

## RH850/U2B Group

R01AN6571JJ0100

Rev.1.00

### RHSIF アプリケーションノート

## 1 はじめに

ルネサスハイスピードインターフェース（RHSIF）は、2つのデバイス間でP2Pハイスピードシリアル通信を行います。

物理層の自律的な処理と、上位の通信層でのアプリケーションデータの転送をサポートします。本アプリケーションノートでは、2つのRH850 MCU デバイスを使用したインターフェースの基本的な初期化と簡単な送受信操作を示します。必要手順は、RHSIF IP が組み込まれた他のRenesas RH850 MCU に容易に適用可能です。

### 1.1 対象デバイス

- RH850/U2Bx シリーズ

本アプリケーションにおけるソフトウェア例はRH850/U2Bx デバイス上で実行されます。

## 2 目次

|       |                                 |    |
|-------|---------------------------------|----|
| 1     | はじめに.....                       | 1  |
| 1.1   | 対象デバイス.....                     | 1  |
| 2     | 目次.....                         | 2  |
| 3     | 図.....                          | 4  |
| 4     | 表.....                          | 5  |
| 5     | RHSIF モジュール.....                | 6  |
| 5.1   | 汎用インターフェース概要.....               | 6  |
| 5.2   | モジュールブロック図.....                 | 7  |
| 5.3   | RHSIF によるデバイス相互接続.....          | 7  |
| 5.4   | メッセージフレームフォーマット.....            | 8  |
| 5.5   | データリンク層 L1.....                 | 9  |
| 5.6   | トランスポート層 L2.....                | 10 |
| 5.7   | セキュリティ/安全機能.....                | 11 |
| 5.7.1 | ID ベース認証.....                   | 11 |
| 5.7.2 | チャレンジ&レスポンス認証.....              | 14 |
| 5.7.3 | メモリアドレスウィンドウ.....               | 16 |
| 5.7.4 | 保護ガード.....                      | 17 |
| 6     | RHSIF インターフェース例 — ハードウェア設定..... | 18 |
| 7     | RHSIF インターフェース例 — ソフトウェア.....   | 20 |
| 7.1   | 認証のためのセキュリティオプションバイト設定.....     | 20 |
| 7.2   | デモソフトウェアの構成.....                | 23 |
| 7.2.1 | C ヘッダーファイル「RHSIF.h」.....        | 23 |
| 7.2.2 | C コードファイル「RHSIF.c」.....         | 23 |
| 7.2.3 | C コードファイル「main_pe0.c」.....      | 24 |
| 7.3   | ソフトウェアプロセスチャート.....             | 26 |
| 7.4   | RHSIF モジュール初期設定.....            | 27 |
| 7.4.1 | 保護ガード設定.....                    | 27 |
| 7.4.2 | メモリアクセスウィンドウ構成.....             | 29 |
| 7.4.3 | ストリーム TX/RX 構成.....             | 29 |
| 7.5   | マイコンポート構成.....                  | 29 |
| 7.6   | RHSIF L1 通信.....                | 30 |
| 7.6.1 | 概要.....                         | 30 |
| 7.6.2 | PING コマンド.....                  | 31 |
| 7.7   | RHSIF L2 通信.....                | 32 |
| 7.7.1 | 概要.....                         | 32 |
| 7.7.2 | 第 1 認証.....                     | 32 |

---

|       |                          |    |
|-------|--------------------------|----|
| 7.7.3 | 第1認証チェック .....           | 33 |
| 7.7.4 | Read32 コマンド .....        | 33 |
| 7.7.5 | Write32 コマンド .....       | 35 |
| 7.7.6 | Stream256 読み込みコマンド ..... | 36 |
| 7.7.7 | Stream256 書き込みコマンド ..... | 36 |
| 7.8   | 送受信データ .....             | 37 |
| 7.8.1 | 非ストリーム送受信データ .....       | 37 |
| 7.8.2 | ストリーム送受信データ .....        | 37 |
| 8     | 略称一覧 .....               | 38 |
| 9     | 付属デモソフトに関する注意事項 .....    | 39 |
| 10    | ホームページとサポート窓口 .....      | 40 |
| 11    | 改訂記録 .....               | 1  |
| 12    | 製品ご使用上の注意事項 .....        | 2  |

### 3 図

|   |    |
|---|----|
| 図 1 : RHSIF モジュールブロック図 [1].....                   | 7  |
| 図 2 : RHSIF デバイス間の相互接続 [1].....                   | 7  |
| 図 3 : RHSIF L1 フレーム詳細 [2].....                    | 8  |
| 図 4 : RHSIF L2 フレーム詳細 [2].....                    | 8  |
| 図 5 : データリンク層ブロック図 [1].....                       | 9  |
| 図 6 : トランスポート層簡易ブロック図.....                        | 10 |
| 図 7 : ID ベースの認証プロセス —概要— [2].....                 | 11 |
| 図 8 : ID ベースの認証プロセス —手順 1— [2].....               | 12 |
| 図 9 : ID ベースの認証プロセス —手順 2— [2].....               | 12 |
| 図 10 : ID ベースの認証プロセス —手順 3— [2].....              | 13 |
| 図 11 : ID ベースの認証プロセス —手順 4— [2].....              | 13 |
| 図 12 : チャレンジ&レスポンス認証 —手順 1— [2].....              | 14 |
| 図 13 : チャレンジ&レスポンス認証 —手順 2— [2].....              | 14 |
| 図 14 : チャレンジ&レスポンス認証 —手順 3— [2].....              | 15 |
| 図 15 : チャレンジ&レスポンス認証 —手順 4— [2].....              | 15 |
| 図 16 : チャレンジ&レスポンス認証 —手順 5— [2].....              | 16 |
| 図 17 : メモリーアドレスウィンドウ [2].....                     | 17 |
| 図 18 : PiggyBack ボードハードウェアの相互接続.....              | 19 |
| 図 19 : RFP ツールによるメモリー読み込み.....                    | 21 |
| 図 20 : RFP ツールによるメモリー書き込み.....                    | 22 |
| 図 21 : 「main_pe0.c」フローチャート.....                   | 25 |
| 図 22 : ソフトウェアプロセスチャート.....                        | 26 |
| 図 23 : 関数「RHSIF_Init」フローチャート.....                 | 28 |
| 図 24 : 関数「loc_RHSIF_portInit」フローチャート.....         | 30 |
| 図 25 : 関数「RHSIF_L1_SendPing」フローチャート.....          | 31 |
| 図 26 : 関数「RHSIF_L2_Send1stAuth」フローチャート.....       | 32 |
| 図 27 : 関数「RHSIF_L2_wait1stAuth_Check」フローチャート..... | 33 |
| 図 28 : 関数「RHSIF_L2_read32」フローチャート.....            | 34 |
| 図 29 : 関数「RHSIF_L2_write32」フローチャート.....           | 35 |
| 図 30 : 関数「RHSIF_L2_stream256_read」フローチャート.....    | 36 |
| 図 31 : 関数「RHSIF_L2_stream256_read」フローチャート.....    | 36 |

## 4 表

表 1 : 略称一覧 ..... 38

## 5 RHSIF モジュール

### 5.1 汎用インターフェース概要

RHSIFは全二重通信インターフェースです。2つのデバイス間の直接通信インターフェースとして使用されます。本アプリケーションにおいて2つのデバイスは、「通信パートナー」または「パートナー」と名づけます。

物理レベルにおいてRHSIFは5つの信号を使用：

- 参照クロック：RHSIF0\_REFCLK
- 受信データ差動入力：RHSIF0\_RXDP  
RHSIF0\_RXDN
- 送信データ差動出力：RHSIF\_TXDP  
RHSIF\_TXDN

参照クロックは、ポート供給電圧（例：3.3Vまたは5V）の電圧フルスイングによる10/20MHzの信号です。差動信号は、(typ.) ±100mVの電圧スイングによる(typ.) 1.2Vのオフセット電圧の際に実行されます。バス速度は、最大320MHzです（デバイス仕様に依存）。

参照クロックは、インターフェースのバスクロック／ボーレートの生成に使用されます。専用PLLが前記のため可能です。参照クロックは、2つの通信パターンの1つによって出力され、もう1つの通信パートナーによって受信されます。出力するパートナーをマスターデバイスと呼び、受信するパートナーをスレーブデバイスと呼びます。この呼称は参照クロックの構成においてのみ重要です。

以降の通信には、両デバイスを同等のパートナーとして扱うことが可能です。したがって、イニシエーターノードとターゲットノードが使用されます。

全二重通信モードのインターフェース機能は、データの送信（TXD信号を使用）と受信（RXD信号を使用）を同時に行うことを可能とします。

非同期インターフェース機能は、データの送信と受信をそれぞれ完全に独立させて行います。 [1]



## 5.4 メッセージフレームフォーマット

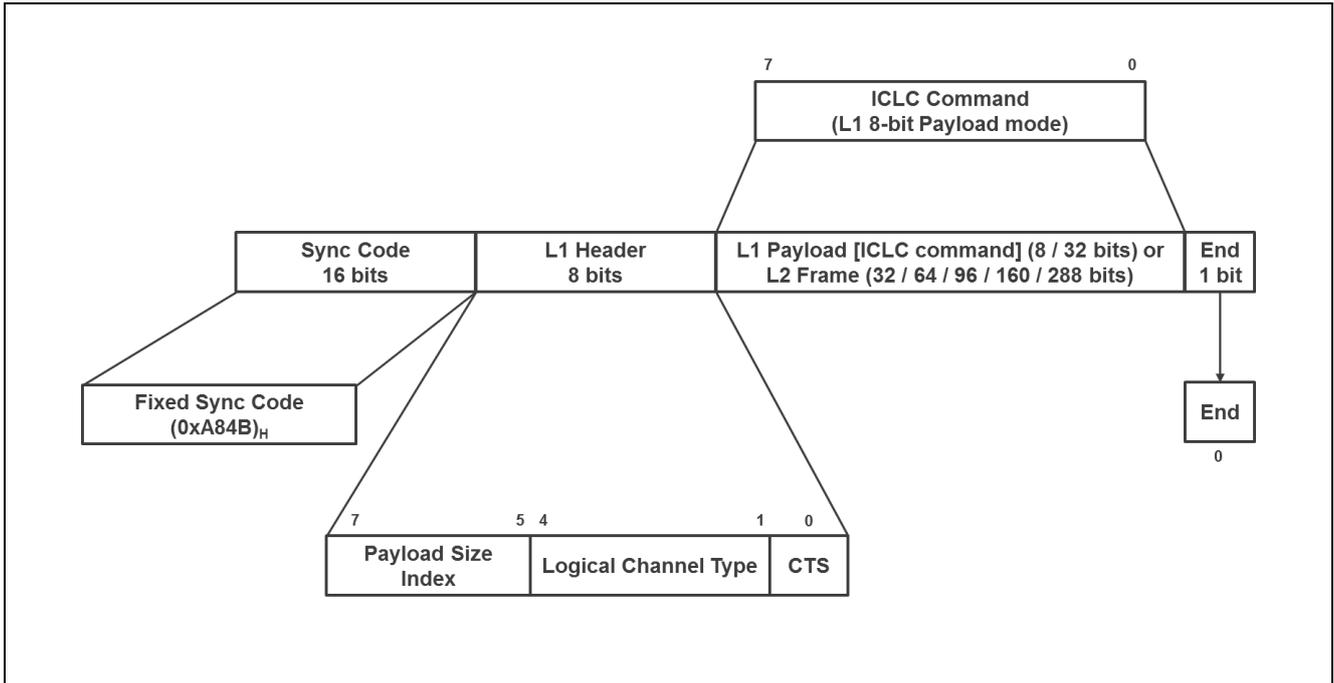


図 3 : RHSIF L1 フレーム詳細 [2]

RHSIF モジュールは 2 つのサブモジュールで構成されます。「トランスポート層 L2」と「データリンク層 L1」です。最後に送信された L1 フレームは L2 モジュールの情報に移行可能です。データリンク層は、L1 ペイロードを L2 メッセージフレームに置き換えることにより、L2 データを最終メッセージフレームに統合します。最終メッセージフレームの詳細な構成は前図に示されています。

L2 フレームフォーマットはヘッダーと CRC 情報で構成されます。ペイロードサイズは 0、32、64、128、256 ビットです。ヘッダー内で、トランザクション ID、コマンド ID、およびチャネル番号が送信されます。図 4 に L2 フレームの詳細な構成を示します。

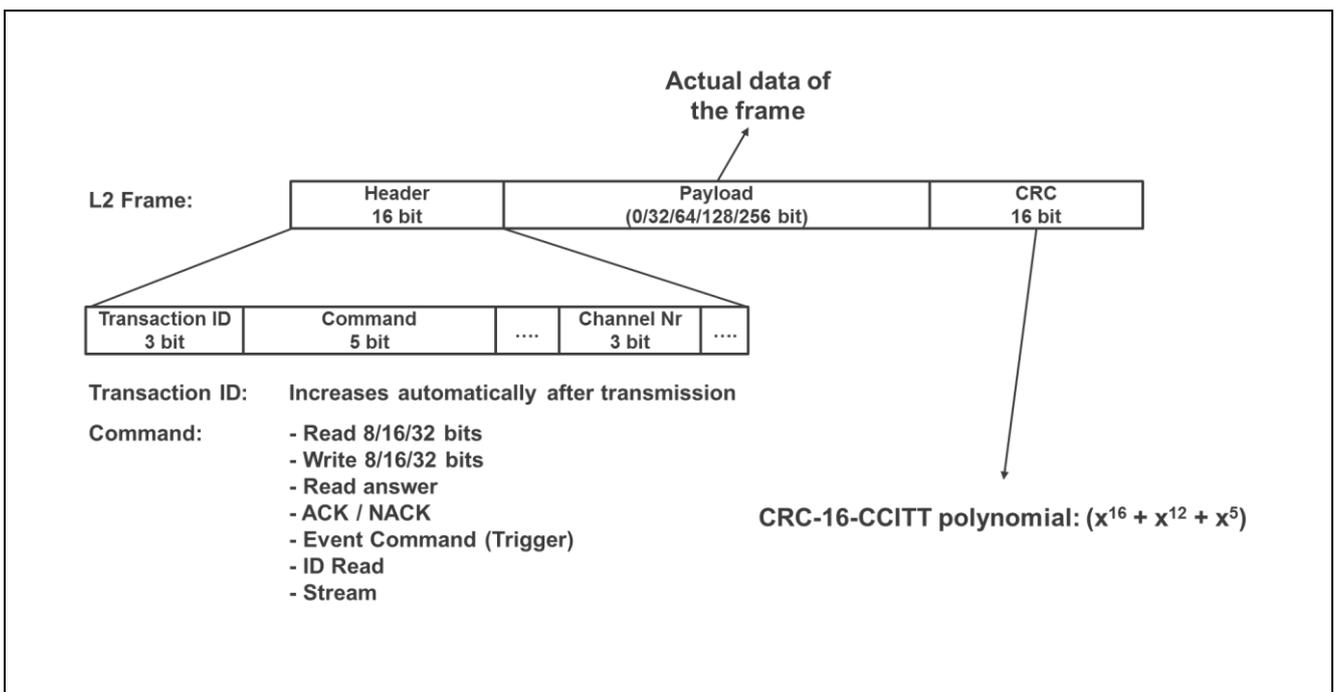


図 4 : RHSIF L2 フレーム詳細 [2]

次の2つの章では、ユーザーが両サブモジュールの相互作用の基本を理解できるよう、両サブモジュールの概要を説明します。詳細な説明については、アプリケーションで使用される RH850 MCU デバイスに適合する最新のユーザーマニュアルの RHSIF の章を参照ください。

## 5.5 データリンク層 L1

データリンク層は、RHSIF マクロを物理インターフェイスに結合し、L1 フレーム情報を最後に送信される L1 メッセージフレームに統合します。以下の機能は本サブモジュールに実装されています。

- ICLC コマンドの生成/受信 (例: READ、WRITE または PING)
- CTS コマンドの生成/送信 (「Clear To Send」コマンド)
- 割り込み発生
- フレームアービトレーション
- フローコントロール
- 手順モード:
  - ループバックテスト
  - パターンテスト

