

# RX62N グループ、RX621 グループ

## ユーザブートモードによる内蔵フラッシュメモリ書き換え (スレーブ)

R01AN0184JJ0102  
Rev.1.02  
2012.02.29

### 要旨

本アプリケーションノートは、RX62N グループ、RX621 グループ アプリケーションノート「ユーザブートモードによる内蔵フラッシュメモリ書き換え (マスタ)」(R01AN0185JJ) からクロック同期式シリアル通信で送信される、消去する消去ブロック番号、書き込みデータのサイズ、および書き込みデータを使用してコード格納用フラッシュメモリ (ユーザマツト) の書き込み/消去する処理について説明しています。

クロック同期式シリアル通信で消去する消去ブロック番号、書き込みデータのサイズ、および書き込みデータを送信する処理に関しては、RX62N グループ、RX621 グループ アプリケーションノート「ユーザブートモードによる内蔵フラッシュメモリ書き換え (マスタ)」(R01AN0185JJ) をご参考ください。

### 対象デバイス

RX62N グループ、RX621 グループ

RX62N グループ、RX621 グループと同様の I/O レジスタ (周辺装置制御レジスタ) を持つ他の RX ファミリでも本プログラムを使用することができます。ただし、一部の機能を機能追加等で変更している場合がありますのでマニュアルで確認してください。このアプリケーションノートをご使用に際しては十分な評価を行ってください。

### 目次

|                   |    |
|-------------------|----|
| 1. 仕様 .....       | 2  |
| 2. 動作確認環境 .....   | 4  |
| 3. 使用機能 .....     | 4  |
| 4. 動作説明 .....     | 5  |
| 5. ソフトウェア説明 ..... | 20 |
| 6. 使用上の注意事項 ..... | 63 |
| 7. 参考ドキュメント ..... | 67 |

## 1. 仕様

- 本アプリケーションノートでは、RX62N グループの R5F562N8BDBG を用いて、ユーザブートモードでユーザマットの書き込み/消去を行います。
- スレーブは、マスタから消去ブロック番号、書き込みデータサイズ、および書き込みデータをクロック同期式シリアル通信で受信し、ユーザマットの書き込み/消去を行います。
- マスタとスレーブ間のクロック同期式シリアル通信は、SCI チャンネル 2 (SCI2) モジュールを使用します。
- クロック同期式シリアル通信仕様は、ビットレート 2.4Mbps、データビット 8 ビット、LSB ファーストとし、マスタから転送クロックを出力します。
- 本アプリケーションノートでは、スレーブは指定された消去ブロック (EB30: 16K バイト) の消去を行い、受信した 8K バイト (256 バイト × 32) の書き込みデータを消去ブロック EB30 の先頭アドレスから書き込みます。
- スレーブとマスタは通信制御するためにハンドシェイクしています。スレーブは I/O ポート (P01) を利用してビジー状態にアサート (Low)、ビジー解除時にネゲート (High) を出力します。マスタはスレーブからの出力を外部割り込み端子 (IRQ9-A) で受け、立ち上がりエッジが入力されることにより次の送信を開始します。
- スレーブは、正常にユーザマットの消去/書き込み処理が完了すると I/O ポートに接続された 4 個の LED で正常終了を知らせます。また、マスタとの通信中および書き込み/消去処理中にエラーが発生した場合は、LED でそのエラー状態を知らせます。

