)
説　明	受信データフル要因の DTC 残り転送回数を 4 回に再設定を行います。
引　数	なし
リターン値	なし
備　考	-

R_Config_DTC_rcv_Create_UserInit	
概　要	受信データフル要因の DTC 転送回数の初期設定処理
ヘッダ	R_Config_DTC_rcv.h
宣　言	void R_Config_DTC_rcv_Create_UserInit(void)
説　明	初期化時、受信データフル要因の DTC 残り転送回数を 0 に初期化します
引　数	なし
リターン値	なし
備　考	-

R_Config_DTC_send_reset	
概　要	送信データエンプティ要因の DTC 転送回数の再設定処理
ヘッダ	R_Config_DTC_send.h
宣　言	void R_Config_DTC_send_reset(void)
説　明	送信データエンプティ要因の DTC 残り転送回数を 3 回に再設定を行います。
引　数	なし
リターン値	なし
備　考	-

R_Config_DTC_send_Create_UserInit	
概　要	送信データエンプティ要因の DTC 転送回数の初期設定処理
ヘッダ	R_Config_DTC_send.h
宣　言	void R_Config_DTC_send_Create_UserInit(void)
説　明	初期化時、送信データエンプティ要因の DTC 残り転送回数を 0 に初期化します
引　数	なし
リターン値	なし
備　考	-

R_Config_DTC_soft_reset	
概　要	ソフトウェア要因の DTC 転送回数の再設定処理
ヘッダ	R_Config_DTC_soft.h
宣　言	void R_Config_DTC_soft_reset(void)
説　明	ソフトウェア要因の DTC 残り転送回数を 1 回に再設定を行います
引　数	なし
リターン値	なし
備　考	-

R_Config_DTC_soft_Create_UserInit	
概　要	ソフトウェア要因の DTC 転送回数の初期化処理
ヘッダ	R_Config_DTC_soft.h
宣　言	void R_Config_DTC_soft_Create_UserInit(void)
説　明	初期化時、ソフトウェア要因の DTC 残り転送回数を 0 に初期化します
引　数	なし
リターン値	なし
備　考	-

5.2 フローチャート

以降フロー内で示される「各モジュールの初期化」はスマート・コンフィグレータにより、R_Config_<モジュール名称>_Create 関数に生成されます。なお、TSIP は R_TSIP_Open 関数に生成されます。