図 5 に L1 サブモジュールのブロック図を示します。

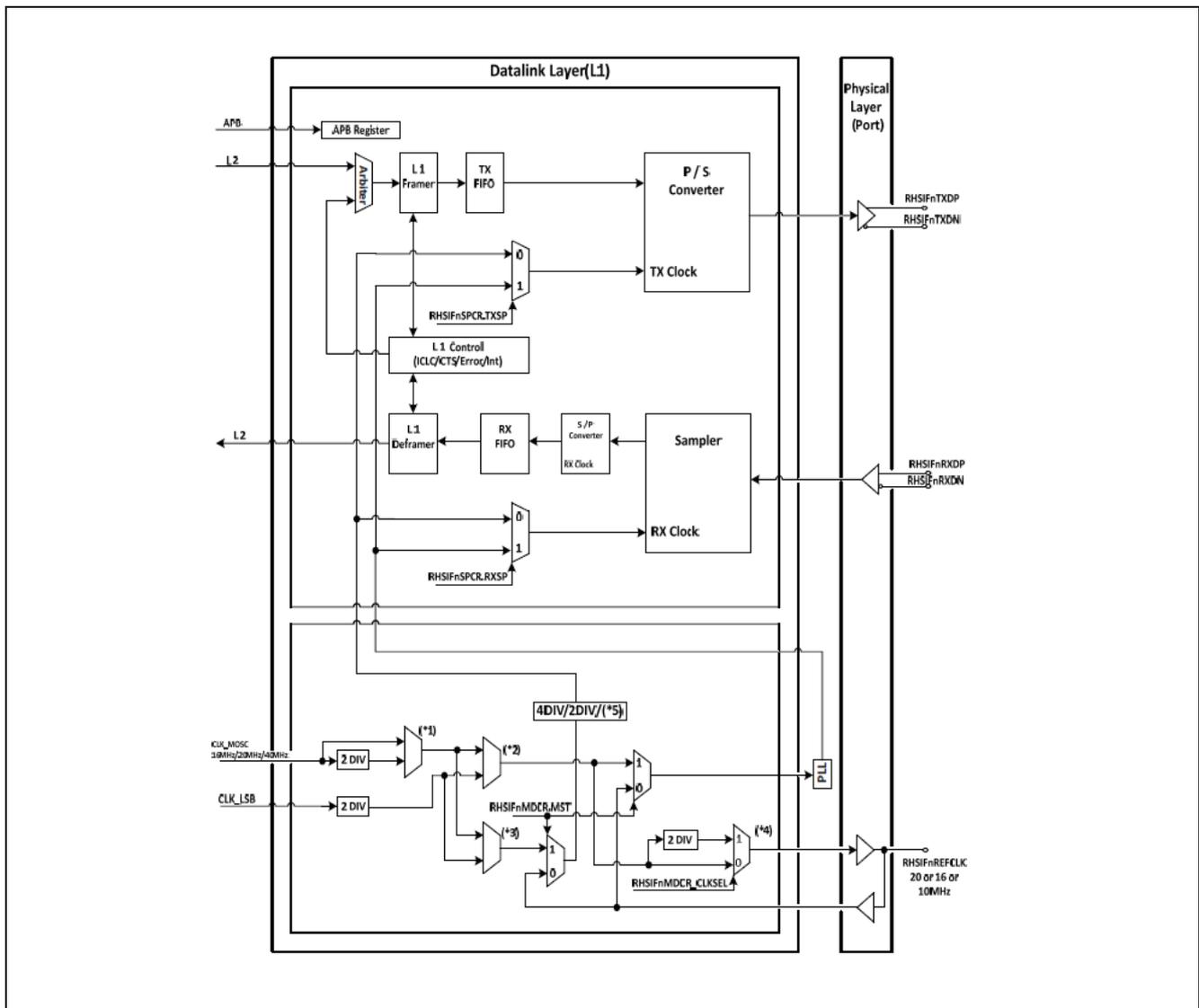


図 5 : データリンク層ブロック図 [1]

## 5.6 トランスポート層 L2

トランスポート層 L2 は、L2 メッセージフレームへのデータの分割を実行し、L1 フレーム内の RHSIF インターフェイスを介して送信されます。データの準備の他、L2 モジュールはイニシエーターとターゲットノードの RHSIF モジュール間のプロセス通信も制御します。

トランスポート層は以下の主機能を提供します。

- 4つの RHSIF チャンネルのサポート：
  - チャンネル2のストリーム転送機能付き
  - チャンネルアービトレーション
- ヘッダーの生成：
  - トランザクション ID の管理
  - チャンネル番号コードテーブル I および II のサポート
- CRC-16 の生成/検証
- 専用機能によるイニシエーターおよびターゲットノードモードのサポート

図 6 に L2 サブモジュール概要のブロック図を示します。

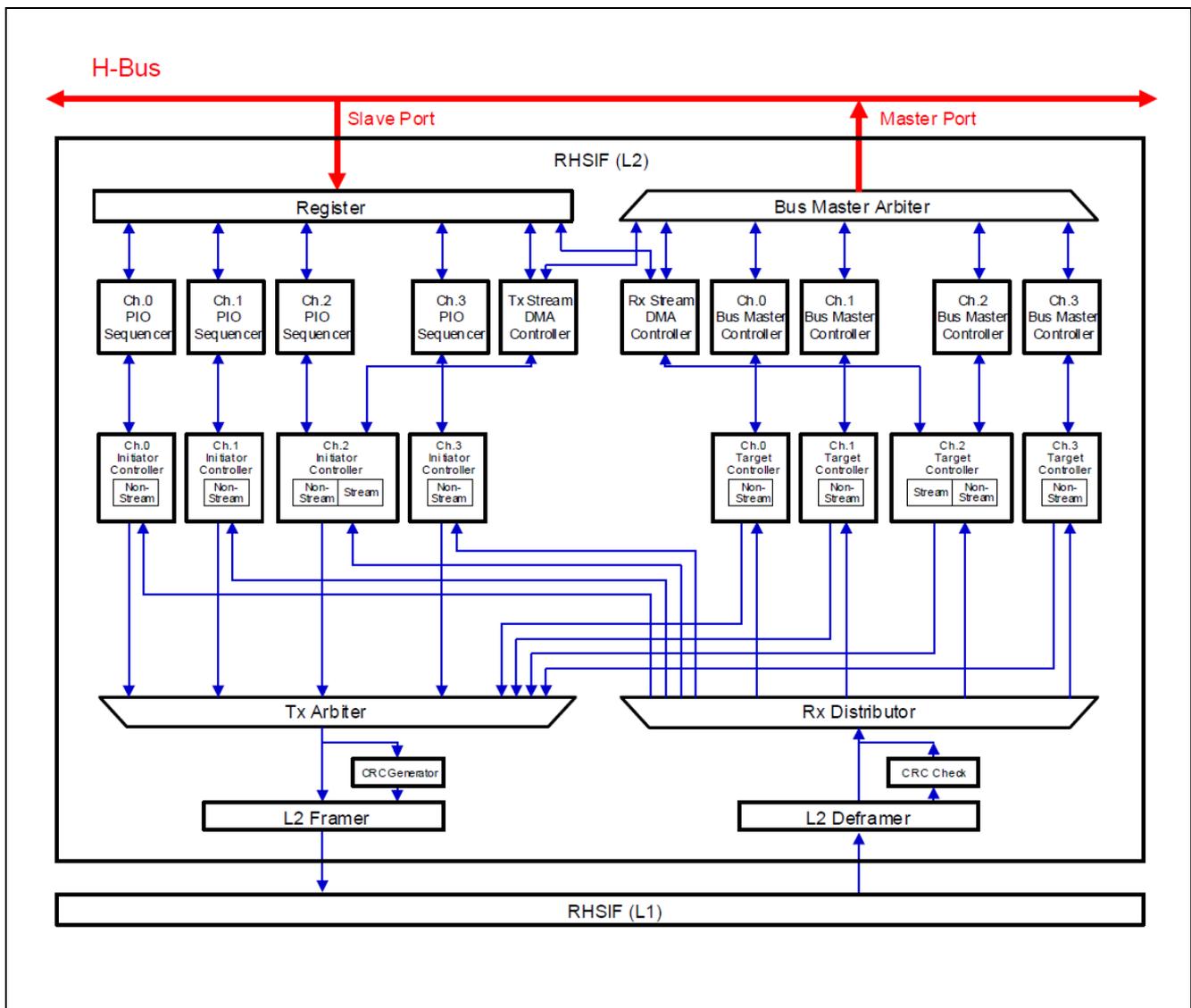


図 6：トランスポート層簡易ブロック図

## 5.7 セキュリティ/安全機能

RHSIF インターフェイスは、以下のセキュリティおよび安全機能に対応しています。これらの機能は、RHSIF コンフィギュレーションレジスタのコンフィギュレーション設定またはオプションバイト設定を介して有効化可能です。

- CRC-16 保護転送
- プログラマブルタイムアウト
- 自動転送 ID 生成によるドロップフレーム検出
- ウィンドウ設定による外部マスターからのアクセス保護
  - 4つのアクセスウィンドウを構成可能
  - 各ウィンドウに独自のアクセスルールを設定可能 (R/W、W、R)
- 初回通信時の2つの認証方式：
  - ID ベースの認証
  - ICUM と連携したチャレンジ&レスポンス認証
- RHSIF モジュールは MCU の全体的な保護ガードセキュリティシステムに実装
  - 必要に応じて MCU メモリまたはレジスタへのガード保護されたアクセスが可能
- 自己診断機能内蔵

本アプリケーションノートでは、ソフトウェア例における「ID ベース認証」処理のみを実装しています。手順の基本を理解していただくために、次章では認証を成功させるための詳細なプロセスについて説明します。「チャレンジ&レスポンス認証」と呼ばれる2番目の手順を使用して、認証のセキュリティを強化することができます。関連するガードシステムの構成と有効化も、セキュリティオプションとしてソフトウェア例に実装されています。

### 5.7.1 ID ベース認証

最初の認証は、RHSIF 認証 ID レジスタ (RHSIFnAID0...7) に格納されている 256 ビットのキーID を介して実行されます。これらのレジスタへのデータの読み取りまたは書き込みは、L2 サブモジュールのマスターポート経由でのみ（つまり、外部接続されたパートナーデバイスからのみ）可能です。別の H-Bus マスターデバイス（デバイスの別のマクロまたは CPU）がこれらのレジスタにアクセスすると、ビットは変更できず、このデバイスでは読み取り専用になります。

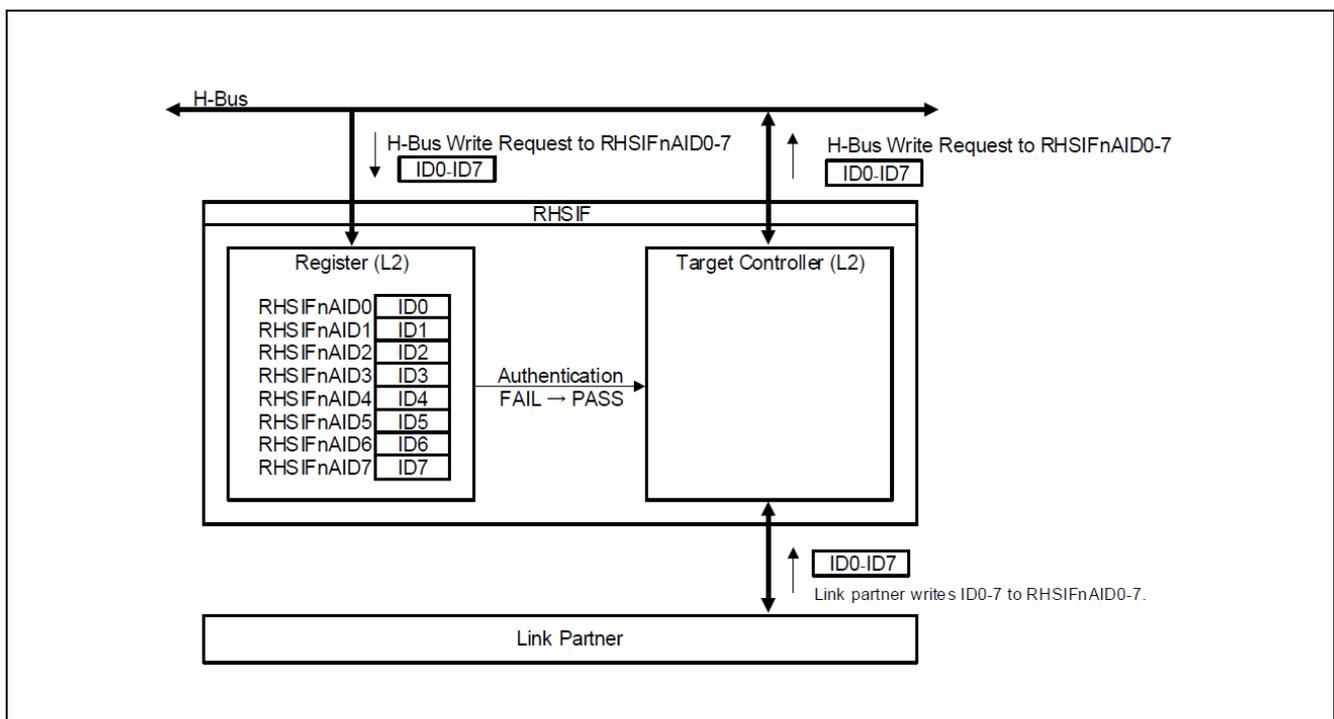


図 7 : ID ベースの認証プロセス —概要— [2]

接続された RHSIF パートナーデバイスが初期通信を実行したい場合、対象となる L2 サブモジュールの RHSIFnAID0...7 レジスタにアドレス指定された書き込みコマンドを発行する必要があります。コマンドとともに送信されるデータは 256 ビットのキーID です。したがって、パートナーデバイスは、パートナーデバイスが期待する正しい ID を認識している必要があります。

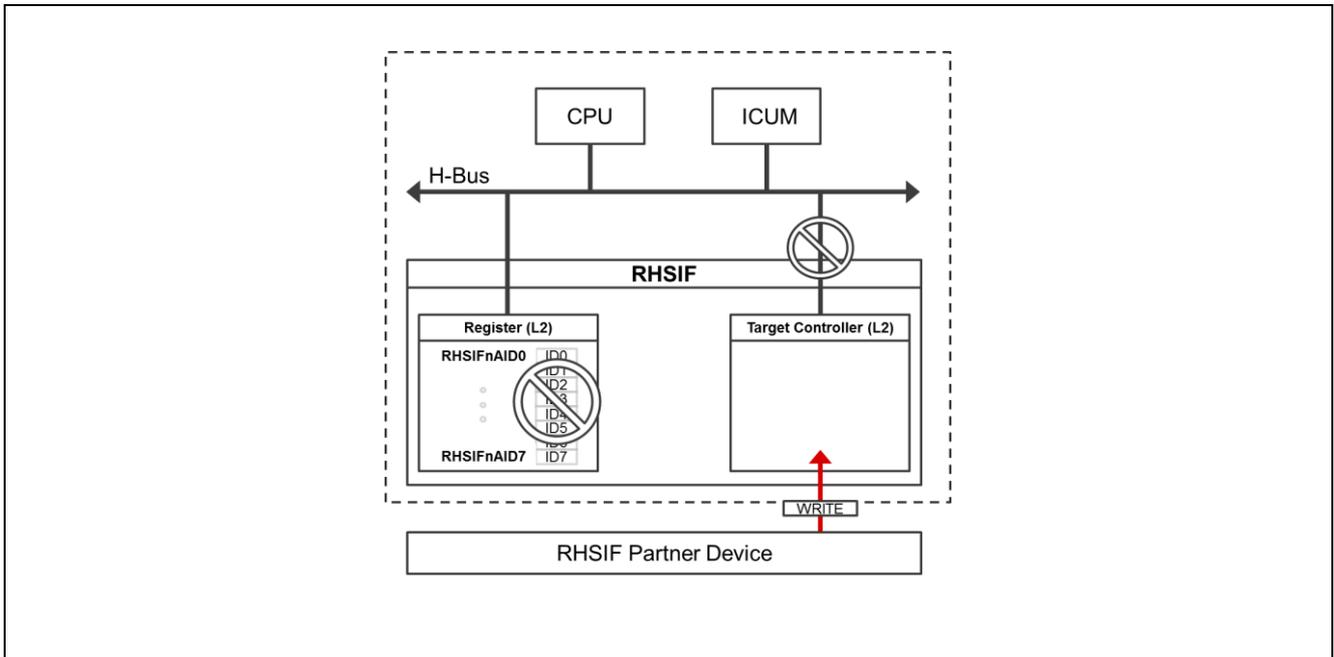


図 8 : ID ベースの認証プロセス -手順 1- [2]

パッシブ RHSIF L2 サブモジュールの対象コントローラーは、キーID セグメントを含むコマンドを受信し、受信したフレームのアドレスチェックを実行します。アドレスがキーID 情報を保持する L2 のレジスタのアドレスと一致すると、パッシブ L2 サブモジュールの対象コントローラーはデバイスの H-Bus へのアクセスを許可します。