図 1 に本アプリケーションノートにおける仕様を示します。

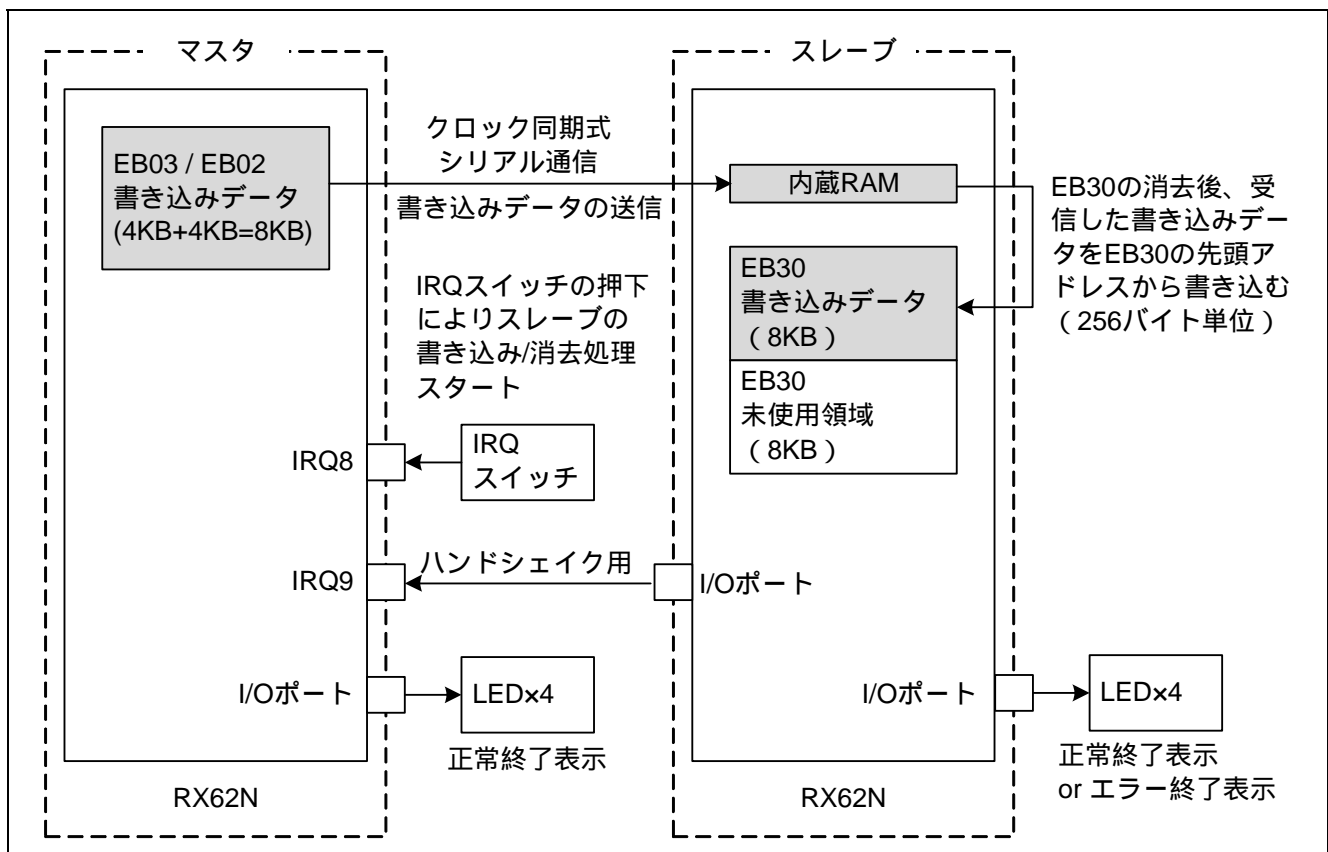


図 1 仕様

図 2 に本アプリケーションノートにおけるスレーブのハードウェア構成図を示します。

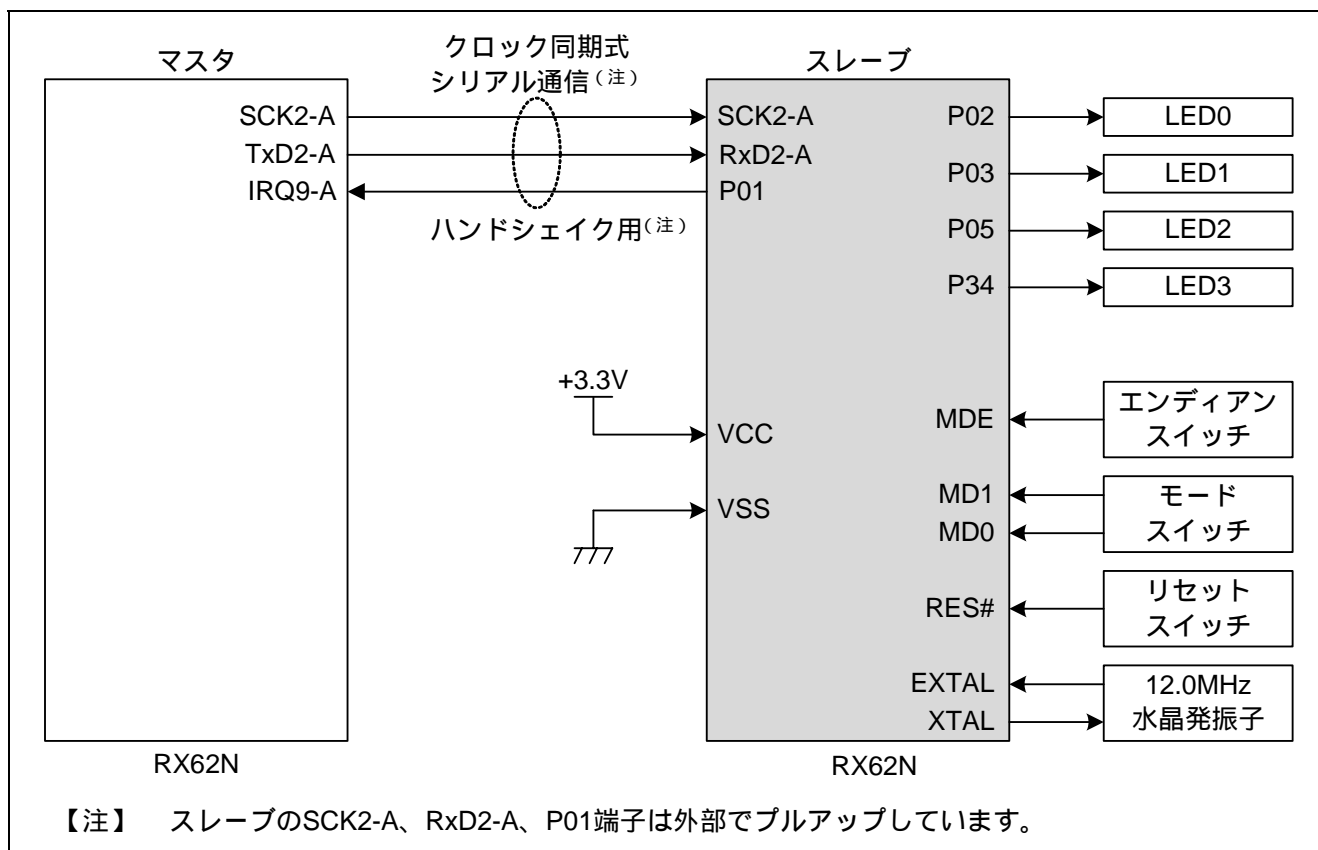


図 2 スレーブのハードウェア構成図

## 2. 動作確認環境

スレーブの動作確認を行った環境を表 1 に示します。

表 1 スレーブの動作確認環境

| 項目                                 | 内容   |
|------------------------------------|--|
| デバイス                               | RX62N グループ: R5F562N8BDBG<br>(ROM 容量: 512K バイト、RAM 容量: 96K バイト) |
| ボード                                | Renesas Starter Kit (R0K5562N0S000BE)                          |
| 電源電圧                               | 5.0V (CPU 動作電圧は 3.3V)  |
| 入力クロック                             | 12.0MHz (ICLK = 96MHz、PCLK = 48MHz、BCLK = 24MHz)               |
| 動作温度                               | 室温   |
| High-performance Embedded Workshop | Version 4.07.00.007  |
| Toolchain                          | RX Standard Toolchain (V.1.0.0.0)                              |
| FDT                                | V.4.06 Release 00  |

## 3. 使用機能

- クロック発生回路
- 低消費電力低減機能
- 割り込みコントローラ
- I/O ポート
- シリアルコミュニケーションインタフェース
- ROM (コード格納用フラッシュメモリ)

詳細は 7. 参考ドキュメント「ユーザーズマニュアル」を参照してください。

## 4. 動作説明

### 4.1 動作モードの設定

本アプリケーションノートでは、スレーブのモード端子を MD1 = 1、MD0 = 0 に設定し動作モードをユーザブートモードに、システムコントロールレジスタ 0 (SYSCR0) の ROME ビットを 1 に設定し内蔵 ROM を有効に、SYSCR0 レジスタの EXBE ビットを 0 に設定し外部バスを無効にそれぞれ設定します。

なお、RX62N のユーザブートマットには USB ブートモード (モード端子の設定はユーザブートモードと同じ) で起動するプログラムが格納されております。ユーザで用意したプログラムをユーザブートマットに書き込むには、フラッシュ書き込みツール (FDT 等) を使用してブートモードでユーザブートマットを消去後、書き込んでください。

スレーブはユーザブートモードでユーザブートマットから起動します。

表 2 に本アプリケーションノートにおけるスレーブの動作モードの設定を示します。

表 2 スレーブの動作モードの設定

| モード端子 |     | SYSCR0 レジスタ |      | 動作モード     | 内蔵 ROM | 外部バス |
|-------|-----|-------------|------|-----------|--------|------|
| MD1   | MD0 | ROME        | EXBE |           |        |      |
| 1     | 0   | 1           | 0    | ユーザブートモード | 有効     | 無効   |

【注】 SYSCR0 レジスタの ROME ビットおよび EXBE ビットの初期値は、SYSCR0.ROME = 1、SYSCR0.EXBE = 0 のため、プログラム中で SYSCR0 レジスタの設定は行っておりません。

### 4.2 クロックの設定

本アプリケーションノートで使用している評価ボードには 12.0MHz の水晶発振子が搭載されています。

そのため、本アプリケーションノートではシステムクロック (ICLK)、周辺モジュールクロック (PCLK)、および外部バスクロック (BCLK) をそれぞれ、8 通倍 (96MHz)、4 通倍 (48MHz)、2 通倍 (24MHz) に設定しています。

### 4.3 エンディアンの設定

本アプリケーションノートは、ビッグエンディアン/リトルエンディアンの両方に対応しています。ハードウェア (MDE 端子) によるエンディアンの設定を表 3 に示します。なお、マスタとスレーブのエンディアンは合わせてください。

表 3 エンディアン設定 (ハードウェア)

| MDE 端子 | エンディアン    |
|--------|-----------|
| 0      | リトルエンディアン |
| 1      | ビッグエンディアン |

コンパイラオプションによるエンディアンの設定を表 4 に示します。

表 4 エンディアン設定 (コンパイラオプション)

| マイコンオプション       | エンディアン    |
|-----------------|-----------|
| endian = little | リトルエンディアン |
| endian = big    | ビッグエンディアン |

【注】 コンパイラオプションで選択したエンディアンに合わせて、MDE 端子によるエンディアンを設定してください。

#### 4.4 クロック同期式シリアル通信仕様

本アプリケーションノートでは、マスタとスレーブ間でクロック同期式シリアル通信により、通信コマンド、消去ブロック番号、書き込みデータサイズ、書き込みデータの受信を行います。転送クロックはマスタが出力します。使用する SCI2 の SCK2-A、RxD2-A 端子はそれぞれ外部でプルアップしています。

表 5 にクロック同期式シリアル通信仕様を示します。

表 5 クロック同期式シリアル通信仕様

| 項目           | 仕様                       |
|--------------|--------------------------|
| チャンネル        | SCI チャンネル 2 (SCI2)       |
| コミュニケーションモード | クロック同期式モード               |
| ビットレート       | 2.4Mbps (PCLK = 48MHz 時) |
| データ転送方向      | LSB ファースト                |
| エラー          | オーバランエラー                 |