5.2.1 全体動作フロー

図 5-1 に本サンプルプログラムの全体動作フローを示します。

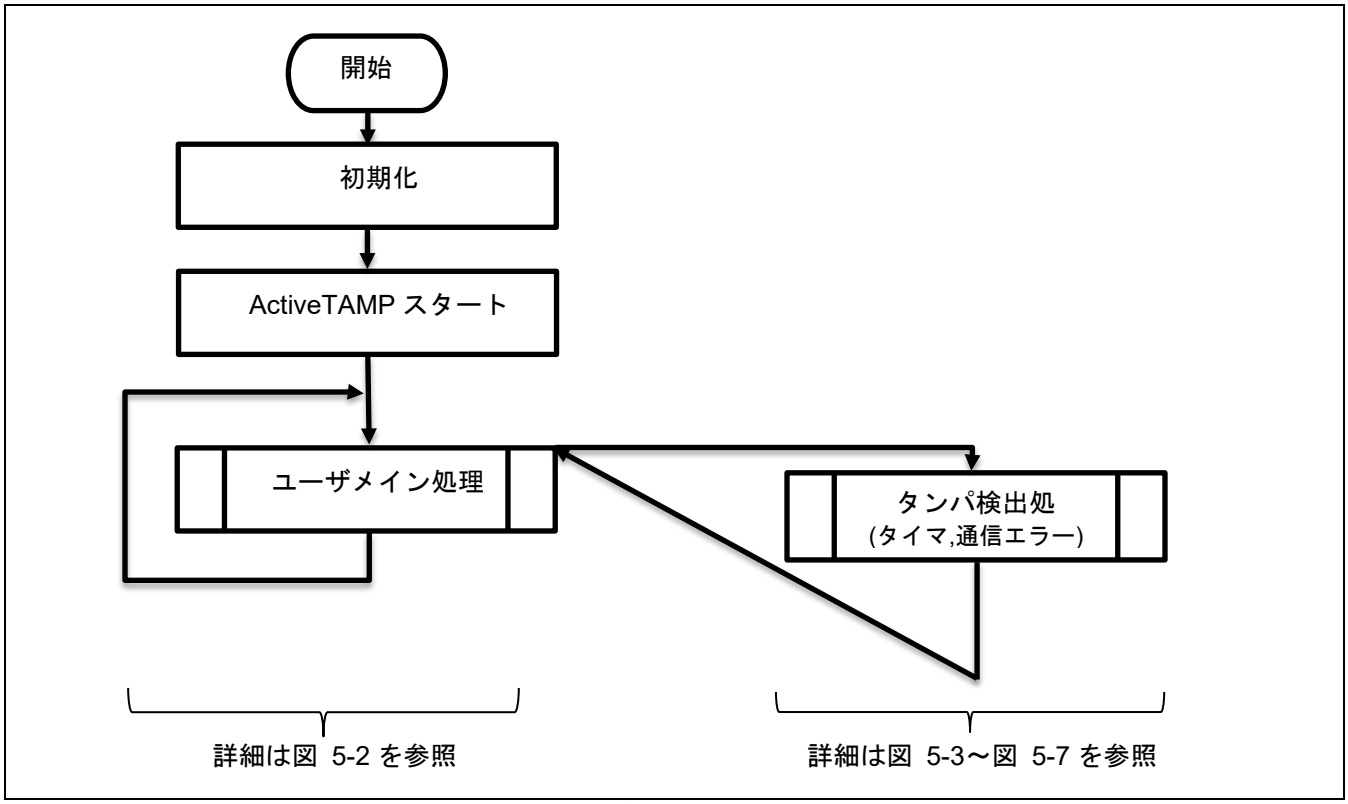


図 5-1 全体動作フロー

5.2.2 メイン関数の初期化および ActiveTAMP スタートのフロー

図 5-2 にメイン関数の初期化および ActiveTAMP スタートのフローを示します。

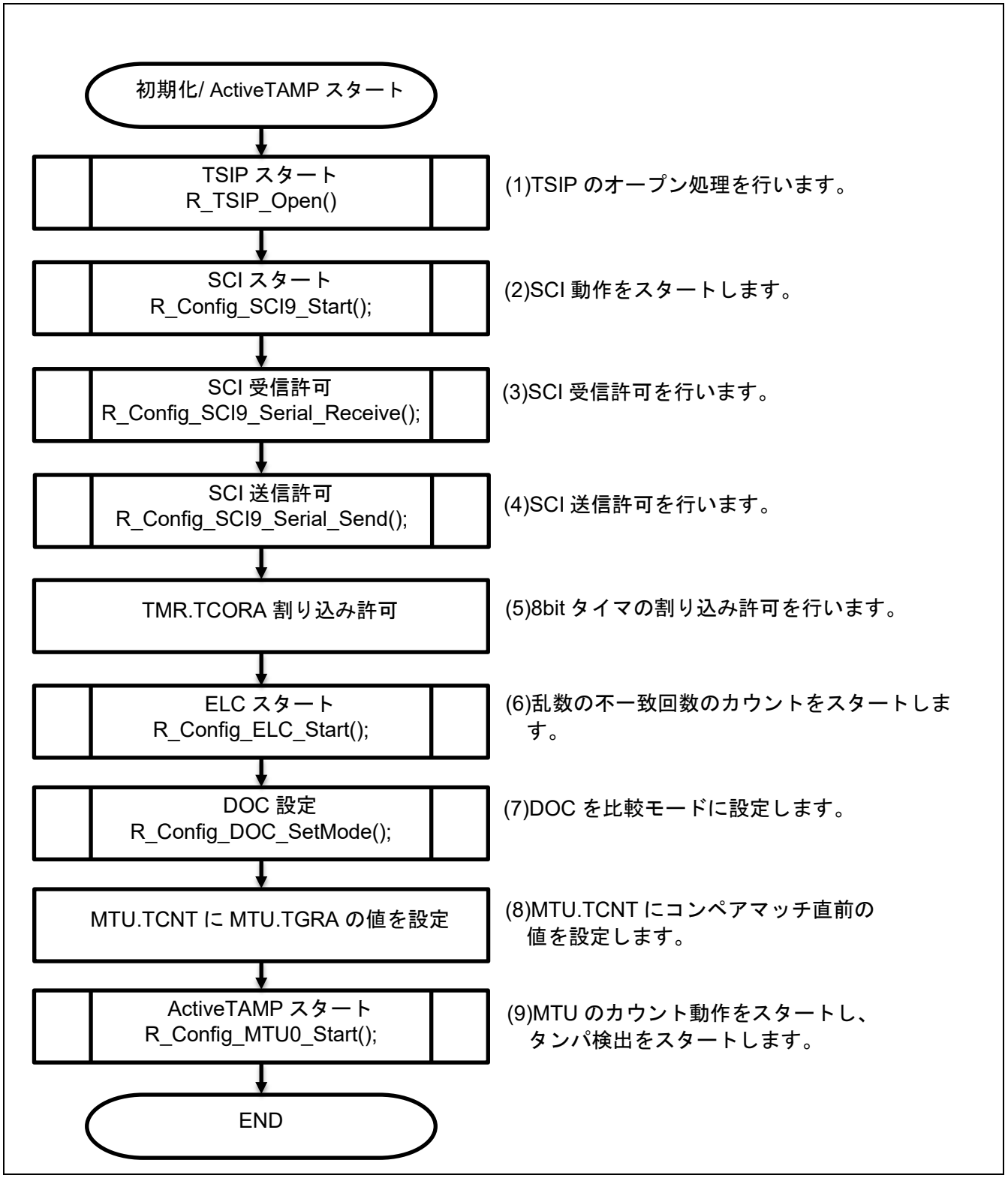


図 5-2 メイン関数の初期化および ActiveTAMP スタートのフロー

5.2.3 タンパ検出処理のフロー

図 5-3 にタイマによるタンパ検出割り込み処理の概要フロー、図 5-4 に詳細フローを示します。

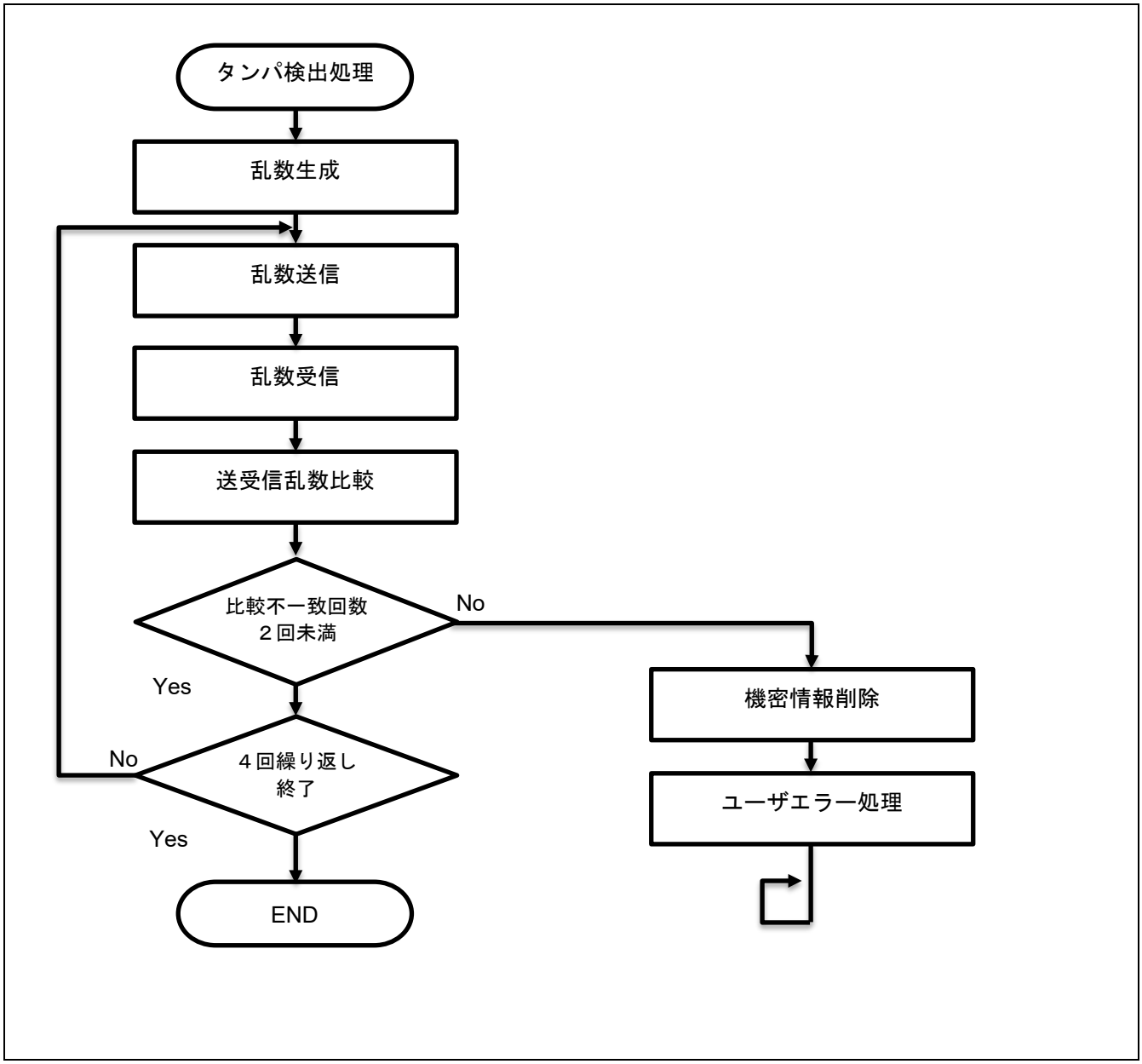


図 5-3 タイマによるタンパ検出割り込み処理 概要フロー

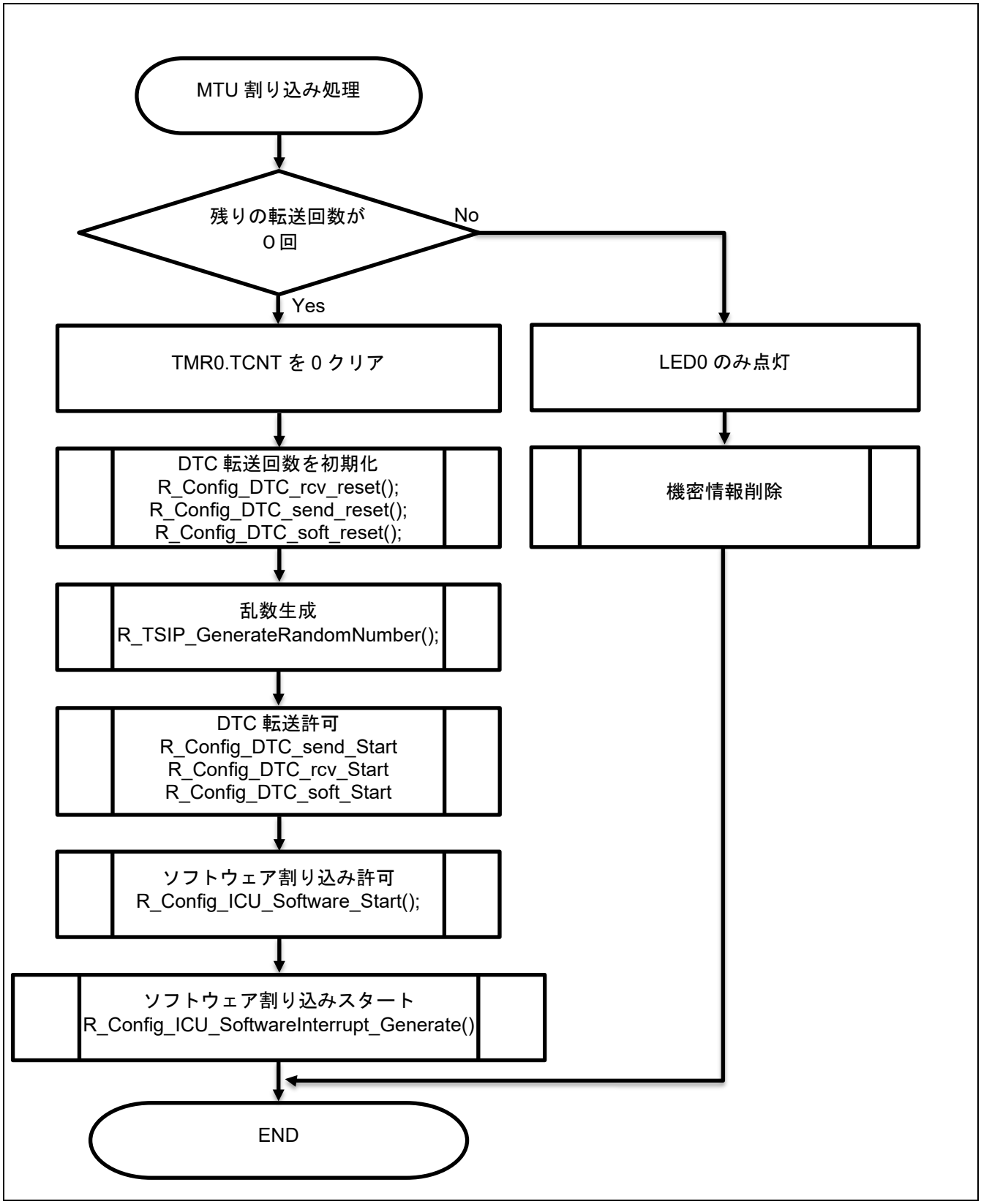