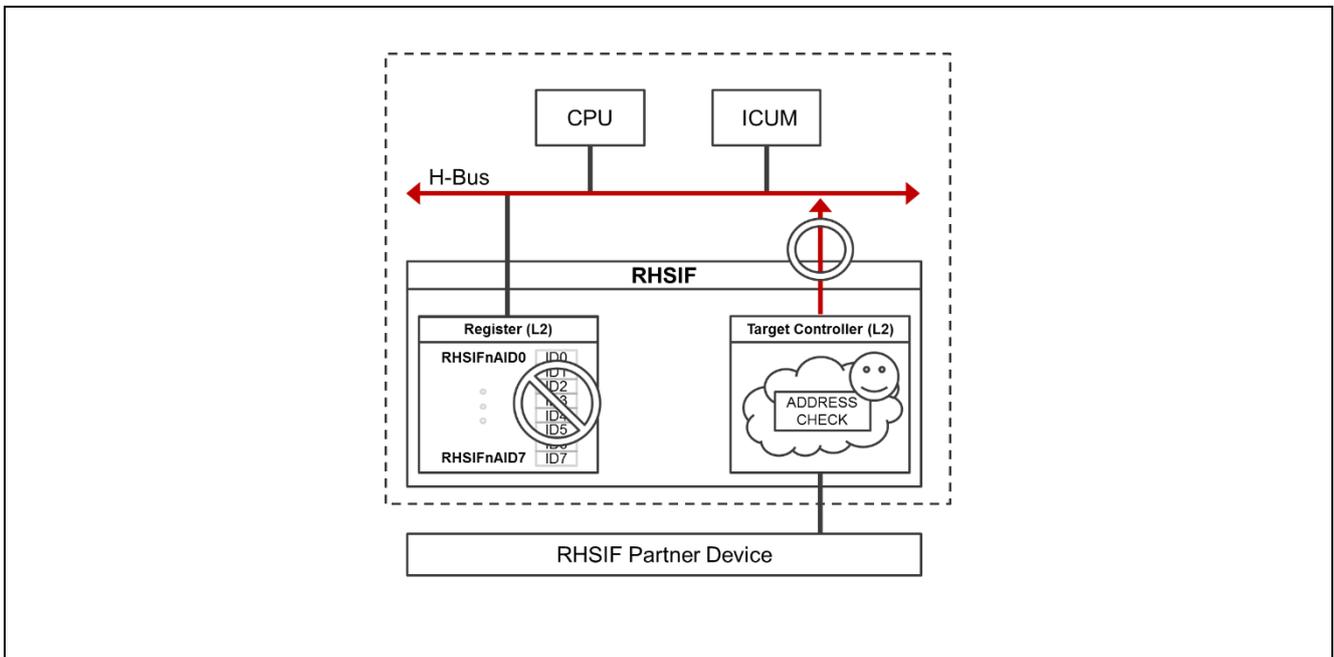


図 9 : ID ベースの認証プロセス -手順 2- [2]

RHSIF モジュールが自身のレジスタへの H-Bus 書き込みリクエストを受信すると、L2 サブモジュールは書き込みリクエスト内のマスターIDを確認します。マスターIDが L2 マスターポートの ID と一致する場合にのみ、書き込み動作が許可されます。

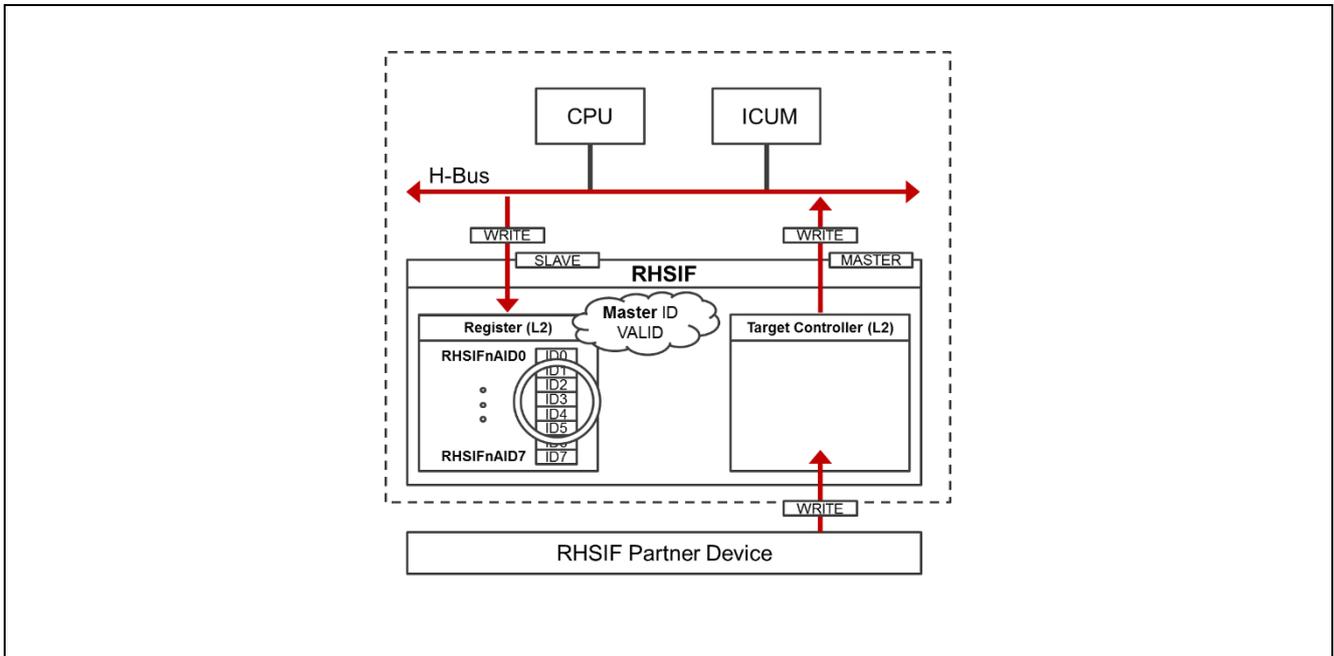


図 10 : ID ベースの認証プロセス -手順 3 - [2]

最後の手順では、対象の RHSIF モジュールの L2 サブモジュールが、書き込まれたキーID をオプションバイト内にプログラムされた ID と比較します。書き込まれた ID がフラッシュメモリ内に保存されている ID と一致する場合、RHSIF パートナーデバイスは 1 回目の認証に成功したことになります。

現在、2つのオプションが可能です：

- それ以上の認証チェックを行わなくても、それ以降の通信は可能です。
- セキュリティレベルを高めるためのチャレンジ&レスポンス認証を行うことが可能です。

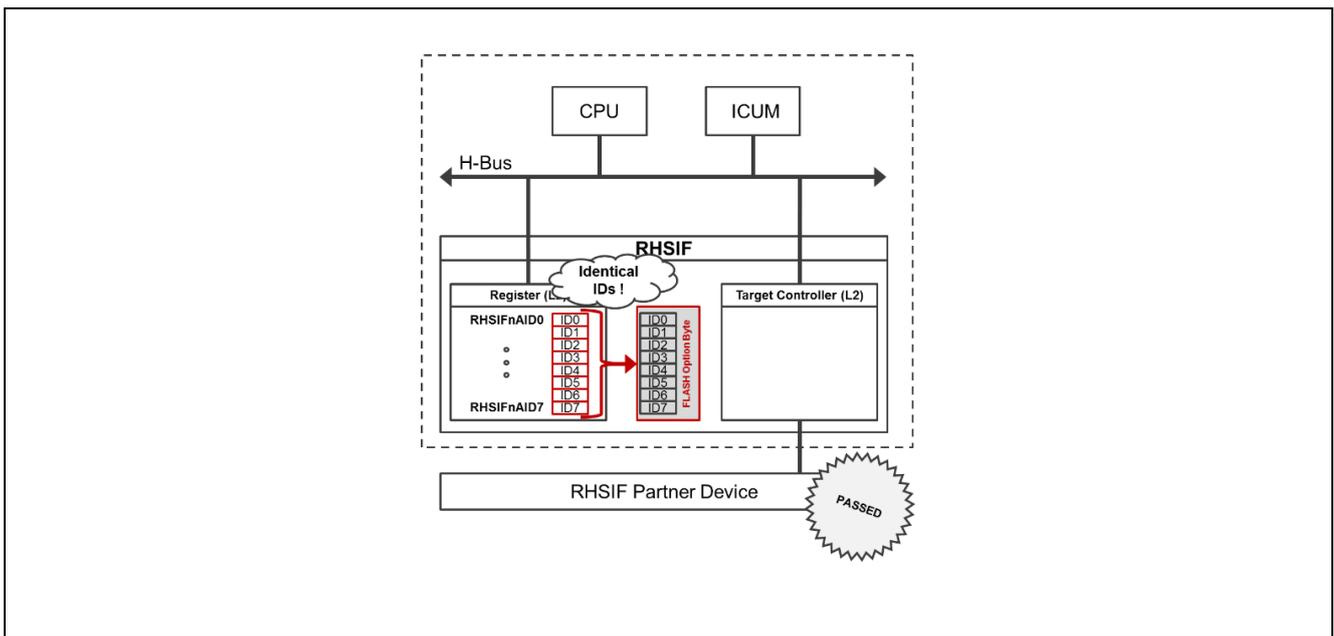


図 11 : ID ベースの認証プロセス -手順 4 - [2]

### 5.7.2 チャレンジ&レスポンス認証

最初の ID ベースの認証が正常に完了すると、2 番目のレベルの認証を実行可能です。この認証は、両方のデバイス ICUM が関与するチャレンジ&レスポンス手順の概念に従います。添付のソフトウェア例では、チャレンジ&レスポンス認証は実装されていません。以下の説明は、RHSIF セキュリティ機能の基本的な概要のみを示し、他 MCU モジュールおよびバスとのモジュールの相互接続についての理解を深めます。

本認証を開始するには、他（パッシブ）デバイスにアクセスしたいデバイス（アクティブ）が、その ICUM でチャレンジデータのセットを準備します。ICUM は、準備されたデータを RHSIF IP コントローラーに送信します。コントローラーは最終的に、準備されたチャレンジデータをリンクパートナーの RHSIF コントローラーに送信します。

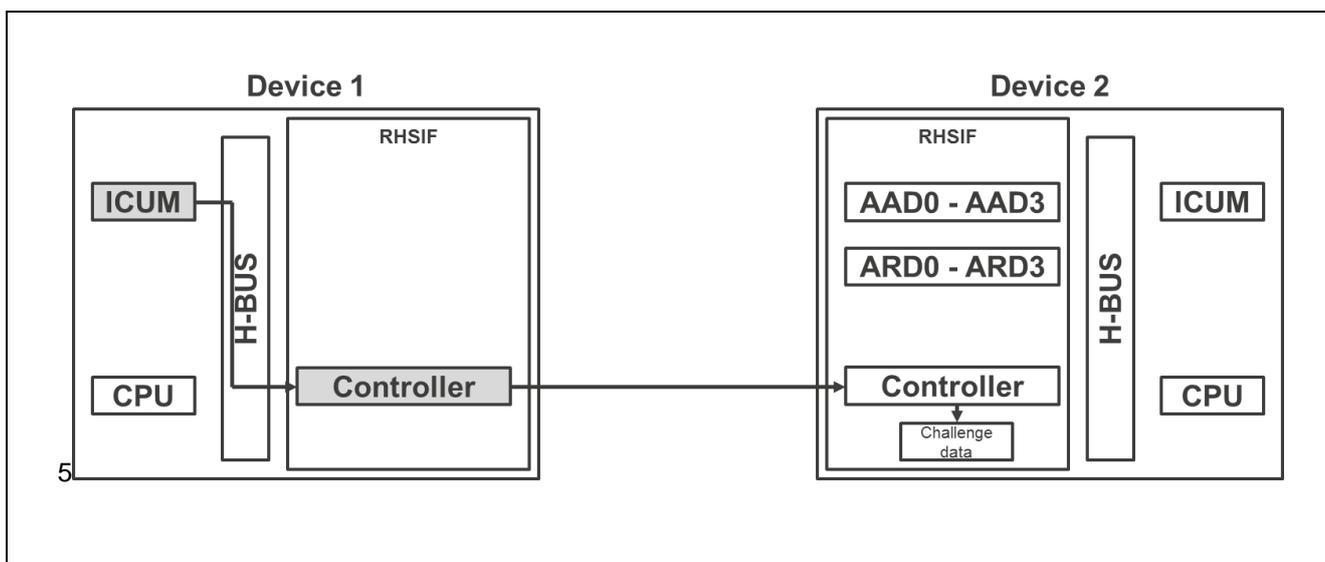


図 12 : チャレンジ&レスポンス認証 –手順 1 – [2]

データ受信後、パッシブ RHSIF コントローラーは対象デバイス上にある ICUM を有効化します。受信したチャレンジデータは、内部 H-Bus 経由で ICUM に転送され、さらなる処理が行われます。

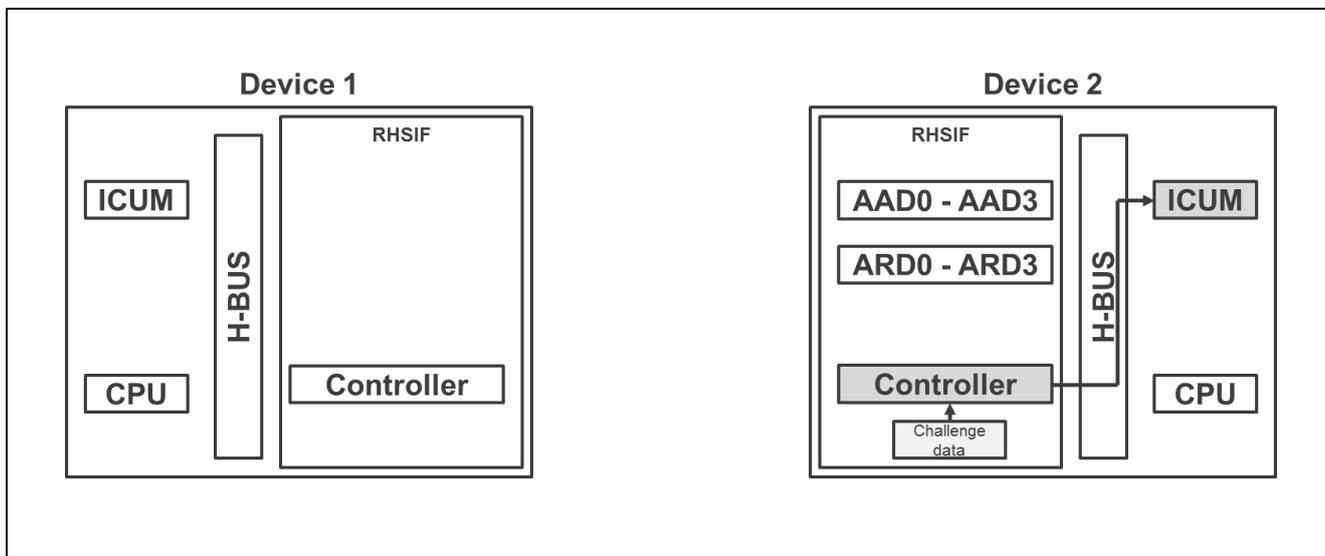


図 13 : チャレンジ&レスポンス認証 –手順 2 – [2]

パッシブデバイスの ICUM はチャレンジデータを処理し、レスポンスデータを RHSIF モジュールの AAD0~3 レジスタに書き込みます。ICUM のビルドのみがこれらのレジスタに書き込み可能です。

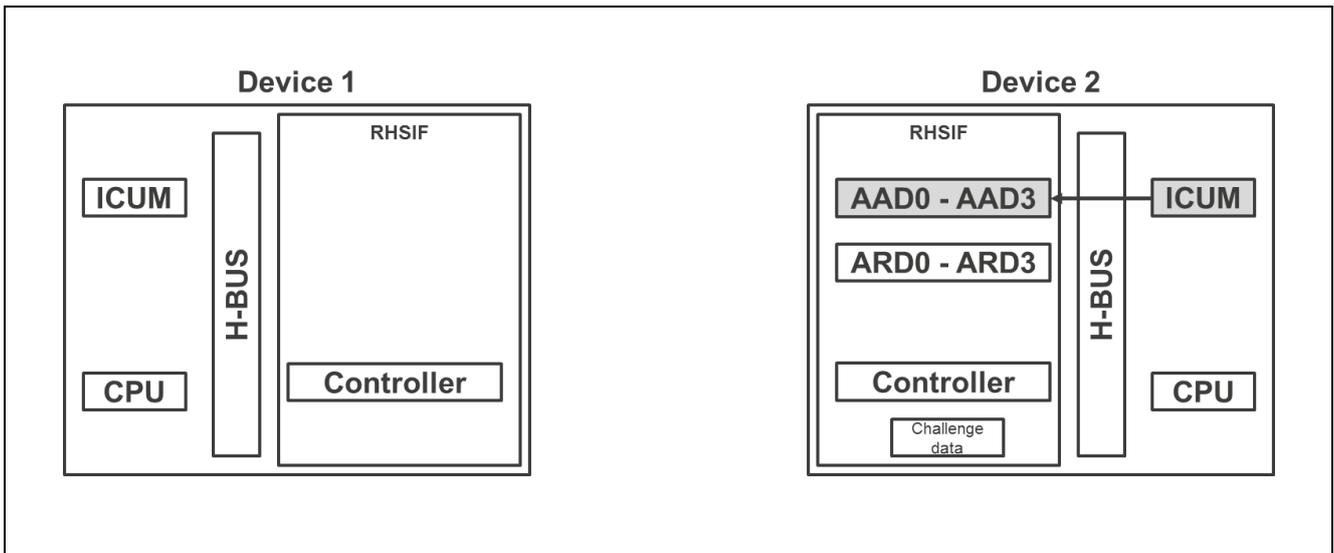


図 14 : チャレンジ&amp;レスポンス認証 –手順 3 – [2]