##### 4.4.1 通信コマンド仕様

マスタとスレーブ間の通信コマンドの仕様を表 6 に示します。

表 6 通信コマンド仕様

| コマンド   | 値   | 説明                                | 通信方向       |
|--------|-----|-----------------------------------|------------|
| FSTART | 10h | スレーブのユーザマットの書き込み/消去処理を開始するためのコマンド | マスタ → スレーブ |
| ERASE  | 11h | スレーブのユーザマットの消去を開始するためのコマンド        | マスタ → スレーブ |
| WRITE  | 12h | スレーブのユーザマットの書き込みを開始するためのコマンド      | マスタ → スレーブ |

4.4.2 通信フロー

マスタとスレーブ間の通信フローを図3～図6に示します。

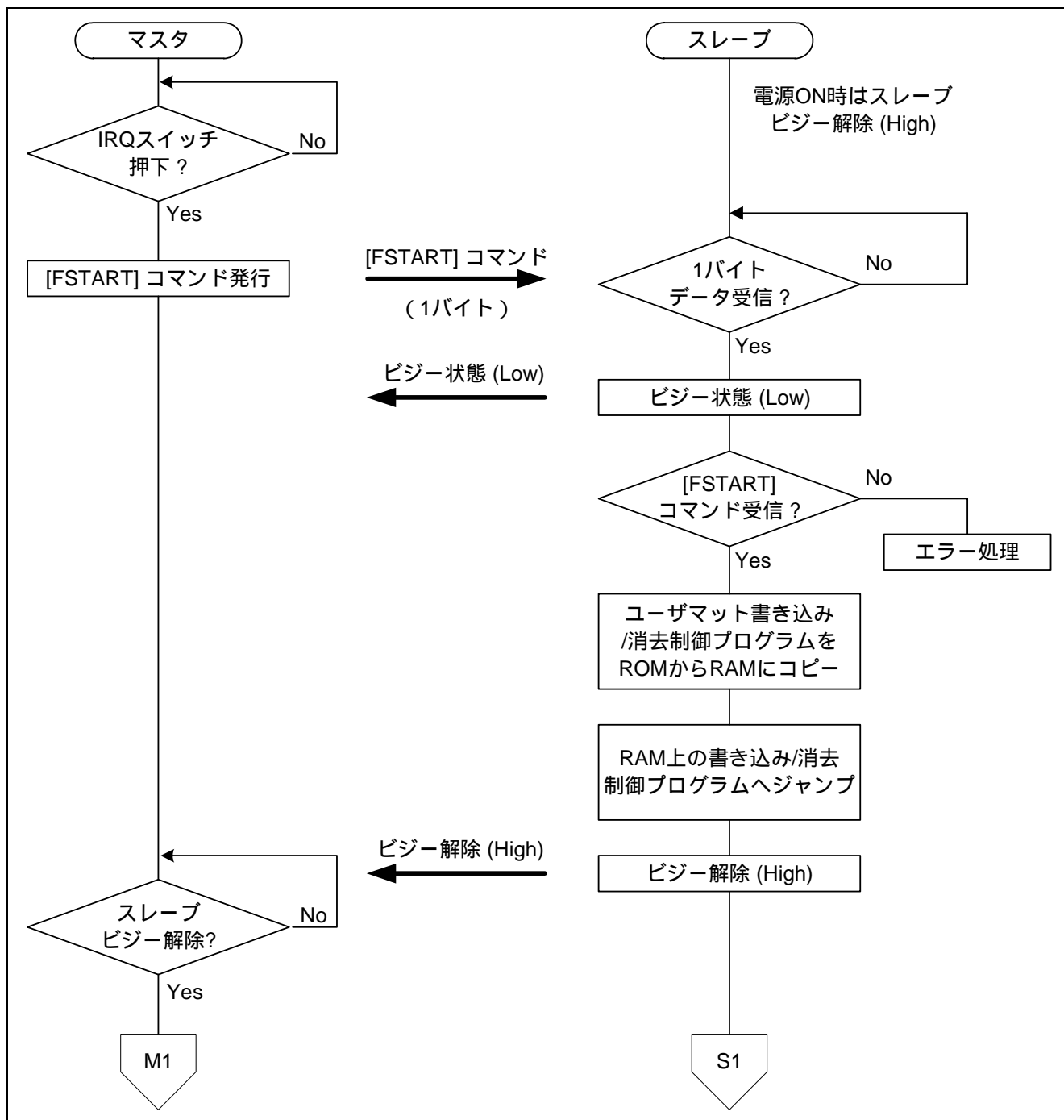


図3 通信フロー (1)

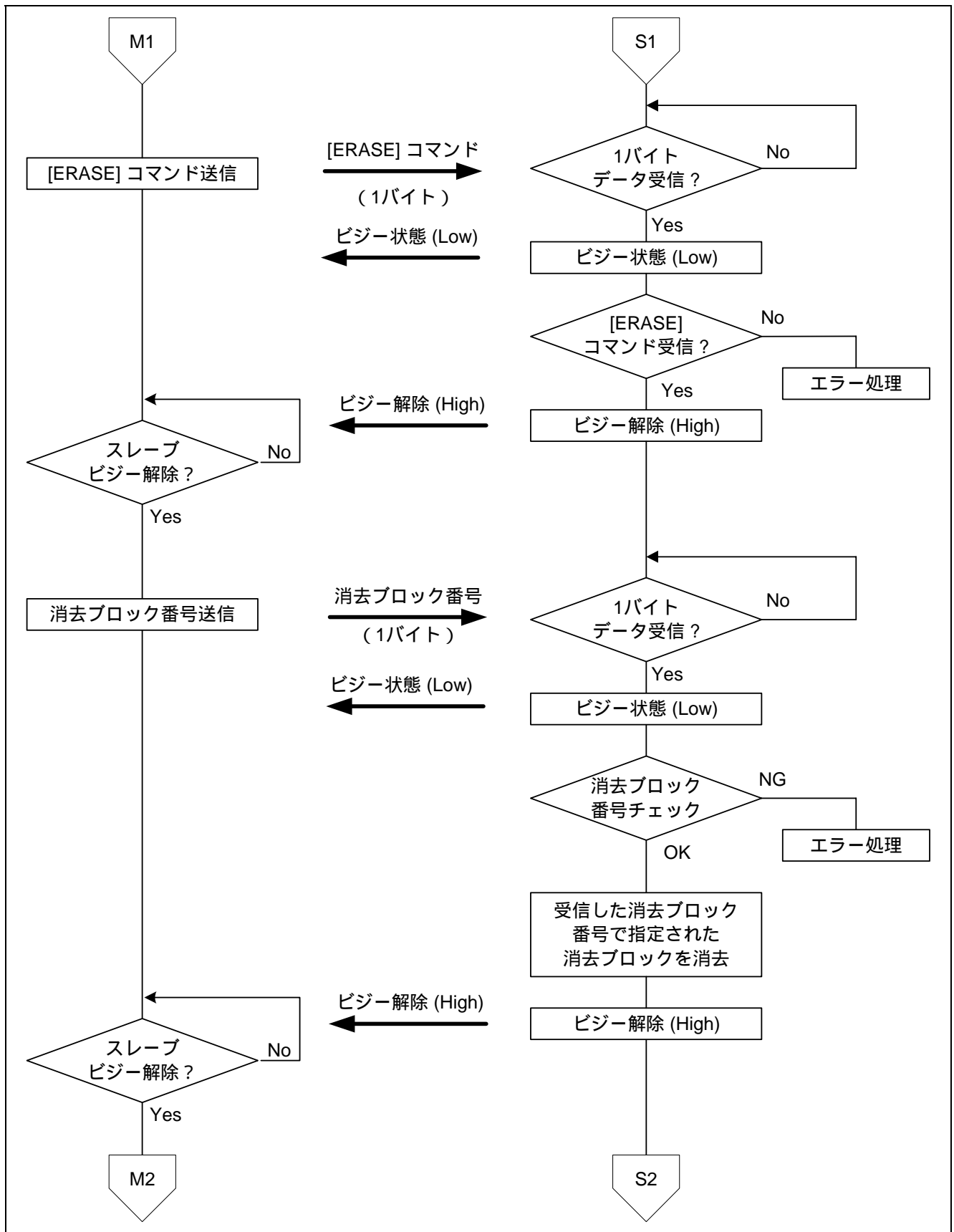


図4 通信フロー (2)