図 5-4 タイマによるタンパ検出処理の割り込み処理 詳細フロー

5.2.4 タイマによる比較不一致回数が 2 回以上時のフロー

図 5-5 にタイマによる比較不一致回数が 2 回以上時のフローを示します。

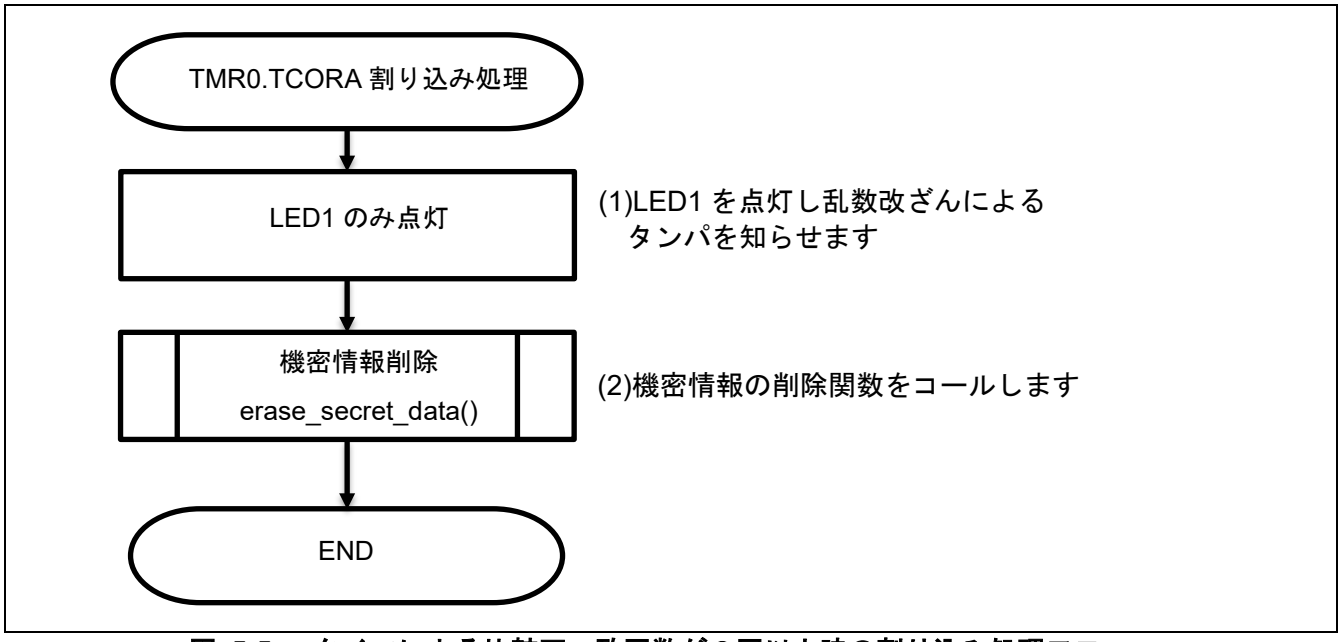


図 5-5 タイマによる比較不一致回数が 2 回以上時の割り込み処理フロー

5.2.5 通信エラー割り込み処理

図 5-6 に通信エラーによる割り込み処理を示します。

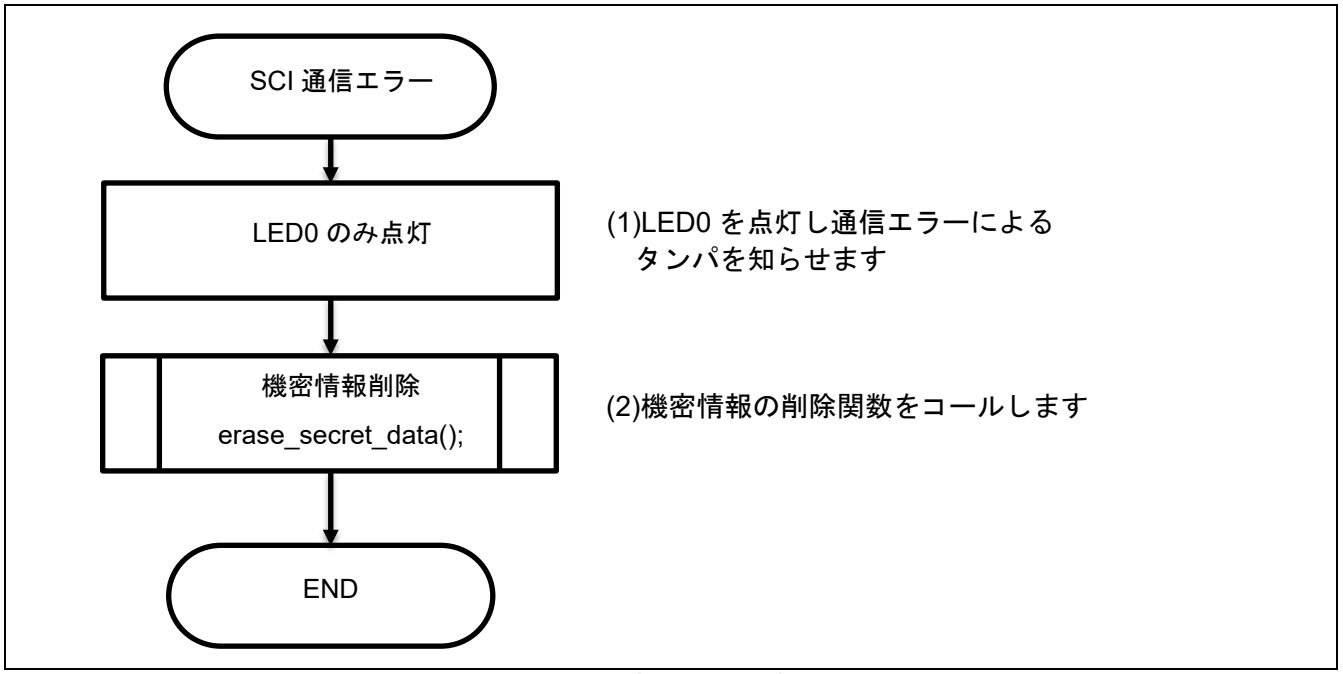


図 5-6 通信エラーによる割り込みの割り込み処理フロー

5.2.6 機密情報削除関数

図 5-7 にタンパ発生時の機密情報削除における処理を示します。

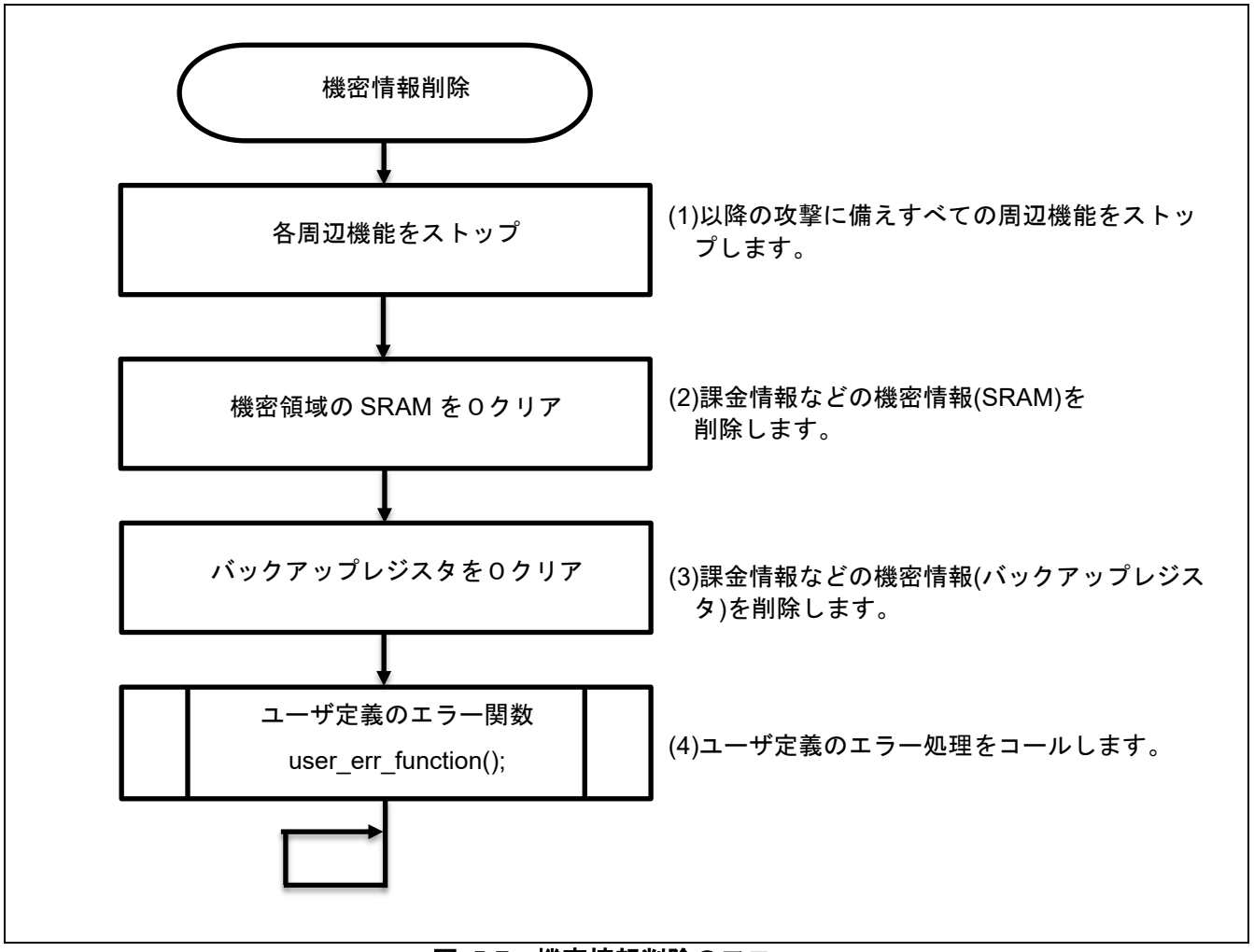


図 5-7 機密情報削除のフロー

5.3 フットプリント