アクティブデバイスの ICUM は独自に設定されたレスポンスデータを計算します。計算が完了すると、データはパッシブ RHSIF コントローラーに送信され、受信した応答データが内部 H-Bus 経由で ARD0~3 レジスタに書き込まれます。デバイス独自の RHSIF モジュールのみが、H-Bus の RHSIF マスターポート経由でこれらのレジスタに書き込み可能です。

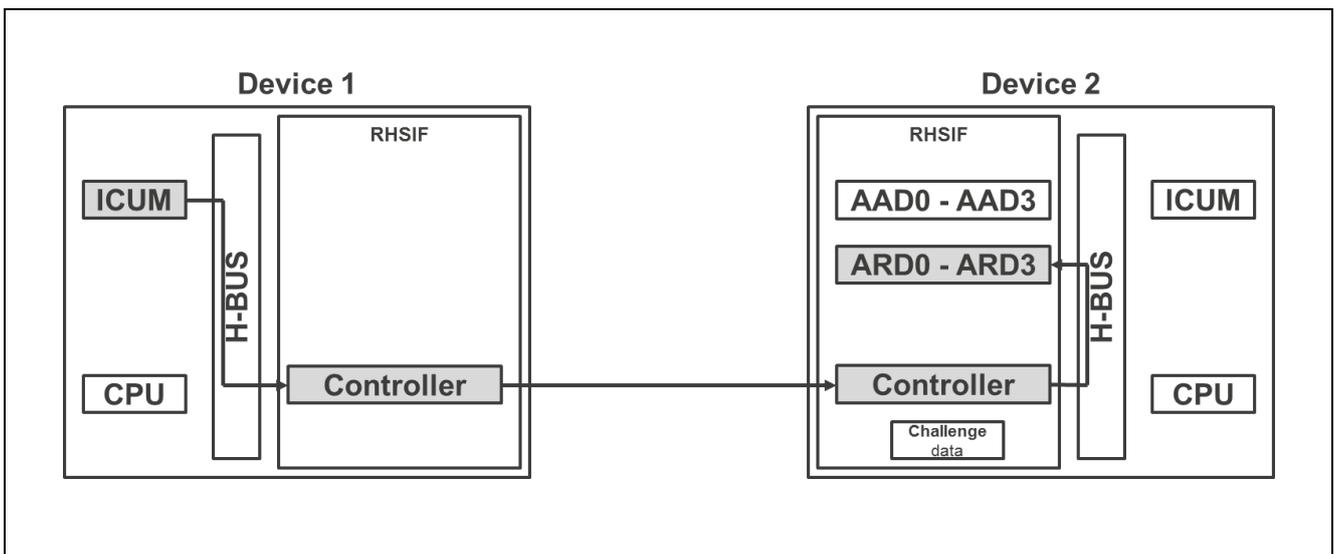


図 15 : チャレンジ&amp;レスポンス認証 –手順 4 – [2]

最後の手順で、パッシブ RHSIF モジュールは、AADn レジスタと ARDn レジスタに設定された両レスポンスデータを比較します。データが互いに一致すると、パッシブ RHSIF モジュールは、リンクパートナーデバイス自身の RHSIF モジュールの H-Bus へのアクセスを許可します。

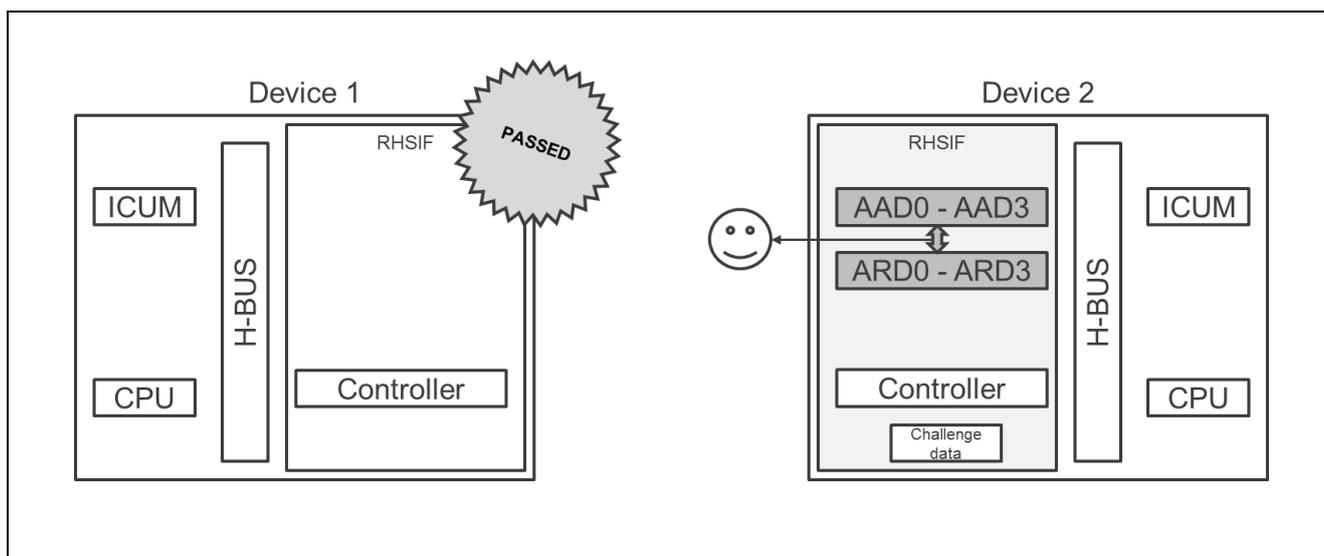


図 16 : チャレンジ&レスポンス認証 –手順 5 – [2]

### 5.7.3 メモリアドレスウィンドウ

RHSIF が対象ノードとして動作する場合、イニシエーターの RHSIF インターフェイスによって対象ノードのメモリにアクセス可能です。不特定のリンクパートナーがパートナーデバイスのメモリにアクセスする場合のセキュリティ問題を回避するため、RHSIF はメモリアドレスウィンドウの概念を実装しています。

これらのウィンドウを使用すると、ターゲットノードのメモリマップを、サイズとアクセス動作が独立した最大 4 つの異なるゾーンに分割できます。イニシエーターノードがウィンドウ化されていないメモリアドレスにアクセスしようとする、対象ノードはアクセスの試行を拒否し、RHSIFnAEST レジスタに示されるエラーが発生します。

メモリアドレスウィンドウを設定する際、メモリマップ内の開始アドレスとウィンドウのサイズを宣言する必要があります。ウィンドウは互いに重なってはいけません。ウィンドウ設定は、RHSIF インターフェイスの初期設定モード中、またはターゲットおよびイニシエーターノード機能のすべてのチャンネルを無効後に可能です。

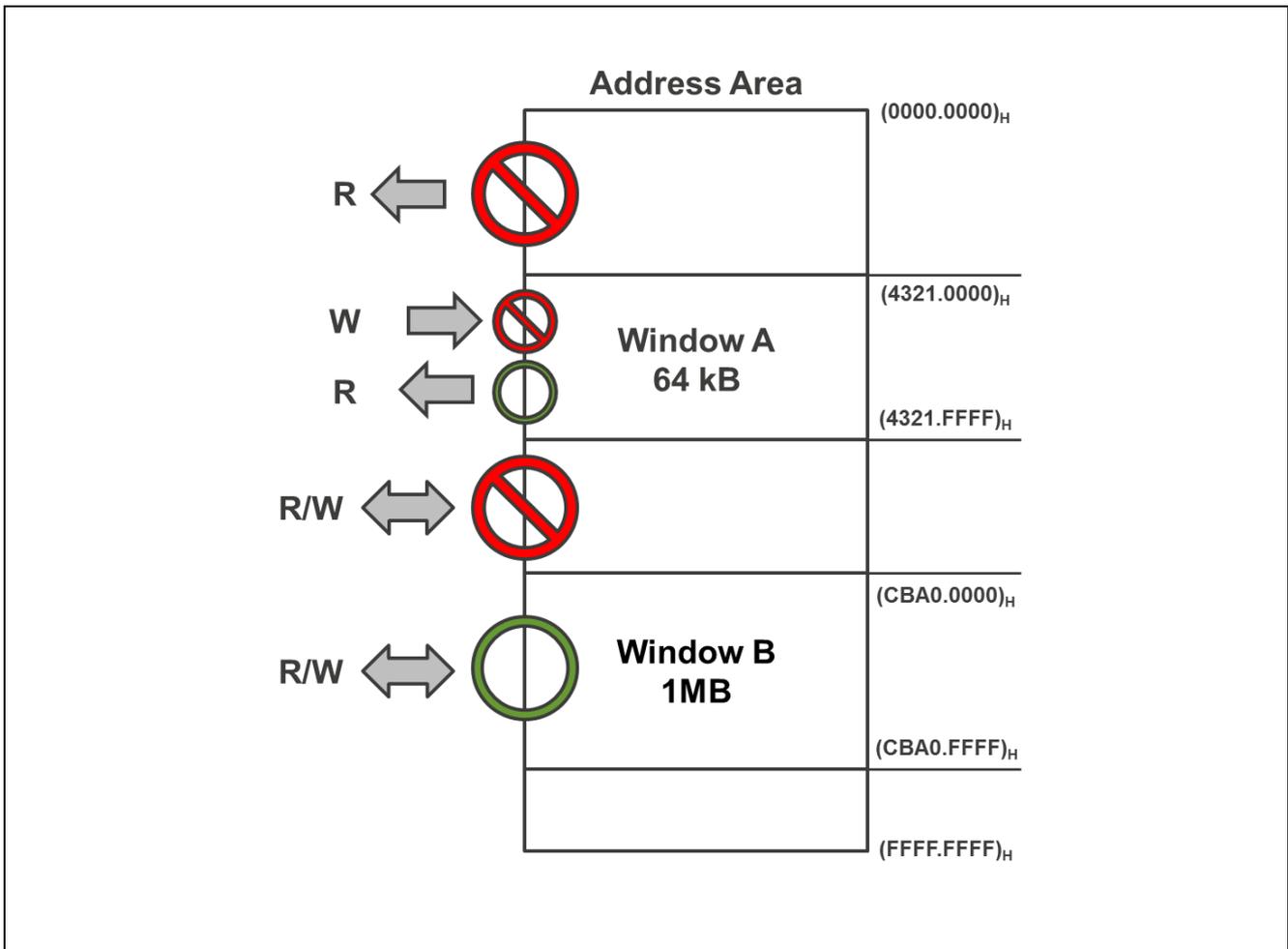


図 17 : メモリアドレスウィンドウ [2]

#### 5.7.4 保護ガード

RHSIF 等の高速周辺機器による内部メモリへのアクセスは、このアプリケーションノートにおいて重要である様々なツリーのガードシステムによって保護されています。

- H-Bus ガード (HBG)
- クラスタ-RAM ガード (CRG)
- 周辺 Bus ガード (PBG)

通常、CPU のローカル RAM を不正なアクセスから保護する PEG (プロセッサエレメントガード) などの他保護システムがあることに留意してください。このソフトウェアを専用の使用例に適応させる場合、他保護システムも同様に構成する必要がある場合があります。

#### H-Bus ガード

H-Bus は高速周辺機器 IP を相互接続し、H-Bus にアクセス可能なコアを制御するスレーブガードシステム (HBG) を備えています。HBG は読み込み/書き込みアクセスを制御します。保護の領域は周辺インスタンスです。各領域には、保護設定用の独自レジスタがあります。これらのレジスタは、不正な変更から LOCK ビットで保護されています。

HBGは2つのフィルターを順に使用して、コアがH-Busにアクセス可否をチェックします。1つ目はPEIDをチェックし、2つ目はSPIDとSPIDフィルターを比較します。不正なアクセスが検出されると、その違反がECMモジュールに通知されます。

### クラスターRAMガード

本ガードシステムは、クラスターRAMへのアクセスを制御します。クラスターRAMガードは、クラスターRAMを独立したアクセス権動作を持つ8つの異なる保護ゾーンに分離可能です。クラスターRAMガードによってアクセス許可が付与されている場合、バスマスターは宣言されたクラスターRAMゾーンの1つにのみアクセス可能です。不正なアクセス試行が検出されると、その違反がECMに通知されます。

### 周辺Busガード

周辺Busガードは、ユーザーにいくつかのPBGグループを提供します。各グループは16つの保護チャンネルを有します。単一のPBGチャンネルは、単一の周辺回路へのアクセスを制御可能です。アクセス試行がPBGによって拒否された場合、このアクセスに関連する情報は、関係するPBGグループのエラーレジスタに保存されます。

## 6 RHSIF インターフェース例 — ハードウェア設定

以下のコンポーネントは、両MCUデバイスでRHSIFアプリケーションを実装するために使用されます。

#### ソフトウェア：

- Green Hills Software MULTI® ver. 7.1.6 / Compiler V2019.5.5
- Renesas Flash Programmer (最新版を使用)
- GHS コンパイラがRH8520/U2Bxで動作するために必要な最新の「EXEC」及び「ESERV」ファイル

#### デバックツール：

- 2x Renesas E2 Emulator
- 2x PiggyBack board V1 RH850-U2Bx Evaluation Platform (486 pin version)

#### ハードウェアコンポーネント：

- マイコン：  
— 2x RH850/U2B24-FCC 486pin R7F702Z23 (WS 1.0)
- 2つのPiggyBackボードRHSIFインターフェイスコネクタを相互接続するケーブル

注：ソースコードエディターの読み込み最適化のため、TABスペースを4文字にする必要があります。

2つのマイクロコントローラーはPiggyBackボードのデバイスソケットに適用されます。PiggyBackボードの使用バージョンは、別のコネクタを介してRHSIFインターフェイスを提供されます。これら2つのコネクタは、適切なケーブルに接続します。

代替としてピンヘッダーを介して2つのデバイスを接続することも可能ですが、より高いボーレートを使用する場合は、この不十分なケーブル接続方法によりEMC問題が発生する可能性があることを留意してください。この方法を使用する場合は、専用ピンがPiggyBackボード上のMCUデバイスポートに接続されていることを確認してください。場合によっては、工場出荷時の設定ではポートがボード上のピンヘッダーに接続されていないため、ユーザーは必要な接続をはんだ付けして手動で配線を確認する必要があります。

使用E2エミュレータは、PiggyBackボードに必要な電源電圧の供給を可能とします。実装ユースケースとデバイスの追加ソフトおよびハードウェアアプリケーションによっては、外部電圧供給を適用する必要がある場合があります。E2エミュレータの電力出力は約200mAに制限されます。

GHSマルチ環境は、1つのホストシステム上で接続された2つのE2エミュレータに対して同時に独立したデバッグを実行可能です。これを実装するには、GHS MULTIの「接続オーガナイザー」の接続文字列に