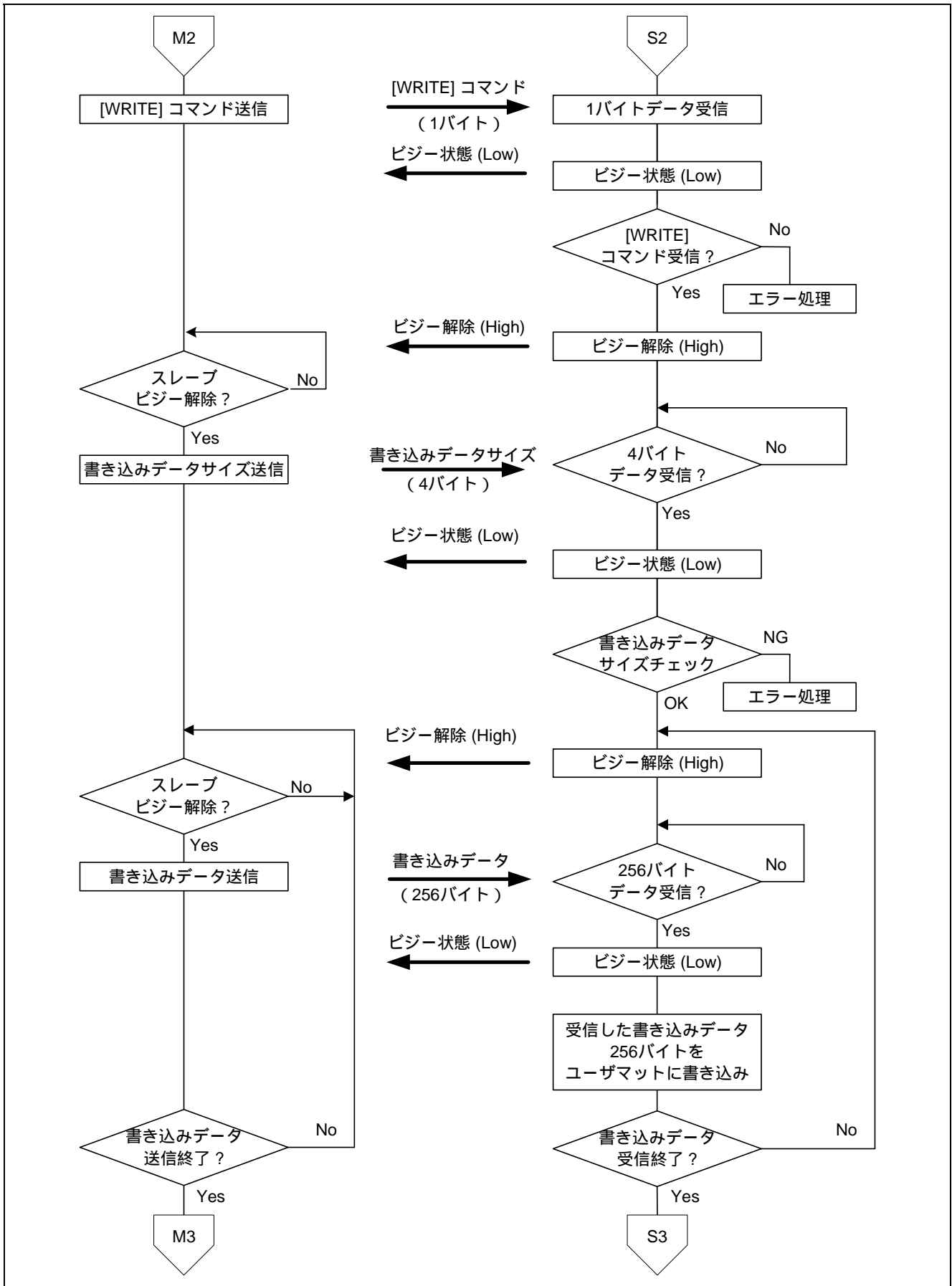


図5 通信フロー (3)

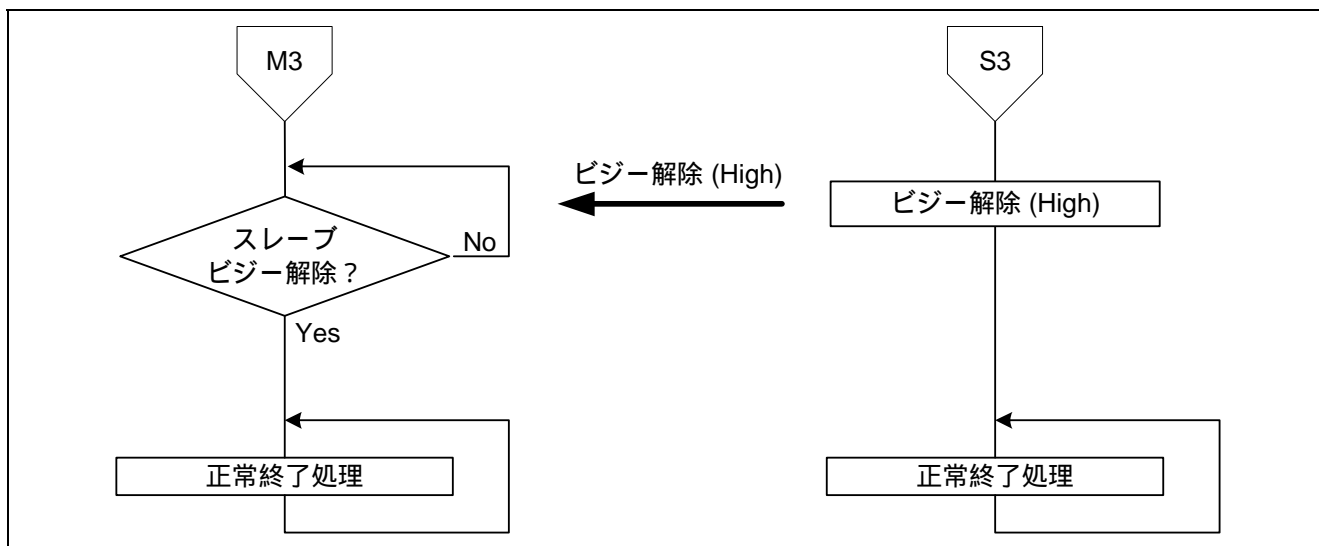


図 6 通信フロー (4)

#### 4.4.3 消去ブロック番号

スレーブは、マスタから[ERASE]コマンド受信後に1バイトの消去ブロック番号 (記号定数で定義された1バイトのデータ) を受信します。表7に消去ブロック番号の一覧を示します。また、図7に消去ブロック番号の仕様を示します。