表 5-4 に本サンプルプログラムのフットプリントを示します。

表 5-4 サンプルプログラムのフットプリント

プログラム	サイズ
TSIP ドライバ	245 kB
サンプルコード(TSIP を除く)	10 kB

なお、RX100,RX200 シリーズに搭載の TSIP-Lite 用ドライバのフットプリントは、最大約 55kB です。
よって、RX100,RX200 シリーズでも本サンプルプログラムの使用は可能です。

5.4 注意事項

5.4.1 タンパ検出を停止させる場合の注意事項

タンパ検出は MTU0.TGRA のコンペアマッチをトリガーにタンパ検出動作を開始しています。そのため検出動作を停止させる場合は MTU0 のカウント動作を停止させてください。

また、MTU0 を除くその他の周辺機能を停止させる場合、MTU0 が停止したのを確認後、SCI9 で送受信が行われていないことをご確認した後停止させてください。

6. 参考ドキュメント

RX66N グループ ユーザーズマニュアル ハードウェア編 (renesas.com)

Target Board for RX66N ユーザーズマニュアル Rev.1.00 (renesas.com)

RX ファミリ TSIP(Trusted Secure IP)モジュール Firmware Integration Technology Rev.1.21 (renesas.com)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2025.01.10	-	初版発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   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. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「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。
5. クロックについて
   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。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
8. 製品間の相違について
   型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、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、ノイズ幅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。

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。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、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 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。
10. 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。
11. 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。
12. 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。
13. 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。
14. 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.5.0-1　2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。