以下のコマンドを追加する必要があります。コマンドを追加する前に、使用するデバッガに個別のシリアル番号を追加してください。

- -ice E2:<SERIAL No. OF E2>

図 18 に PiggyBack ボードを使用したハードウェア設定の詳細を示します。

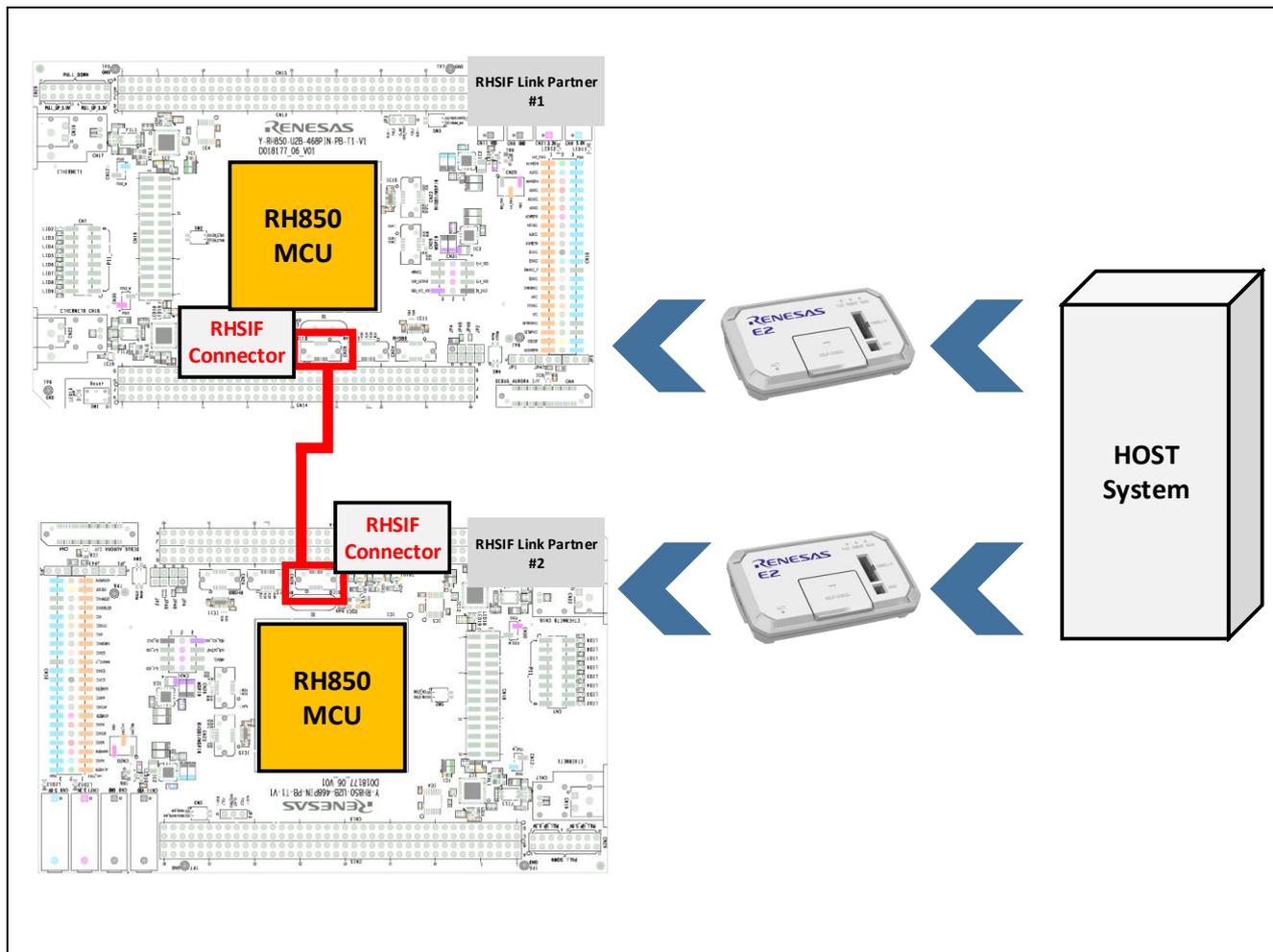


図 18 : PiggyBack ボードハードウェアの相互接続

## 7 RHSIF インターフェース例 — ソフトウェア

添付のソフトウェア例は、RH850/U2Bx MCU デバイスに実装されています。RHSIF IP を搭載した他のルネサス MCU デバイスを使用する場合は、ソフトウェアを適応させる必要があります。設定と操作の基本的な手順は同じですが、例えば、レジスタ名が異なる場合があります。

インターフェースを動作させるには、RHSIF 相互接続を有効化する前に、デバイスの次のモジュールを設定する必要があります。

- RHSIF インターフェース
- RH850/U2Bx ポート関連 RHSIF
- メモリアクセスウィンドウと関連するバスガード

各構成の詳細については、次章で説明します。説明されているソースコードセクションをユーザーがより認識しやすくするために、本アプリケーションノートでは、各セクションの先頭に説明に関連する番号が付けられています。次の短い例では、ソースコードで使用されている数値書式設定が黄色のマークで強調表示されています。

```
/*#1*/
if ( SYSCTRL.MSR_RHSIF.BIT.MSR_RHSIF_0 == 1 )
{
    SYSCTRL.MSRKCPROT.UINT32 = 0xa5a5a501; // Unlock the MSTBMS_RHSIF0 register
    SYSCTRL.MSR_RHSIF.UINT32 = 0; // Put the RHSIF to operating mode
    SYSCTRL.MSRKCPROT.UINT32 = 0xa5a5a500; // Lock the MSTBMS_RHSIF0 register
}
```

### 7.1 認証のためのセキュリティオプションバイト設定

RH850/U2Bx では、本章で示すように、ユーザーがキーID ベースの認証をアクティブにする必要があります。ユーザーは、セキュリティオプションバイト S\_OPBT\_6 を使用して、選択したアクティブバージョンのステータスを確認できます。レジスタのデフォルト値は以下となります：

- S\_OPBT6 = (FFFE FFFF)<sub>H</sub>

ビット 16 (S\_OPBT6.HSIF\_IDAUTH\_NEED) は、リンクパートナー認証が必要かどうかを定義します。

ICUM が MCU デバイス上で有効化されると、前章で示したように、キーID ベースの第 1 レベルの認証が第 2 レベルのチャレンジ&レスポンス認証手順にアップグレードされます。説明されているソフトウェア例では、ICUM が使用されないため、キーID ベースの認証のみが実装されています。

セキュリティオプションバイト変更を実行するには、Renesas Flash Programmer (RFP) を使用する必要があります。RFP の現実装では、プロジェクトファイルをプログラマツールにロードする必要があります。「デバイス情報」ボタンをクリックしてから「メモリ情報の読み取り」をクリックして、メモリ設定を読み出す必要があります。選択範囲で SECURITY 領域のみを選択し、.mot ファイルをローカルドライブに保存します。

必要設定を変更するには、ファイル内容を変更する必要があります（例「S-Record」修正プログラム）。オプションで、通常エディタを使用して設定を変更できますが、その場合は正しいチェックサムを手動で計算する必要があります。

1

2

3

### Read out HEX dump

| #  | Sx | Size | Address  | Data                             | Csum |
|----|----|------|----------|----------------------------------|------|
| 0  | 50 | 11   | 0000     | 4372656174656428627920524650     | 33   |
| 1  | 53 | 15   | FF300040 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 8B   |
| 2  | 53 | 15   | FF300050 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 7B   |
| 3  | 53 | 15   | FF300060 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 6B   |
| 4  | 53 | 15   | FF300070 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 5B   |
| 5  | 53 | 15   | FF300080 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 4B   |
| 6  | 53 | 15   | FF300090 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 3C   |
| 7  | 53 | 15   | FF3000A0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 2B   |
| 8  | 53 | 15   | FF3000B0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 1B   |
| 9  | 53 | 15   | FF3000C0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 0B   |
| 10 | 53 | 15   | FF3000D0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | FB   |
| 11 | 53 | 15   | FF3000E0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | EB   |
| 12 | 53 | 15   | FF3000F0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | DB   |
| 13 | 53 | 15   | FF300100 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | CA   |
| 14 | 53 | 15   | FF300110 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | BA   |
| 15 | 53 | 15   | FF300120 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | AA   |
| 16 | 53 | 15   | FF300130 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 9A   |
| 17 | 53 | 15   | FF300140 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 8A   |
| 18 | 53 | 15   | FF300150 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 7A   |
| 19 | 53 | 15   | FF300160 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 6A   |
| 20 | 53 | 15   | FF300170 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 5A   |
| 21 | 53 | 15   | FF300180 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 4A   |
| 22 | 53 | 15   | FF300190 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 3A   |
| 23 | 53 | 15   | FF3001A0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 2A   |
| 24 | 53 | 15   | FF3001B0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 1A   |
| 25 | 53 | 15   | FF3001C0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 0A   |
| 26 | 53 | 15   | FF3001D0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | FA   |
| 27 | 53 | 15   | FF3001E0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | EA   |
| 28 | 53 | 15   | FF3001F0 | FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF | 5A   |
| 29 | 57 | 05   | 00000000 |                                  | FA   |

図 19 : RFP ツールによるメモリー読み込み

最後に、変更した`.mot`ファイルをマイコンにロードする必要があります。RFPで「参照」ボタンをクリックして、前手順で変更したファイルを選択することによって実行されます。RFPの「動作設定」の**ERASE、PROGRAM、VERIFY**に必ずチェックを入れてください。「操作」セクションの「開始」ボタンを押すと、プログラミングプロセスが開始されます。

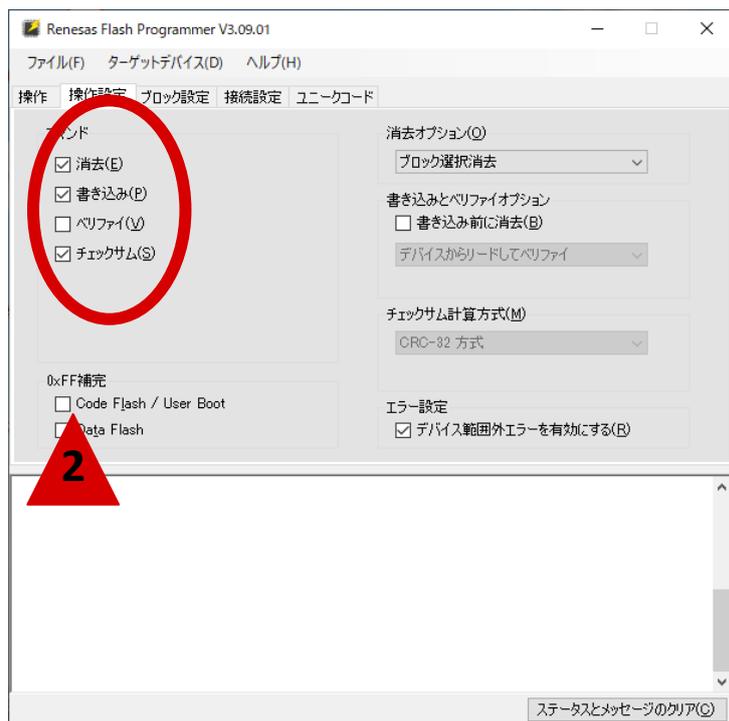
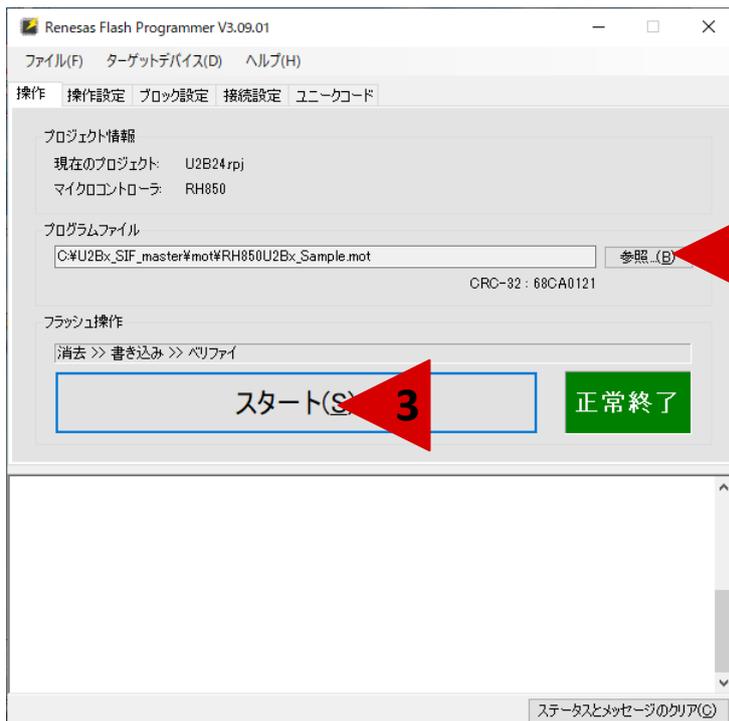


図 20 : RFP ツールによるメモリ書き込み

## 7.2 デモソフトウェアの構成

RHSIF デモプロジェクトは、デュアルコア RH850/U2Bx で実現されます。RHSIF 操作の重要なコード行は、以下のソースコードとヘッダーファイルにあります：

- RHSIF.h
- main\_pe0.c
- RHSIF.c

### 7.2.1 C ヘッダーファイル「RHSIF.h」

本ヘッダーファイルは、プロジェクトで使用されるいくつかの型を定義します。定義は以下のとおりです。

- **RHSIF\_State\_t**  
— RHSIF ドライバーSW の現状態を表示するために使用されます。
- **L1\_Payload\_Size\_t**  
— L1 のさまざまな有効ペイロードサイズを定義します。
- **L1\_Logical\_Channel\_Type\_t**  
— 4つのチャンネル、CTS、およびインターフェイス制御を定義します。
- **RHSIF\_Mode\_t**  
— 2つのモードを定義します。：MASTER/SLAVE
- **RHSIF\_L2Command\_t**  
— L2 コマンドの様々な形式を定義します。

注：利用可能な ICLC、L1、L2 コマンドのすべてがデモソフトウェアで使用されているわけではありません。

すべての型の定義の後、RHSIF.c ファイルで使用されるサブ関数が、使用される変数を使用して宣言されます。

### 7.2.2 C コードファイル「RHSIF.c」

本関数は、本アプリケーションノートで使用されるすべての RHSIF 関連関数を定義します。以下の関数が実装されています。

- **RHSIF\_Init**  
— インターフェース初期化
- **RHSIF\_L2\_Send1stAuth**  
— キーID ベースの認証送信処理
- **RHSIF\_L2\_wait1stAuth\_Check**  
— 認証が正常に進行中か確認
- **RHSIF\_L1\_SendPing**  
— ICLC コマンドを介して L2 サブモジュールに PING コマンドを発行
- **RHSIF\_L2\_setWindow**  
— メモリアクセスウィンドウ設定
- **RHSIF\_L2\_read32**  
— 読み込み動作実行
- **RHSIF\_L2\_write32**  
— 書き込み送信準備
- **loc\_RHSIF\_portInit**  
— ポートグループ及びピン関連 RHSIF 初期化
- **RHSIF\_L2\_stream256\_read**  
— ストリーム読み取り動作を実行
- **RHSIF\_L2\_stream256\_write**  
— ストリーム書き込み送信準備

### 7.2.3 C コードファイル「main\_pe0.c」

本プロセッサエレメント 0 (PE0) の主な関数です。ソースコード内で、以下の RHSIF 関連イベントが実装されています。

- **マスターデバイス用：**
  - マスターデバイスとしての RHSIF IP 初期化の関数呼び出し
  - 第 1 レベル認証関数呼び出し
  - 第 1 レベル認証確認関数呼び出し
  - 読み込み関数の関数呼び出し
  - 書き込み関数の関数呼び出し
  - 送信 PING 関数の関数呼び出し
  - ストリーム書き込み関数の関数呼び出し
  - 書き込みデータ準備
- **スレーブデバイス用：**
  - スレーブデバイスとしての RHSIF IP 初期化の関数呼び出し
  - 第 1 レベル認証関数呼び出し
  - 第 1 レベル認証確認関数呼び出し
  - 読み込み関数の関数呼び出し
  - 書き込み関数の関数呼び出し
  - 送信 PING 関数の関数呼び出し
  - ストリーム書き込み関数の関数呼び出し
  - 書き込みデータ準備