表7 消去ブロック番号一覧

| 消去ブロック番号   |     | 内容                             |
|------------|-----|--------------------------------|
| 記号定数名      | 値   |                                |
| EB37_INDEX | 00h | 消去ブロック EB37 を指定 (サイズ: 16K バイト) |
| EB36_INDEX | 01h | 消去ブロック EB36 を指定 (サイズ: 16K バイト) |
| EB35_INDEX | 02h | 消去ブロック EB35 を指定 (サイズ: 16K バイト) |
| EB34_INDEX | 03h | 消去ブロック EB34 を指定 (サイズ: 16K バイト) |
| EB33_INDEX | 04h | 消去ブロック EB33 を指定 (サイズ: 16K バイト) |
| EB32_INDEX | 05h | 消去ブロック EB32 を指定 (サイズ: 16K バイト) |
| EB31_INDEX | 06h | 消去ブロック EB31 を指定 (サイズ: 16K バイト) |
| EB30_INDEX | 07h | 消去ブロック EB30 を指定 (サイズ: 16K バイト) |
| EB29_INDEX | 08h | 消去ブロック EB29 を指定 (サイズ: 16K バイト) |
| EB28_INDEX | 09h | 消去ブロック EB28 を指定 (サイズ: 16K バイト) |
| EB27_INDEX | 0Ah | 消去ブロック EB27 を指定 (サイズ: 16K バイト) |
| EB26_INDEX | 0Bh | 消去ブロック EB26 を指定 (サイズ: 16K バイト) |
| EB25_INDEX | 0Ch | 消去ブロック EB25 を指定 (サイズ: 16K バイト) |
| EB24_INDEX | 0Dh | 消去ブロック EB24 を指定 (サイズ: 16K バイト) |
| EB23_INDEX | 0Eh | 消去ブロック EB23 を指定 (サイズ: 16K バイト) |
| EB22_INDEX | 0Fh | 消去ブロック EB22 を指定 (サイズ: 16K バイト) |
| EB21_INDEX | 10h | 消去ブロック EB21 を指定 (サイズ: 16K バイト) |
| EB20_INDEX | 11h | 消去ブロック EB20 を指定 (サイズ: 16K バイト) |
| EB19_INDEX | 12h | 消去ブロック EB19 を指定 (サイズ: 16K バイト) |
| EB18_INDEX | 13h | 消去ブロック EB18 を指定 (サイズ: 16K バイト) |
| EB17_INDEX | 14h | 消去ブロック EB17 を指定 (サイズ: 16K バイト) |
| EB16_INDEX | 15h | 消去ブロック EB16 を指定 (サイズ: 16K バイト) |
| EB15_INDEX | 16h | 消去ブロック EB15 を指定 (サイズ: 16K バイト) |
| EB14_INDEX | 17h | 消去ブロック EB14 を指定 (サイズ: 16K バイト) |
| EB13_INDEX | 18h | 消去ブロック EB13 を指定 (サイズ: 16K バイト) |
| EB12_INDEX | 19h | 消去ブロック EB12 を指定 (サイズ: 16K バイト) |
| EB11_INDEX | 1Ah | 消去ブロック EB11 を指定 (サイズ: 16K バイト) |
| EB10_INDEX | 1Bh | 消去ブロック EB10 を指定 (サイズ: 16K バイト) |
| EB09_INDEX | 1Ch | 消去ブロック EB09 を指定 (サイズ: 16K バイト) |
| EB08_INDEX | 1Dh | 消去ブロック EB08 を指定 (サイズ: 16K バイト) |
| EB07_INDEX | 1Eh | 消去ブロック EB07 を指定 (サイズ: 4K バイト)  |
| EB06_INDEX | 1Fh | 消去ブロック EB06 を指定 (サイズ: 4K バイト)  |
| EB05_INDEX | 20h | 消去ブロック EB05 を指定 (サイズ: 4K バイト)  |
| EB04_INDEX | 21h | 消去ブロック EB04 を指定 (サイズ: 4K バイト)  |
| EB03_INDEX | 22h | 消去ブロック EB03 を指定 (サイズ: 4K バイト)  |
| EB02_INDEX | 23h | 消去ブロック EB02 を指定 (サイズ: 4K バイト)  |
| EB01_INDEX | 24h | 消去ブロック EB01 を指定 (サイズ: 4K バイト)  |
| EB00_INDEX | 25h | 消去ブロック EB00 を指定 (サイズ: 4K バイト)  |

消去ブロックデータ (unsigned char型)

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| b7  | b6  | b5  | b4  | b3  | b2  | b1  | b0  |
| BD7 | BD6 | BD5 | BD4 | BD3 | BD2 | BD1 | BD0 |

本アプリケーションノートでは、スレーブの消去ブロックEB30の書き込み/消去を行うため、消去ブロックデータを [EB30\_INDEX (07h)] としています。

【注】 消去ブロック番号は、表7に示した [EB37\_INDEX (00h)] ~ [EB00\_INDEX (25h)] の値を指定してください。消去ブロック番号を [26h] ~ [FFh] に指定した場合、スレーブはエラーと判定し、エラー処理を行います。

図7 消去ブロック番号仕様

4.4.4 書き込みデータサイズ

スレーブは、マスタから[WRITE]コマンド受信後に4バイトの書き込みデータサイズを受信します。図8に書き込みデータサイズの仕様を示します。

書き込みデータサイズ (unsigned long型)

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| b31  | b30  | b29  | b28  | b27  | b26  | b25  | b24  |
| SZ31 | SZ30 | SZ29 | SZ28 | SZ27 | SZ26 | SZ25 | SZ24 |
| b23  | b22  | b21  | b20  | b19  | b18  | b17  | b16  |
| SZ23 | SZ22 | SZ21 | SZ20 | SZ19 | SZ18 | SZ17 | SZ16 |
| b15  | b14  | b13  | b12  | b11  | b10  | b9   | b8   |
| SZ15 | SZ14 | SZ13 | SZ12 | SZ11 | SZ10 | SZ09 | SZ08 |
| b7   | b6   | b5   | b4   | b3   | b2   | b1   | b0   |
| SZ07 | SZ06 | SZ05 | SZ04 | SZ03 | SZ02 | SZ01 | SZ00 |

本アプリケーションノートでは、書き込みサイズを8Kバイトにしているため、書き込みデータサイズを [0000 2000h] としています。

【注1】 書き込みデータサイズは、0より大きい値かつ消去ブロック番号で指定した消去ブロックサイズ以下としてください。0の場合もしくは消去ブロック番号で指定した消去ブロックより大きいサイズを指定した場合、スレーブはエラーと判定し、エラー処理を行います。

【注2】 書き込みデータの送信は256バイト固定としています。したがって、書き込みデータサイズが256バイトの倍数でない場合、マスタは256バイトごとに送信を行っていき、最後の256バイトに満たない書き込みデータに関してはFFhを追加して256バイトの書き込みデータとしてスレーブに送信します。

図8 書き込みデータサイズ仕様

#### 4.4.5 オーバランエラー

本アプリケーションノートでは、スレーブのクロック同期式シリアル通信の受信時にオーバランエラーが発生 (SCI2.SSR.ORER ビットが1にセット) した場合には、エラー処理します。

#### 4.5 正常終了処理

スレーブはユーザマットの書き込み/消去が正常に終了すると、I/Oポートに接続されている4個のLEDに正常終了の表示を行います。正常終了の表示は、LED0~LED3を順番に1つつ点灯する処理を繰り返します。

#### 4.6 エラー処理

本アプリケーションノートにおけるスレーブのエラー一覧を表8に示します。スレーブのエラー処理では、エラー状態を接続されている4個のLEDに表示する処理を行います。

表8 スレーブのエラー一覧

: 点灯、 : 消灯

| エラー番号    | 内容   | LED表示 |      |      |      |
|----------|--|-------|------|------|------|
|          |  | LED3  | LED2 | LED1 | LED0 |
| エラーNo.01 | オーバランエラーが発生した場合  |       |      |      |      |
| エラーNo.02 | [FSTART]コマンド受信待ち状態でマスタから受信したコマンドが[FSTART]コマンドでなかった場合                                     |       |      |      |      |
| エラーNo.03 | [ERASE]コマンド受信待ち状態でマスタから受信したコマンドが[ERASE]コマンドでなかった場合                                       |       |      |      |      |
| エラーNo.04 | マスタから受信した消去ブロック番号が、EB00~EB37でなかった場合  |       |      |      |      |
| エラーNo.05 | FCUファームウェア転送前、ROMリードモードへ遷移する際にタイムアウト (tE16K×1.1) になった場合                                  |       |      |      |      |
| エラーNo.06 | 周辺クロック通知コマンド発行前、ROM P/Eモードに遷移する際に ILGLERR、ERSERR、PRGERR、および FCUERR ビットのいずれかが"1"にセットされた場合 |       |      |      |      |
| エラーNo.07 | 周辺クロック通知コマンド発行時にタイムアウト (tPCKA) になった場合、もしくは ILGLERR ビットが"1"にセットされた場合                      |       |      |      |      |
| エラーNo.08 | 消去ブロックの消去時にタイムアウト (tE16K×1.1) になった場合、もしくは ILGLERR、または ERSERR ビットのいずれかが"1"にセットされた場合       |       |      |      |      |
| エラーNo.09 | [WRITE]コマンド受信待ち状態でマスタから受信したコマンドが[WRITE]コマンドでなかった場合                                       |       |      |      |      |
| エラーNo.10 | マスタから受信した書き込みデータサイズが、0もしくは消去ブロック番号で指定されたブロックサイズより大きい場合                                   |       |      |      |      |
| エラーNo.11 | データ書き込み時にタイムアウト (tP256×1.1) になった場合、もしくは ILGLERR、または PRGERR ビットのいずれかが"1"にセットされた場合         |       |      |      |      |
| エラーNo.12 | データ書き込み終了後、ROMリードモードへ遷移する際にタイムアウト (tE16K×1.1) になった場合                                     |       |      |      |      |