図 21 にソースファイル「main\_pe0.c」のフローチャート関連 RHSIF を示します。

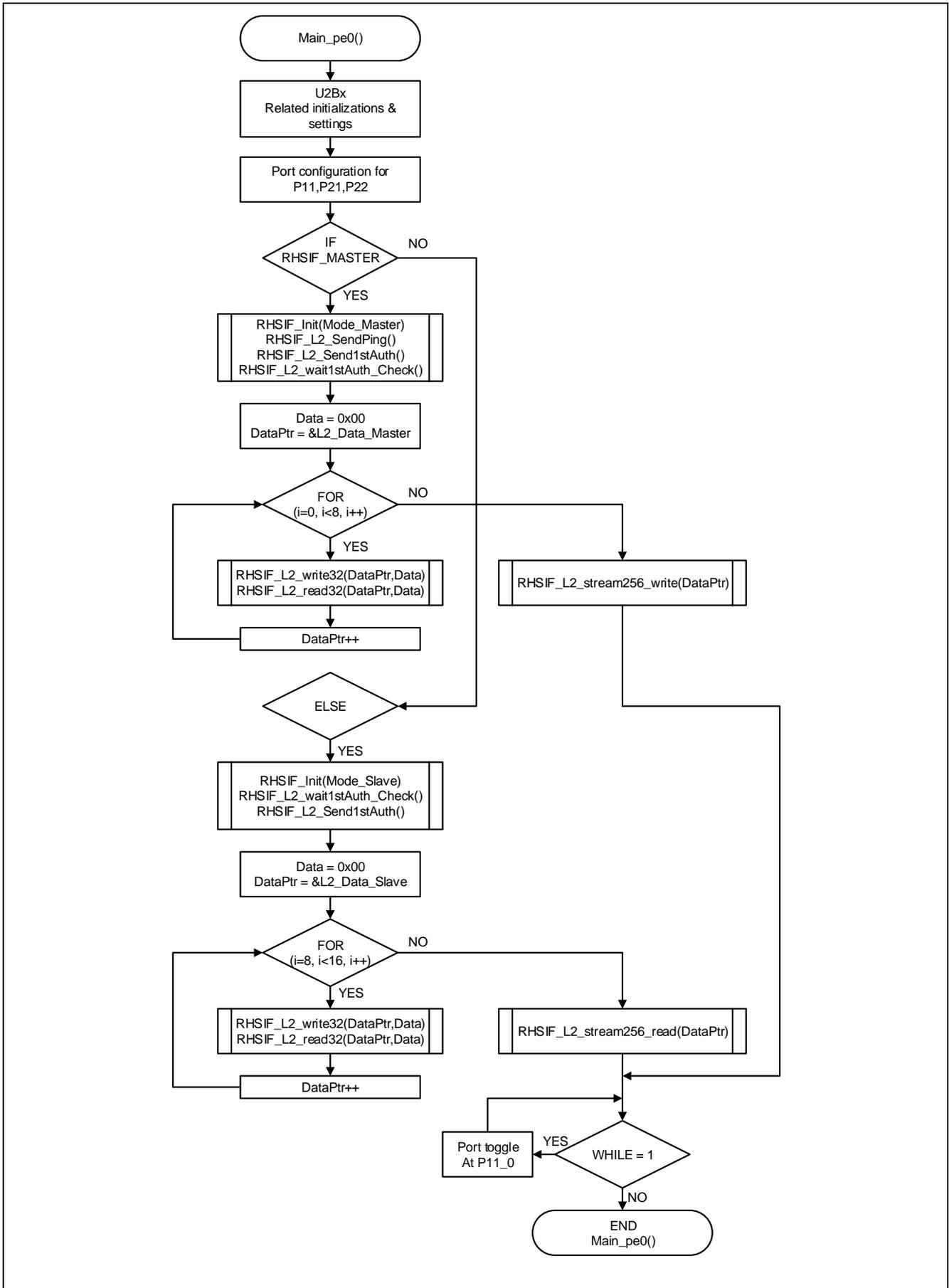


図 21 : 「main\_pe0.c」 フローチャート

### 7.3 ソフトウェアプロセスチャート

図 22 に全ソフトウェアプロセスと結合されたワークフローにおける 2 つの各デバイスの依存関係を示します。

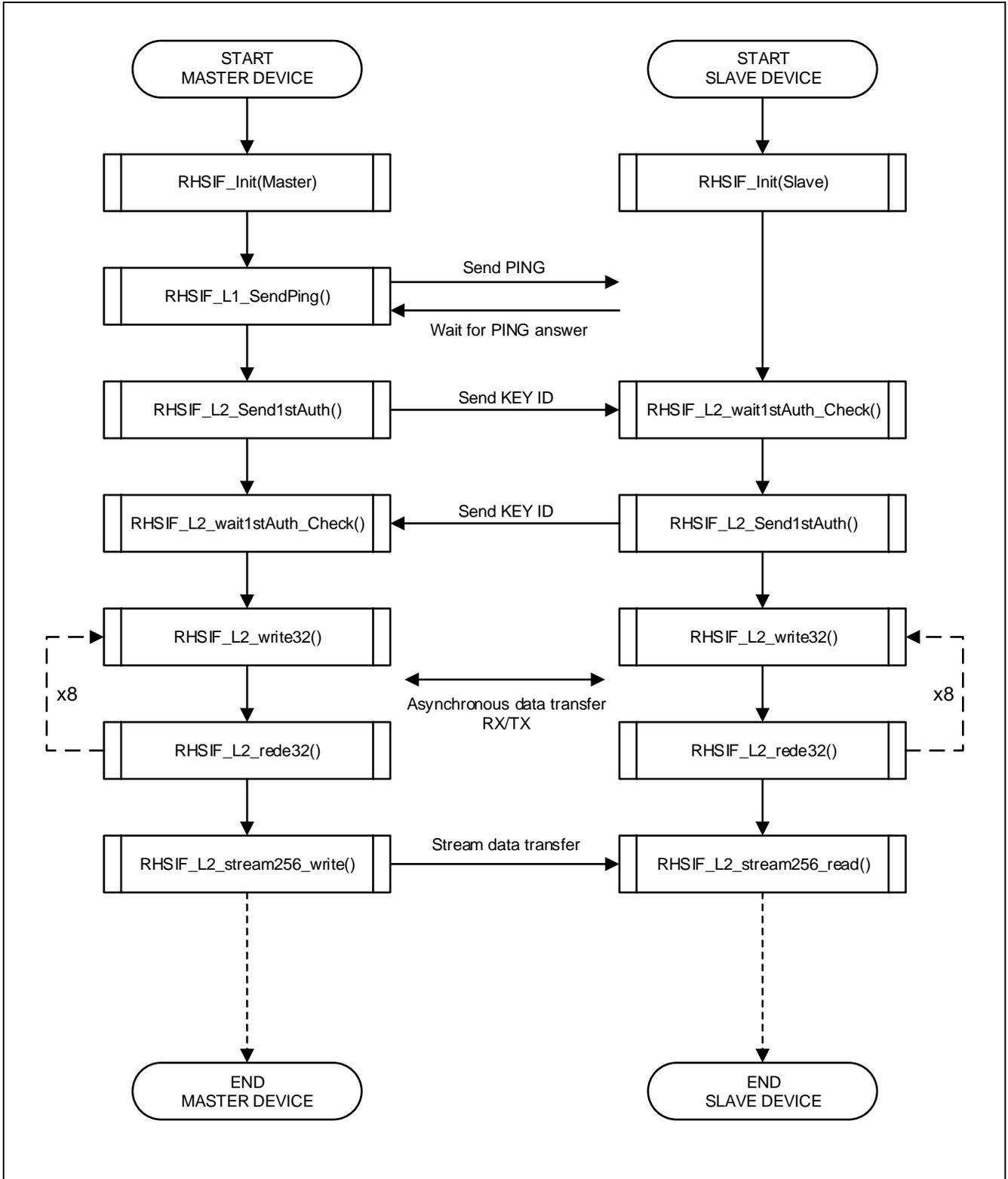


図 22 : ソフトウェアプロセスチャート

## 7.4 RHSIF モジュール初期設定

RHSIF インターフェイスの初期設定は、ソースコードファイル「**RHSIF.c**」内で実行されます。関連関数は「**RHSIF\_Init**」です。

- #1. 待機状態から RHSIF へ解除します。これは、MSR\_RHSIF レジスタの MS\_RHSIF0 ビットを 0 にクリアすることによって行われます。レジスタは意図しない書き込みアクセスから保護されているため、書き込みアクセス前に MSRKCPROT レジスタを介してロックを解除する必要があります。
- #2. インターフェイスの設定のために RHSIF の PLL を無効にします：  
RHSIFnPCR.PLLSTBY ビットを 1 に設定。
- #3. アプリケーション要求に従って RHSIF0MDCR レジスタを設定：
  - a. デバイスを RHSIF マスターまたは RHSIF スレーブとして設定します。  
RHSIF0MDCR.MST ビットを対応した設定をする必要があります。
  - b. CLKSEL は REFCLK 周波数を 10MHz または 20MHz に設定します。
  - c. CTSV ビットは、L1 ヘッダーの CTS 値を設定します。
  - d. CTSEN は自動フロー制御を設定します。
- #4. RHSIFnTXRXCR レジスタを設定して RHSIF の送受信を有効にします。
- #5. アプリケーション要求に従って RHSIFnSPCR レジスタを設定：
  - a. RXSP ビットと TXSP ビットは、インターフェイスを受信/送信用に低速モードまたは高速モードに設定します。
  - b. 高速モードが選択されている場合、FMBR1 ビットは実際のボーレートです。
- #6. RHSIFnPCR.PLLSTBY ビットを 0 にクリアして PLL を有効化します。
- #7. 最後に RHSIF 関連のポートピンを設定します。

### 7.4.1 保護ガード設定

ガード設定は、関数「**RHSIF\_Init**」内にあります。デフォルトでは、パワーオンリセット後にガードは有効化しません。ガードシステムを使用したい場合は、デモコード内のコメント行のコメントを削除する必要があります。

- #8. ユーザーマニュアルに記載されている設定手順に従い、クラスタ RAM ガードを無効化して下さい。
  - a. クラスタ RAM ガードの制御レジスタ保護を解除して、設定変更を有効化
  - b. アドレス領域を設定
  - c. ガードを有効化 (任意)
  - d. クラスタ RAM ガードの制御レジスタを保護
- #9. ユーザーマニュアルの設定手順に従って、H-Bus ガードを無効にします。
  - a. H-Bus ガード制御レジスタの保護解除
  - b. 制御レジスタを介して H-Bus ガードを有効化 (任意)
  - c. 専用ビットシーケンスを保護レジスタに書き込むことで制御レジスタを保護

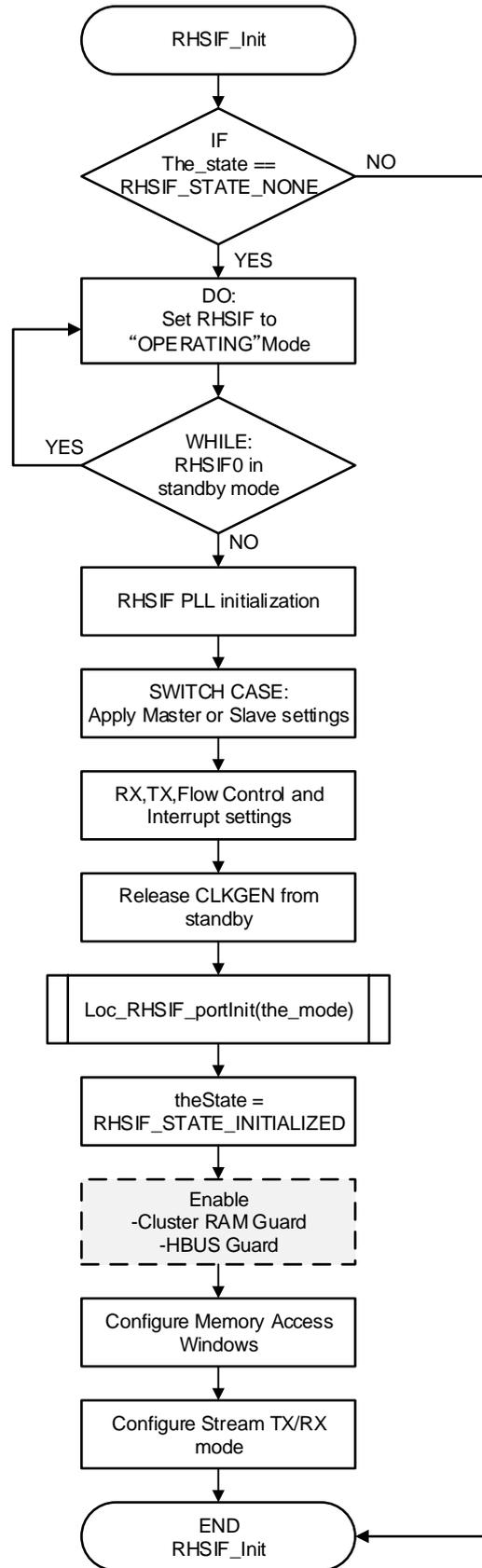


図 23 : 関数「RHSIF\_Init」フローチャート

## 7.4.2 メモリアクセスウィンドウ構成

単一メモリアクセスウィンドウの構成は、関数「**RHSIF\_Init**」内に示されます。

#10.関数「**RHSIF\_L2\_setWindow**」を使用して、メモリアクセスウィンドウの **RHSIF** 開始アドレスとサイズを設定します。

- a. 関数「**RHSIF\_Init**」内の関数呼び出しとともにユーザーが渡した値に基づいて、**Start**、**Size**、および **Window No.**パラメーターを計算します。

## 7.4.3 ストリーム TX/RX 構成

以下はストリーム転送の初期設定です。

#11.ストリーム Tx モード構成

- a. **RHSIFnMRT** レジスタをモジュール応答タイムアウトに設定します。
- b. **RHSIFnSTMD** レジスタの **STPS** ビットでデータペイロードサイズを 128 ビットまたは 256 ビットに設定します。
- c. **RHSIFnSTBC** レジスタにストリーム Tx バイト数を設定します。

#12.ストリーム Rx モード構成

- a. **RHSIFnSRMD** レジスタの **SRPS** ビットでデータペイロードサイズを 128 ビットまたは 256 ビットに設定します
- b. **RHSIFnSRDS** レジスタにストリーム Rx 先領域サイズを設定します。
- c. **RHSIFnSRBC** レジスタにストリーム Rx バイト数を設定します。

## 7.5 マイコンポート構成

**RHSIF** インターフェイスに必要なポート設定の構成は、ソースコードファイル「**RHSIF.c**」内で実行されます。関連関数は「**loc\_RHSIF\_portInit**」です。

- #1. ポート制御レジスタは、設定変更から保護されています。構成を有効化するには、ポートアクセス保護レジスタ **PORT0.PKCPROT** に正しいビットシーケンスを設定して、書き込み許可を有効にする必要があります。
  - a. 前手順の後、**PORT0.PWE** レジスタへの書き込みによって、ポートグループ **P20** および **P21** への書き込みアクセスを有効にする必要があります。
- #2. ポートグループ 21 の場合、ピン 2 と 3 は LVDS モードでの **RHSIF RX** 機能をサポートします。
  - a. **PORT0.LVDSCTRLA** レジスタは、ピンを入力ピンとして 3.3V LVDS モードに設定
  - b. **PORT0.PFC21** および **PORT0.PM21** レジスタを使用すると、ピン **P21\_2** および **P21\_3** が **HSIF\_RXDN** および **HSIF\_RXDP** モードに設定されます。
- #3. ポートグループ 21 の場合、ピン 4 と 5 は LVDS モードでの **RHSIF TX** 機能をサポートします。
  - a. **PORT0.LVDSCTRLA** レジスタは、**P21\_4** および **P21\_5** の LVDS 出力を有効化
  - b. **PORT0.PFC21**、**PORT0.PMC21**、**PORT0.PFC21**、および **PORT0.PSFSC21** レジスタを使用すると、ピン **P21\_4** と **P21\_5** は **HSIF\_TSDN** と **HSIF\_TXDP** で出力モードに設定されます。
- #4. クロック信号はピン **P22\_3** から供給されます。このピンは、**PORT0.PFC22** レジスタ、**PORT0.PFCAE22** レジスタ、および **PORT0.PMC22** レジスタを介して設定されます。
  - a. 選択した **RHSIF** モードに応じて、**P22\_3** ピンが **PORT0.PM22** レジスタによって入力または出力されるように設定されます。

- #5. 全ポートレジスタの必要設定を設定後、PORT11.PWE レジスタの書き込みアクセスを無効にし、ポートアクセス保護を有効にする必要があります。最後に、PORT11.PKCPROT レジスタを有効化シーケンス値で設定する必要があります。

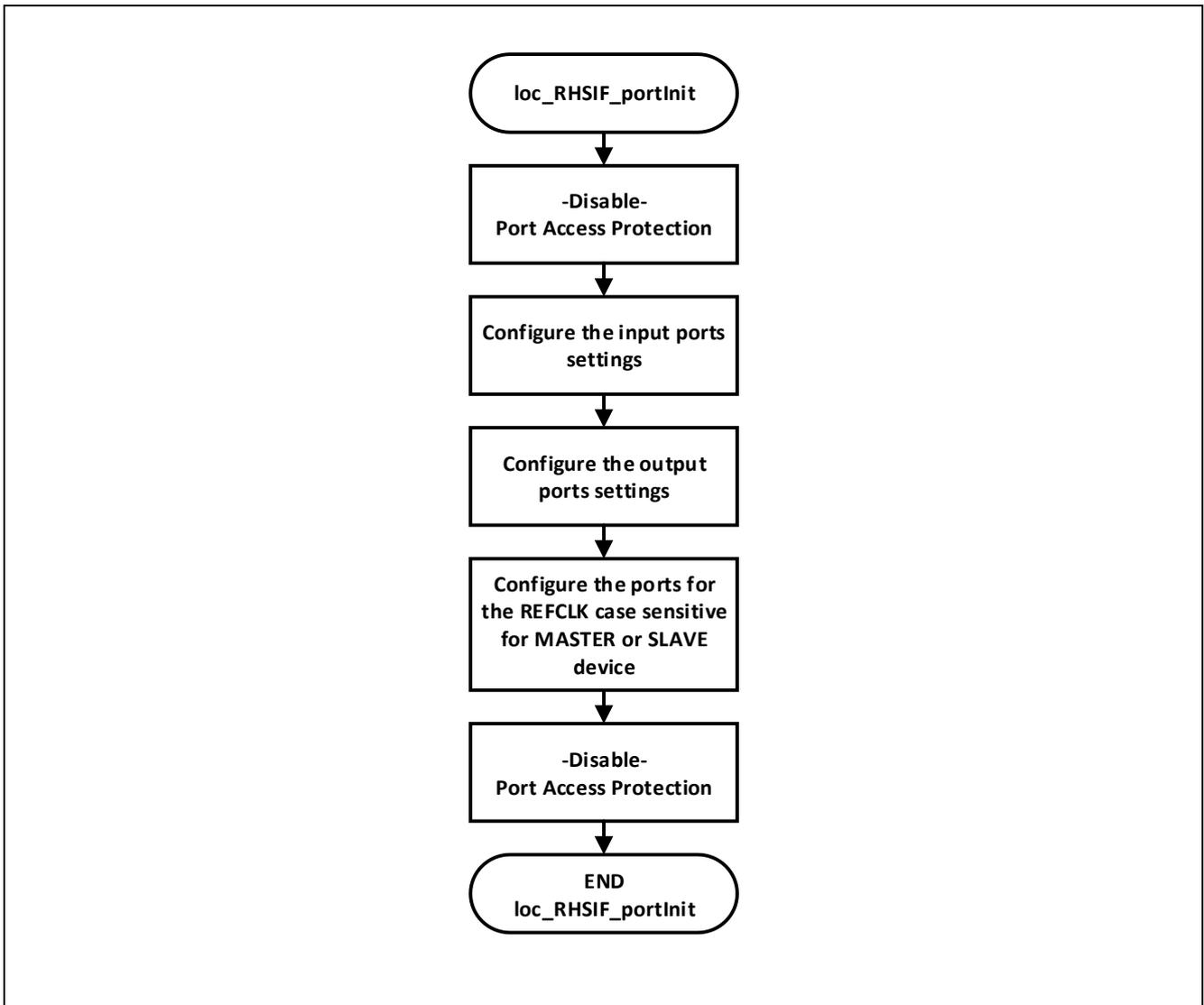


図 24 : 関数「loc\_RHSIF\_portInit」フローチャート

## 7.6 RHSIF L1 通信

### 7.6.1 概要

L1 サブモジュールで使用できる設定とコマンドは非常に限られています。本例では、様々な ICLC コマンドの 1 つを使用して、ソフトウェアに PING コマンドを実装します。ソフトウェアはマニュアルに示されているワークフローに従って実装されます。

## 7.6.2 PING コマンド

RHSIF L1 サブモジュールは、自動 PING 機能を提供します。この機能は、関数「**RHSIF\_L1\_SendPing**」を使用してソースコードファイル「**RHSIF.c**」内に実装されます。

- #1. 関数は RHSIF ドライバーが初期化されているかどうかを確認します。このステートメントが正しい場合、プログラムは PING 送信を実行します。
  - a. プログラムは、RHSIF0.ICCR の ITRG ビットが、現 ICLC コマンドがトリガーされていないことを示すまで、次の実行を停止します。
  - b. 全てのエラーおよびステータスレジスタが、関連するステータスクリアレジスタを介してクリアされます。
  - c. RHSIF0.ICCR レジスタに書き込むことによって、PING コマンド (選択されたペイロード (0x00)H によって示される) と ICLC コマンド TX トリガーが発行されます。

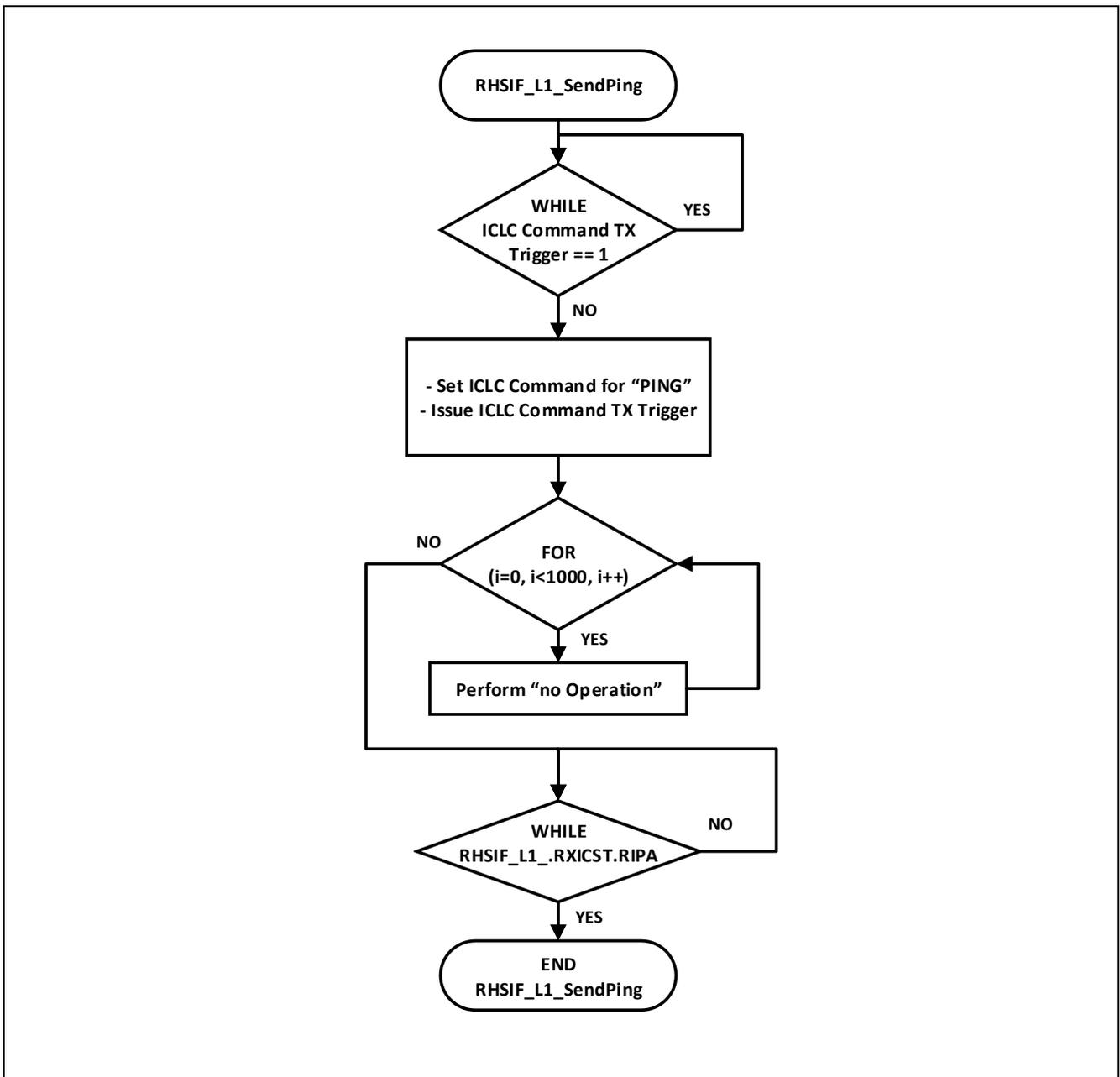


図 25 : 関数「RHSIF\_L1\_SendPing」フローチャート

## 7.7 RHSIF L2 通信

### 7.7.1 概要

L2 関連レジスタを使用すると、ユーザーは通常 RHSIF 通信を制御できます。例えば、READ および WRITE コマンドが実装され、SEND コマンドによってトリガーされます。2つのリンクパートナー間の相互接続のロックを解除して確立するには、最初の送信前に第1レベルを認証(キーIDベース)する必要があります。

### 7.7.2 第1認証

キーIDベースの第1レベル認証は、関数のファイル「**RHSIF.c**」内に示されています

#### “RHSIF\_L2\_Send1stAuth”

- #1. イニシエーターRHSIF ノードコマンドは、定義されたキーIDを対象のRHSIF ノードに書き込みます。書き込みシーケンスは RHSIF\_L2\_write32 関数で実行されます。本章に示します。

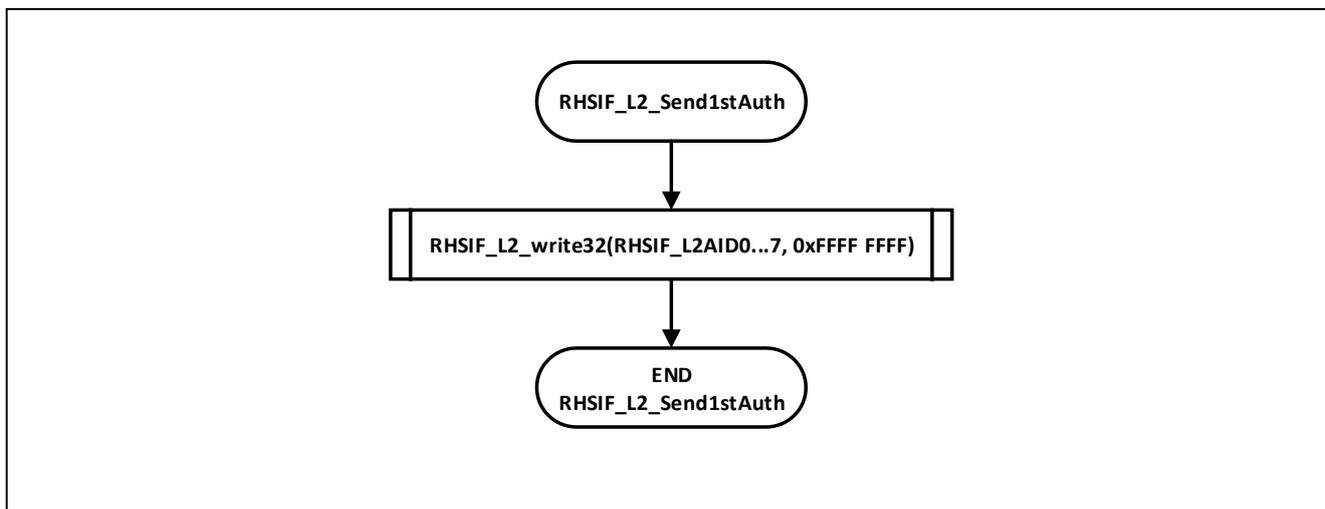


図 26 : 関数「RHSIF\_L2\_Send1stAuth」フローチャート

### 7.7.3 第1認証チェック

有効なキー ID ベースの第1レベル認証のチェックは、関数「**RHSIF\_L2\_wait1stAuth\_Check**」を使用してファイル「**RHSIF.c**」内で実行されます。

- #1. この関数は、RHSIF0\_L2MST.AUTS0 ビットのステータスをポーリングする while ループから始まります。このビットが「1」になると、有効な認証が検出され、変数「the\_state」がステータス RHSIF\_STATE\_AUTHENTICATED に変わります。

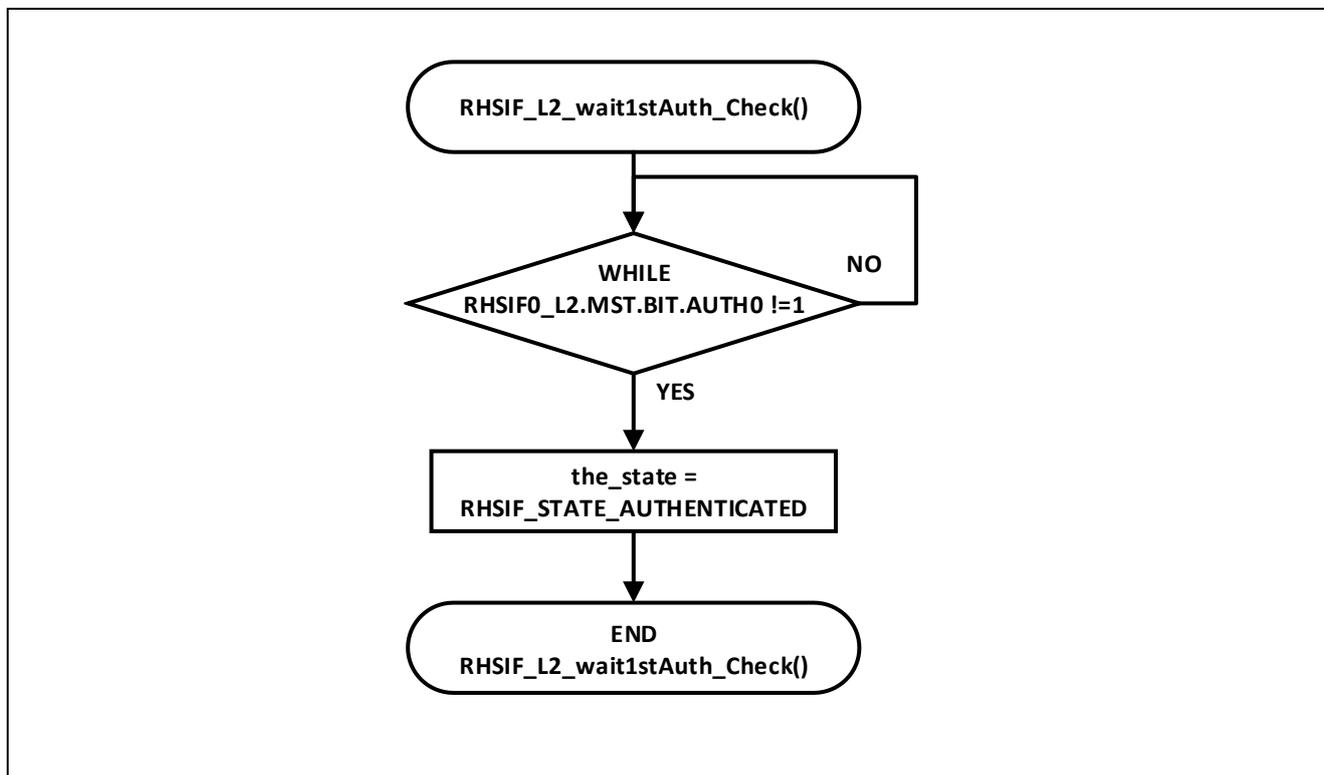


図 27 : 関数「RHSIF\_L2\_wait1stAuth\_Check」フローチャート

### 7.7.4 Read32 コマンド

読み取りコマンド動作は、関数「**RHSIF\_L2\_read32**」内のファイル「**RHSIF.c**」に示されます。

- #1. この関数は、インターフェイスチャンネル n 待機ビットの 4 つの異なるスイッチケースから始まります。この関数により、インターフェイスが新しいコマンドを受信する準備ができた場合にのみ、読み取りプロセスが開始されることが保証されます。従って、インターフェイスがビジー状態になってもデータが失われることはありません。
- #2. 次の最初の手順では、ウィンドウ番号が関数呼び出しとともに受信した値と照合されます。以下の手順は、ウィンドウ番号が「4」未満の場合にのみ実行されます。
  - a. ステータスレジスタがクリアされます。
  - b. その後アドレスが書き込まれます。
  - c. 次に、L2 Command\_read32 がトリガーされます。
- #3. ループは、チャンネル n ステータスレジスタの RDY ビットをチェックします。「1」に設定すると、リードコマンドがリンクパートナーに発行されます。
- #4. 最後に、IF ステートメントは、リンクパートナーからの読み取りデータがレジスタ内で利用可能かどうかをチェックします。

- ステートメントが true の場合、データはレジスタから読み取られ、「\*data」が指す変数に格納されます。そして関数は「1」を返します。
- それ以外の場合は「0」が返されます。