## 4.7 LED 接続

スレーブの I/O ポートに接続されている LED0 ~ LED3 の接続図を図 9 に示します。

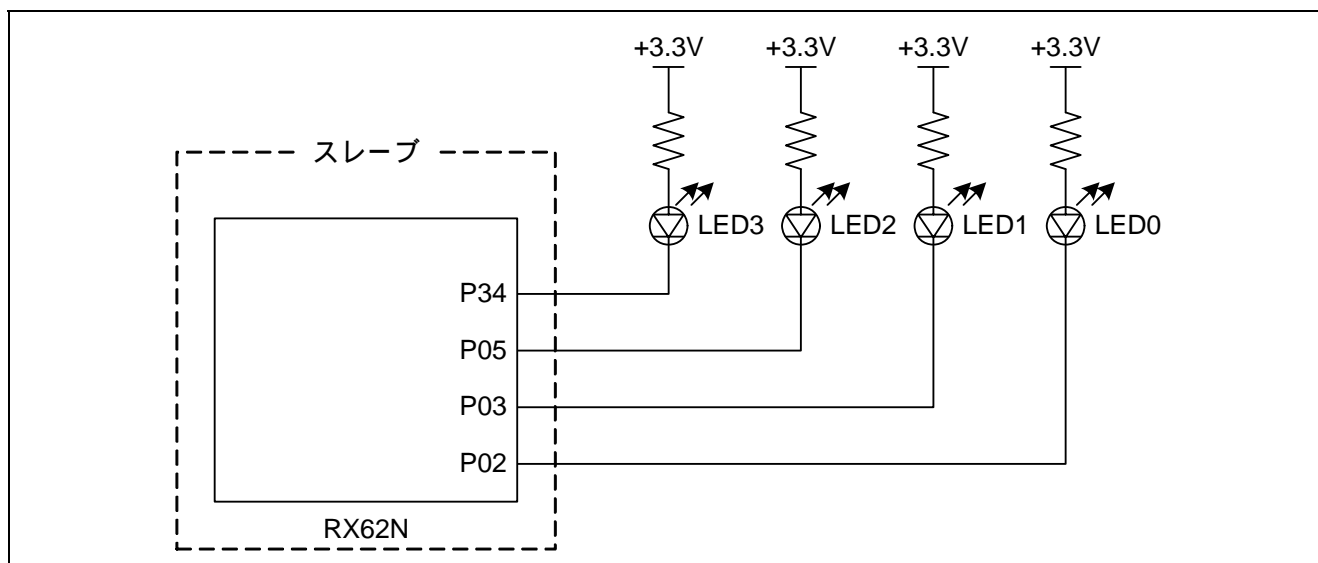


図 9 スレーブの LED 接続図

図 9 に示すように、I/O ポート (P02、P03、P05 および P34) から "High" を出力すると LED0 ~ LED3 は消灯、"Low" を出力すると LED0 ~ LED3 は点灯します。表 9 に I/O ポート出力と LED の状態を示します。

表 9 スレーブの I/O ポート出力と LED の状態

| I/O ポート | レジスタ設定                           | I/O ポート状態 | LED 状態 |    |
|---------|----------------------------------|-----------|--------|----|
| P02     | PORT0.DR.B2 = 1、PORT0.DDR.B2 = 1 | "High"出力  | LED0   | 消灯 |
|         | PORT0.DR.B2 = 0、PORT0.DDR.B2 = 1 | "Low"出力   |        | 点灯 |
| P03     | PORT0.DR.B3 = 1、PORT0.DDR.B3 = 1 | "High"出力  | LED1   | 消灯 |
|         | PORT0.DR.B3 = 0、PORT0.DDR.B3 = 1 | "Low"出力   |        | 点灯 |
| P05     | PORT0.DR.B5 = 1、PORT0.DDR.B5 = 1 | "High"出力  | LED2   | 消灯 |
|         | PORT0.DR.B5 = 0、PORT0.DDR.B5 = 1 | "Low"出力   |        | 点灯 |
| P34     | PORT3.DR.B4 = 1、PORT3.DDR.B4 = 1 | "High"出力  | LED3   | 消灯 |
|         | PORT3.DR.B4 = 0、PORT3.DDR.B4 = 1 | "Low"出力   |        | 点灯 |

## 4.8 ハンドシェイク制御

スレーブは通信制御するためにマスタとハンドシェイクしており、ハンドシェイク用の信号を Busy ポート (P01) から出力しています。

ハンドシェイク制御としてスレーブはマスタからのシリアル通信受信後に Busy ポートをビジー状態にアサート (Low) します。なお、次のシリアル通信が受信可能になってから Busy ポートをビジー解除にネゲート (High) します。表 10 にスレーブの Busy ポート出力と状態を示します。

表 10 スレーブの Busy ポート出力

| I/O ポート | レジスタ設定                           | I/O ポート状態 | Busy ポート |       |
|---------|----------------------------------|-----------|----------|-------|
| P01     | PORT0.DR.B1 = 1、PORT0.DDR.B1 = 1 | "High"出力  | ネゲート     | ビジー解除 |
|         | PORT0.DR.B1 = 0、PORT0.DDR.B1 = 1 | "Low"出力   | アサート     | ビジー状態 |

## 4.9 ユーザマットの書き込み/消去

本アプリケーションノートにおけるユーザマットの書き込み/消去について、以下に説明します。詳細については7. 参考ドキュメント「ユーザーズマニュアル」を参照してください。

### 4.9.1 RX62N グループ (R5F562N8) のメモリマット構成

R5F562N8 のコード格納用フラッシュメモリは、512K バイトのユーザマットと 16K バイトのユーザブートマットから構成されています。図 10 に R5F562N8 のユーザマットとユーザブートマットのアドレスを示します。