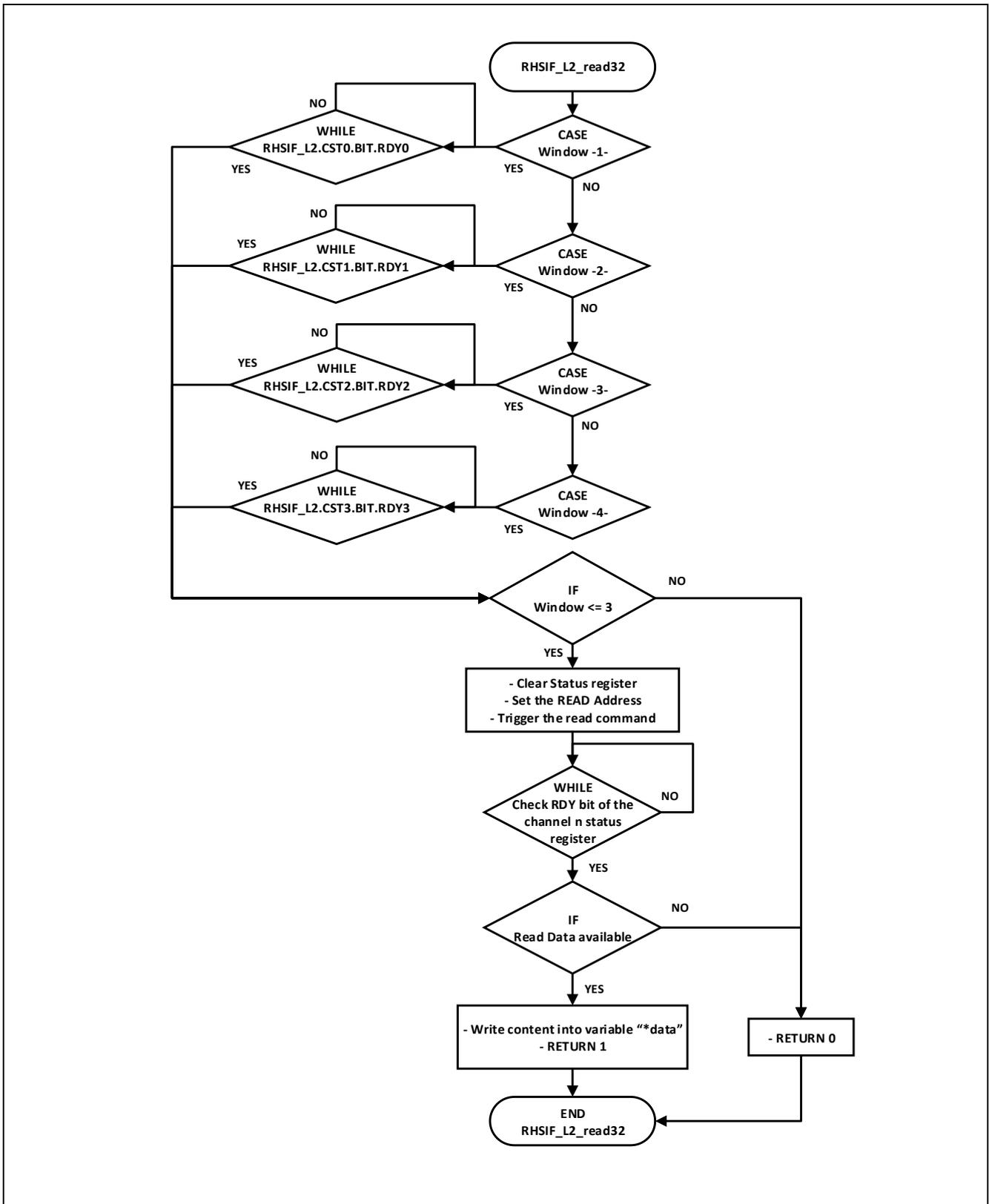


図 28 : 関数「RHSIF\_L2\_read32」フローチャート

### 7.7.5 Write32 コマンド

書き込みコマンド動作は、関数「**RHSIF\_L2\_write32**」内のファイル「**RHSIF.c**」に示されます。

#1. この関数はスイッチケースコマンドで実装されます。初めに、考えられる4つの異なるケース(ウィンドウ0からウィンドウ3)が定義されます。いずれの場合においても、同様の手順が実行されます。唯一の違いは、専用レジスタのアドレスです。このアドレスは、選択したウィンドウ番号に応じて、チャンネル0または3のRHSIFnCSTn.BIT.RDYnと一致する必要があります。

- while ループが実行され、RHSIF\_L2.CST0.BIT.RDY ビットが「0」であるかどうかをチェックされます。これが false の場合、L2 はリクエストの送信準備完了であり、以下の手順が実行されます。
- チャンネル n ステータスレジスタがクリアされます。
- 書き込まれデータが設定されます。
- 書き込みデータが準備されます。
- write32 の L2 コマンドが発行されます。

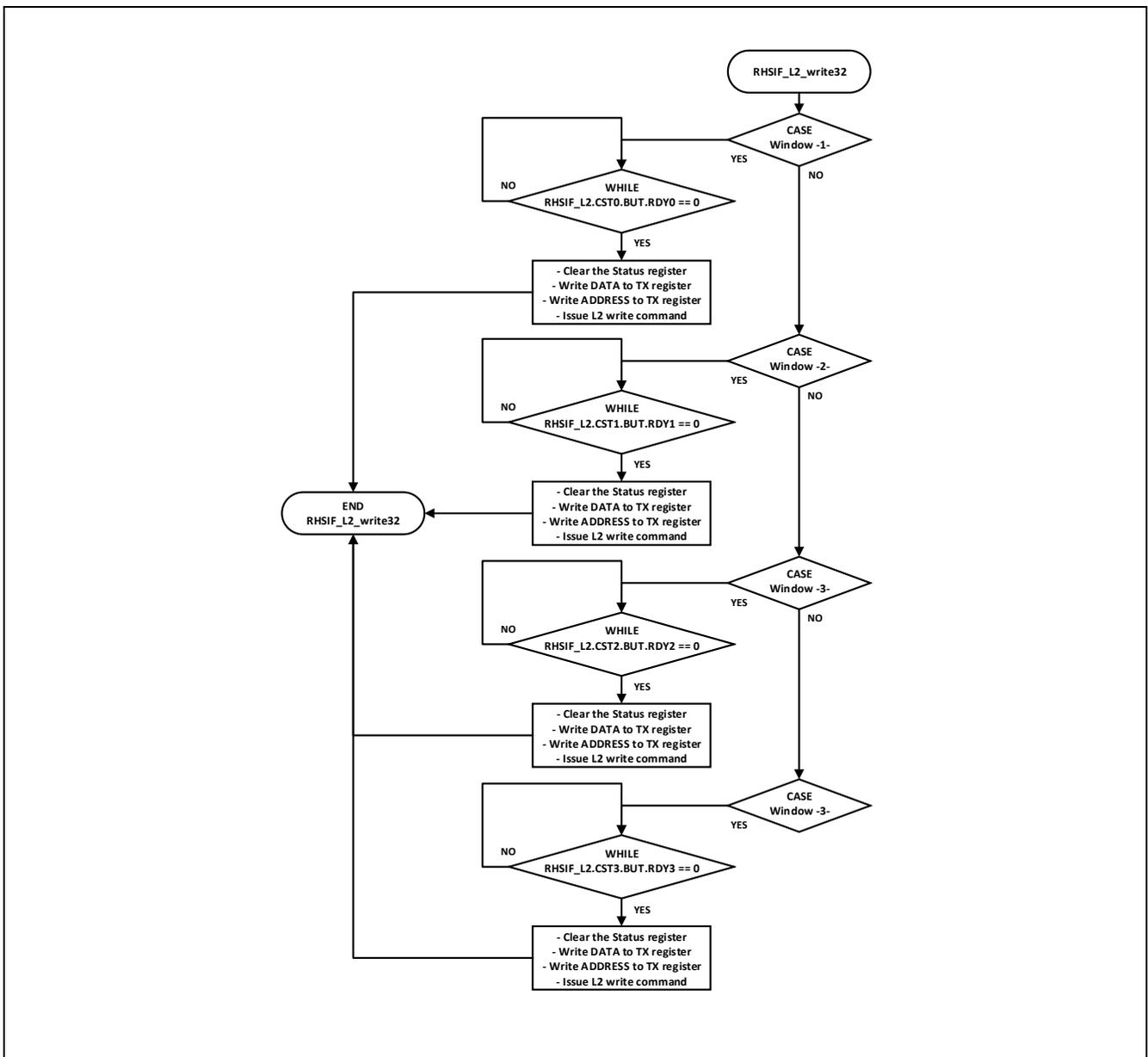


図 29 : 関数「RHSIF\_L2\_write32」フローチャート

### 7.7.6 Stream256 読み込みコマンド

読み取りコマンド動作は、関数「**RHSIF\_L2\_stream256\_read**」内のファイル「**RHSIF.c**」に示されます。

- #1. この関数はストリーム受信コマンドで実装されます。
- a. データ先領域アドレスを設定します。
  - b. ストリーム受信の L2 コマンドが発行されます。

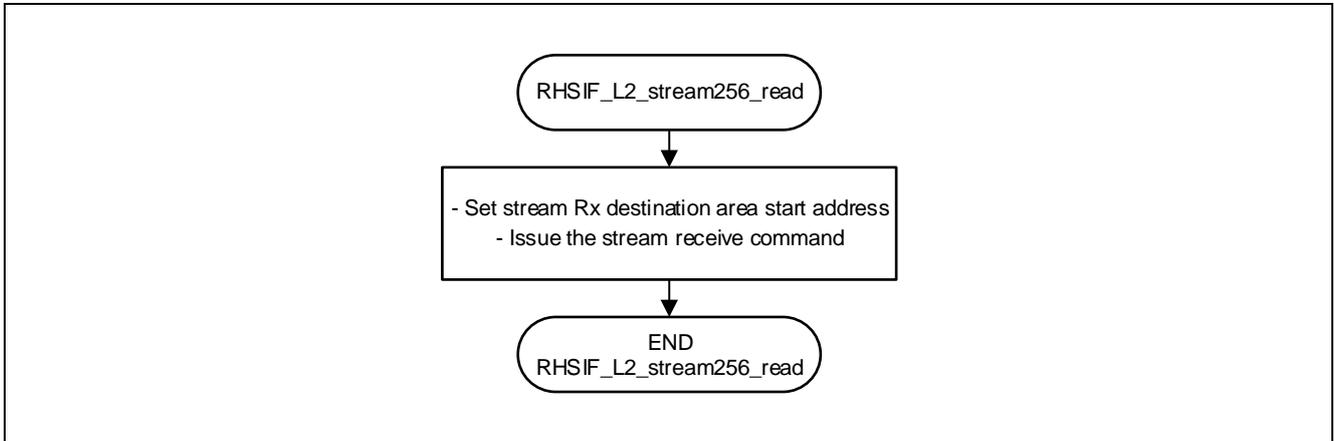


図 30 : 関数「RHSIF\_L2\_stream256\_read」フローチャート

### 7.7.7 Stream256 書き込みコマンド

書き込みコマンド動作は関数「**RHSIF\_L2\_stream256\_write**」内のファイル「**RHSIF.c**」に示されます。

- #1. この関数はストリーム送信コマンドで実装されます
- a. チャンネル 2 ステータスレジスタがクリアされます。
  - b. データ先領域アドレスを設定します。
  - c. チャンネル 2 書き込みアドレスが準備されます。
  - d. ストリーム送信の L2 コマンドが発行されます。

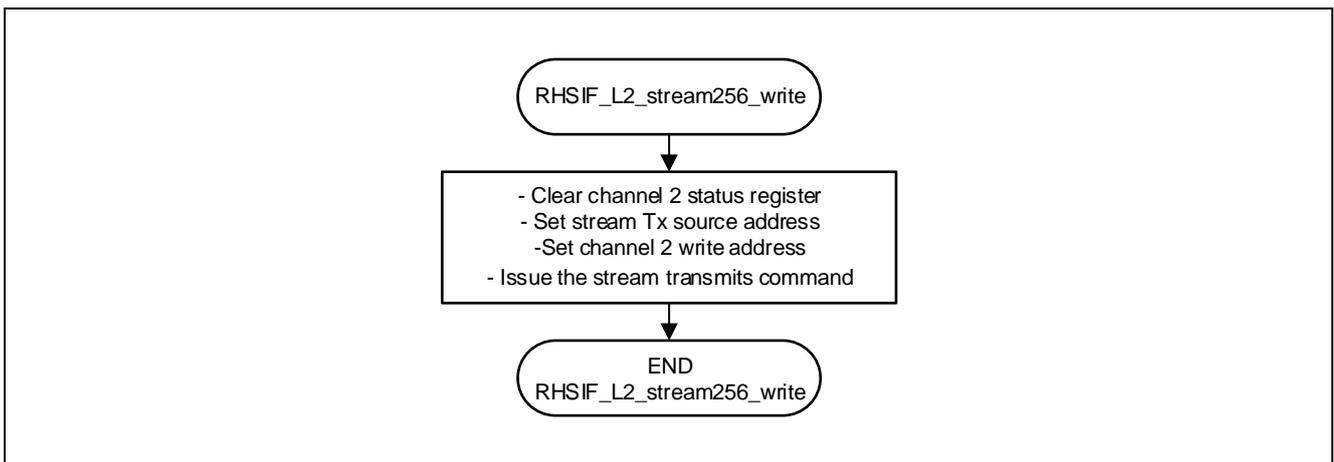


図 31 : 関数「RHSIF\_L2\_stream256\_read」フローチャート

## 7.8 送受信データ

### 7.8.1 非ストリーム送受信データ

データはそれぞれのターゲットノードに送信され、イニシエータによって読み戻されます。データ自体は 2 つの異なる配列に保存され、送受信が完了した後にデバッガで読み取ることが可能です。

- **L2\_Data\_Master[32]**  
— マスターデバイスから受信したデータを含む
- **L2\_Data\_Slave[32]**  
— スレーブデバイスから受信したデータを含む

ソフトウェアがエラーなしで実行される場合、2 つの変数には以下のデータが含まれます。

- **L2\_Data\_Master[32]**
  - [0] = 1
  - [1] = 2
  - [2] = 4
  - [3] = 8
  - [4] = 16
  - [5] = 32
  - [6] = 64
  - [7] = 128
- **L2\_Data\_Slave[32]**
  - [0] = 256
  - [1] = 512
  - [2] = 1024
  - [3] = 2048
  - [4] = 4096
  - [5] = 8192
  - [6] = 16384
  - [7] = 32768

### 7.8.2 ストリーム送受信データ

ストリームデータはマスターデバイスからスレーブデバイスに送信されます。

スレーブデバイスには送信するデータサイズの受信データ領域が必要です。

- **L2\_Data\_Master\_stream[65536]**  
マスターデバイスへのストリーム送信データを含む
- **L2\_Data\_Slave\_stream[65536]**  
マスターデバイスからのストリーム受信データを含む

## 8 略称一覧

表 1：略称一覧

| 略称    | 概説  |
|-------|---|
| MCU   | Micro-Controller-Unit                     |
| RHSIF | Renesas High-Speed Serial Interface       |
| L1    | The “Datalink Layer” sub module of RHSIF  |
| L2    | The “Transport Layer” sub module of RHSIF |
| PEID  | Processor Element ID                      |
| SPID  | Serial Programmer ID                      |
| ECM   | Error Correction Module                   |
| HBG   | H-Bus Guard                               |
| CRG   | Cluster RAM Guard                         |
| PBG   | Peripheral Bus Guard                      |
| ICUM  | Integrated-Cryptographic-Unit-Medium      |
| ICLC  | Interface Control Logical Channel         |
| CTS   | Clear To Send                             |
| RFP   | Renesas Flash Programmer                  |

## 9 付属デモソフトに関する注意事項

本アプリケーションノートの開発ソフトウェアは、編集・デバッグ環境「Greenhills MULTI」を使用して作成されています。プロジェクトデータは、RHSIFアプリケーションノートパッケージファイル内に.zip圧縮ファイルとして保存されています。使用するデバイスに応じて、デバイス上でソフトウェアを動作させるには、プロジェクトフォルダ内の関連レジスタ内のデバイス ファイルを編集する必要があります。

---

### 注意：

- プログラムされたソフトウェアはデモンストレーションのみを目的として設計されています。
  - その一部を独自のプロジェクトに実装する前に、EMC、機能安全、品質、セキュリティの目的に関するそれぞれの要件を検討してください。
-

## 10 ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://www.renesas.com/>

お問い合わせ先

<http://www.renesas.com/contact/>

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。

## 11 改訂記録

| Rev. | 発行日        | 改定内容 |      |
|------|------------|------|------|
|      |            | ページ  | ポイント |
| 1.00 | 2023.10.10 | All  | 初版発行 |

## 12 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しないでください。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、リセットを解除してください。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、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  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、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  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えば、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、ノイズ輻射量などが異なる場合があります。型名が異なる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）、交通管制（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た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したします。
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（豊洲フォレストシア）

[www.renesas.com](http://www.renesas.com)

## 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。

## 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/](http://www.renesas.com/contact/)