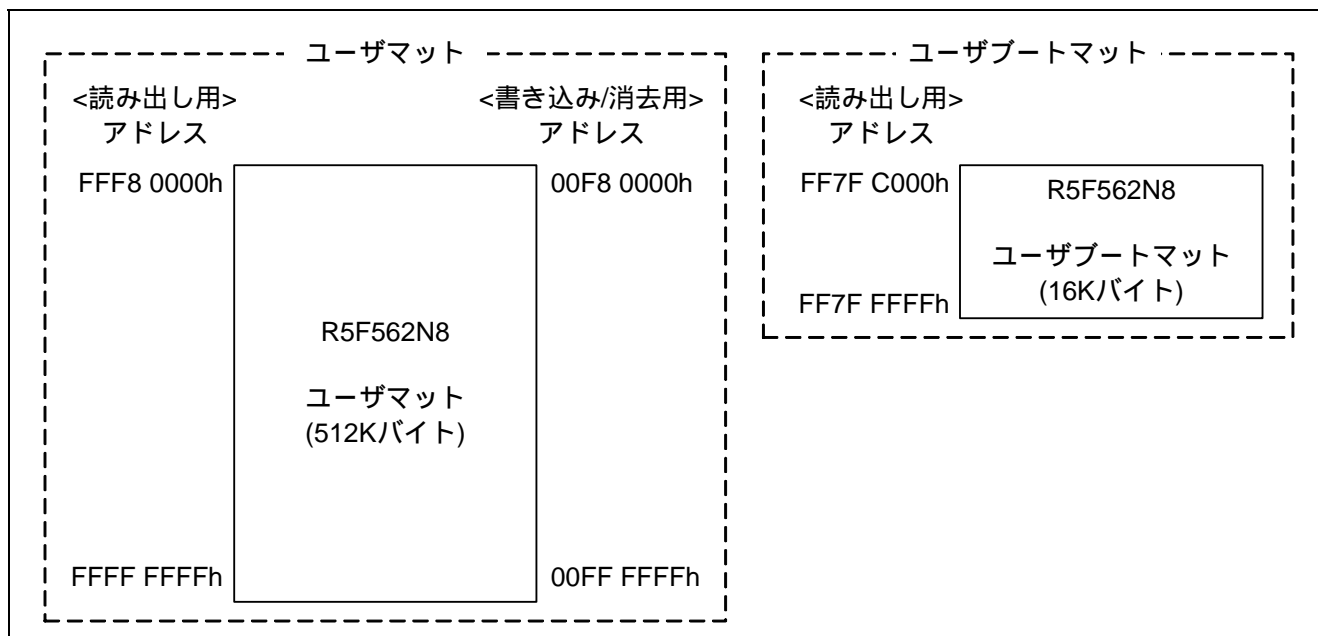


図 10 R5F562N8 のメモリマット構成

#### 4.9.2 RX62N グループ (R5F562N8) の消去ブロック構成

R5F562N8 のユーザマットは 16K バイト (30 ブロック)、4K バイト (8 ブロック) に分割されており、消去はこのブロック単位で行います。

また、ユーザマットへの書き込みは、下位アドレスが 00h で始まる 256 バイト単位で行います。

表 11 に R5F562N8 のユーザマットの消去ブロック構成を示します。

表 11 R5F562N8 の消去ブロック構成

| 消去ブロック | 読み出し用      |            | 書き込み/消去用   |            | サイズ<br>(バイト) |
|--------|------------|------------|------------|------------|--------------|
|        | 開始アドレス     | 終了アドレス     | 開始アドレス     | 終了アドレス     |              |
| EB37   | FFF8 0000h | FFF8 3FFFh | 00F8 0000h | 00F8 3FFFh | 16K          |
| EB36   | FFF8 4000h | FFF8 7FFFh | 00F8 4000h | 00F8 7FFFh | 16K          |
| EB35   | FFF8 8000h | FFF8 BFFFh | 00F8 8000h | 00F8 BFFFh | 16K          |
| EB34   | FFF8 C000h | FFF8 FFFFh | 00F8 C000h | 00F8 FFFFh | 16K          |
| EB33   | FFF9 0000h | FFF9 3FFFh | 00F9 0000h | 00F9 3FFFh | 16K          |
| EB32   | FFF9 4000h | FFF9 7FFFh | 00F9 4000h | 00F9 7FFFh | 16K          |
| EB31   | FFF9 8000h | FFF9 BFFFh | 00F9 8000h | 00F9 BFFFh | 16K          |
| EB30   | FFF9 C000h | FFF9 FFFFh | 00F9 C000h | 00F9 FFFFh | 16K          |
| EB29   | FFFA 0000h | FFFA 3FFFh | 00FA 0000h | 00FA 3FFFh | 16K          |
| EB28   | FFFA 4000h | FFFA 7FFFh | 00FA 4000h | 00FA 7FFFh | 16K          |
| EB27   | FFFA 8000h | FFFA BFFFh | 00FA 8000h | 00FA BFFFh | 16K          |
| EB26   | FFFA C000h | FFFA FFFFh | 00FA C000h | 00FA FFFFh | 16K          |
| EB25   | FFFB 0000h | FFFB 3FFFh | 00FB 0000h | 00FB 3FFFh | 16K          |
| EB24   | FFFB 4000h | FFFB 7FFFh | 00FB 4000h | 00FB 7FFFh | 16K          |
| EB23   | FFFB 8000h | FFFB BFFFh | 00FB 8000h | 00FB BFFFh | 16K          |
| EB22   | FFFB C000h | FFFB FFFFh | 00FB C000h | 00FB FFFFh | 16K          |
| EB21   | FFFC 0000h | FFFC 3FFFh | 00FC 0000h | 00FC3FFFh  | 16K          |
| EB20   | FFFC 4000h | FFFC 7FFFh | 00FC 4000h | 00FC 7FFFh | 16K          |
| EB19   | FFFC 8000h | FFFC BFFFh | 00FC 8000h | 00FC BFFFh | 16K          |
| EB18   | FFFC C000h | FFFC FFFFh | 00FC C000h | 00FC FFFFh | 16K          |
| EB17   | FFFD 0000h | FFFD 3FFFh | 00FD 0000h | 00FD 3FFFh | 16K          |
| EB16   | FFFD 4000h | FFFD 7FFFh | 00FD 4000h | 00FD 7FFFh | 16K          |
| EB15   | FFFD 8000h | FFFD BFFFh | 00FD 8000h | 00FD BFFFh | 16K          |
| EB14   | FFFD C000h | FFFD FFFFh | 00FD C000h | 00FD FFFFh | 16K          |
| EB13   | FFFE 0000h | FFFE 3FFFh | 00FE 0000h | 00FE 3FFFh | 16K          |
| EB12   | FFFE 4000h | FFFE 7FFFh | 00FE 4000h | 00FE 7FFFh | 16K          |
| EB11   | FFFE 8000h | FFFE BFFFh | 00FE 8000h | 00FE BFFFh | 16K          |
| EB10   | FFFE C000h | FFFE FFFFh | 00FE C000h | 00FE FFFFh | 16K          |
| EB09   | FFFF 0000h | FFFF 3FFFh | 00FF 0000h | 00FF 3FFFh | 16K          |
| EB08   | FFFF 4000h | FFFF 7FFFh | 00FF 4000h | 00FF 7FFFh | 16K          |
| EB07   | FFFF 8000h | FFFF 8FFFh | 00FF 8000h | 00FF 8FFFh | 4K           |
| EB06   | FFFF 9000h | FFFF 9FFFh | 00FF 9000h | 00FF 9FFFh | 4K           |
| EB05   | FFFF A000h | FFFF AFFFh | 00FF A000h | 00FF AFFFh | 4K           |
| EB04   | FFFF B000h | FFFF BFFFh | 00FF B000h | 00FF BFFFh | 4K           |
| EB03   | FFFF C000h | FFFF CFFFh | 00FF C000h | 00FF CFFFh | 4K           |
| EB02   | FFFF D000h | FFFF DFFFh | 00FF D000h | 00FF DFFFh | 4K           |
| EB01   | FFFF E000h | FFFF EFFFh | 00FF E000h | 00FF EFFFh | 4K           |
| EB00   | FFFF F000h | FFFF FFFFh | 00FF F000h | 00FF FFFFh | 4K           |



### 4.9.3 FCU コマンド

本アプリケーションノートで使用している FCU コマンドのフォーマットを表 12 に示します。詳細はユーザーズマニュアル ハードウェア編の ROM (コード格納用フラッシュメモリ) をご参照ください。

なお、FCU コマンドは volatile および evenaccess でコンパイルの最適化を抑制する必要があります。

表 12 FCU コマンドのフォーマット

| コマンド            | バス<br>サイ<br>クル<br>数 | 1 サイクル目  |         | 2 サイクル目  |         | 3 サイクル目  |         | 4~5 サイクル目 |         | 6 サイクル目  |         | 7~130 サイクル目 |         | 131 サイクル目 |         |
|-----------------|---------------------|----------|---------|----------|---------|----------|---------|-----------|---------|----------|---------|-------------|---------|-----------|---------|
|                 |                     | アド<br>レス | デー<br>タ | アド<br>レス | デー<br>タ | アド<br>レス | デー<br>タ | アド<br>レス  | デー<br>タ | アド<br>レス | デー<br>タ | アド<br>レス    | デー<br>タ | アド<br>レス  | デー<br>タ |
| P/E ノーマルモード移行   | 1                   | RA       | FFh     | —        | —       | —        | —       | —         | —       | —        | —       | —           | —       | —         | —       |
| 周辺クロック通知        | 6                   | RA       | E9h     | RA       | 03h     | RA       | 0F0Fh   | RA        | 0F0Fh   | RA       | D0h     | —           | —       | —         | —       |
| プログラム           | 131                 | RA       | E8h     | RA       | 80h     | WA       | WDn     | RA        | WDn     | RA       | WDn     | RA          | WDn     | RA        | D0h     |
| ブロックイレーズ        | 2                   | RA       | 20h     | BA       | D0h     | —        | —       | —         | —       | —        | —       | —           | —       | —         | —       |
| ステータスレジスタ<br>リア | 1                   | RA       | 50h     | —        | —       | —        | —       | —         | —       | —        | —       | —           | —       | —         | —       |

**【記号説明】**

- アドレスの列 RA: ROM 書き込み/消去用のアドレス
- WA: ROM 書き込み先アドレス
- BA: ROM 消去ブロックアドレス
- データの列 WDn: 書き込みデータ n ワード目 (n = 1 ~ 128)

4.9.4 ユーザマットの書き込み/消去手順

本アプリケーションノートにおけるユーザマット書き込み/消去手順を図 11 に示します。

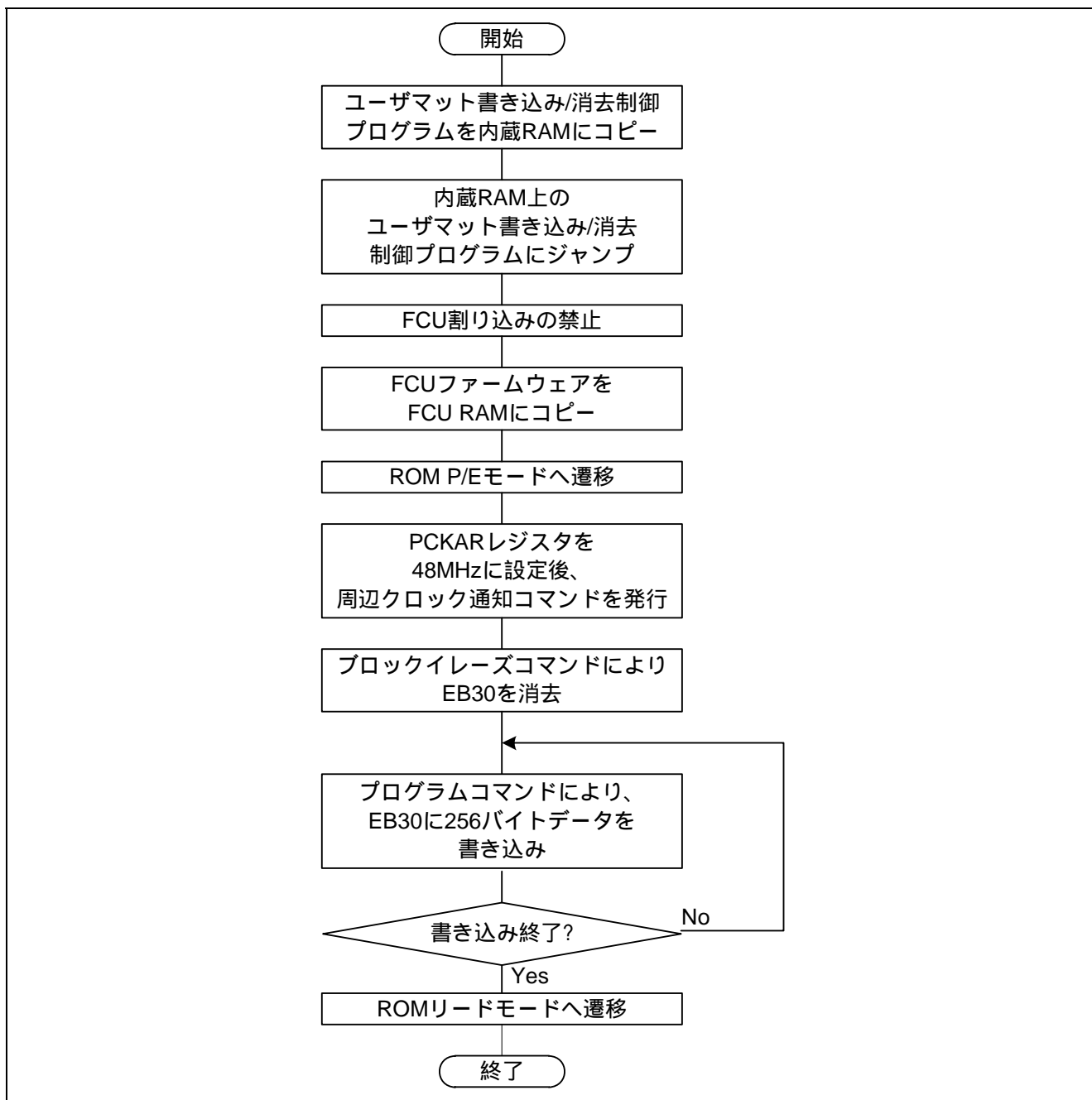


図 11 本アプリケーションノートにおけるユーザマット書き込み/消去手順

#### 4.10 セクション設定

スレーブのセクション設定を表 13 に示します。

表 13 スレーブのセクション設定

| セクション名         | 開始アドレス     | 説明   |                                  |
|----------------|------------|--|----------------------------------|
| B_1            | 0000 1000h | 未初期化データ領域 (ALIGN = 1)                                |                                  |
| B              |            | 未初期化データ領域 (ALIGN = 4)                                |                                  |
| R              |            | [D]セクションを ROM 化支援オプションにより RAM 上にマップした領域              |                                  |
| SU             |            | ユーザスタック領域  |                                  |
| SI             |            | 割り込みスタック領域   |                                  |
| RF_UPDATE_FUNC |            | [PF_UPDATE_FUNC]セクションを ROM 化支援オプションにより RAM 上にマップした領域 |                                  |
| PRResetPRG     |            | FF7F C000h   | プログラム領域 (PowerON_Reset_PC プログラム) |
| C              | FF7F C100h | 定数領域 (ALIGN = 4)                                     |                                  |
| C\$DSEC        |            | 初期化データ領域のセクション初期化用テーブル                               |                                  |
| C\$BSEC        |            | 未初期化データ領域のセクション初期化用テーブル                              |                                  |
| C\$VECT        |            | 可変ベクタ領域  |                                  |
| D              |            | 初期化データ領域 (ALIGN = 4)                                 |                                  |
| P              |            | プログラム領域  |                                  |
| PIntPRG        |            | プログラム領域 (割り込みプログラム)                                  |                                  |
| PF_UPDATE_FUNC |            | プログラム領域 (ユーザマットの書き込み/制御プログラム)                        |                                  |
| FIXEDVECT      |            | FF7F FFFCh   | 固定ベクタ領域 (リセットベクタ)                |

## 5. ソフトウェア説明

### 5.1 ファイル構成

スレーブのファイル構成を表 14 に示します。表 14 で示されたファイル以外は、High-performance Embedded Workshop が自動生成するファイルを使用しています。

表 14 スレーブのファイル構成

| ファイル名                        | 内容   |
|------------------------------|--|
| resetprg.c (* <sup>1</sup> ) | 初期設定処理   |
| vecttbl.c (* <sup>2</sup> )  | 固定ベクタテーブル定義  |
| vect.h (* <sup>3</sup> )     | 割り込み処理関数定義   |
| intprg.c (* <sup>4</sup> )   | 割り込み処理   |
| main.c                       | マスタとのクロック同期式シリアル通信による通信コマンド、消去ブロック番号、書き込みデータサイズ、および書き込みデータの受信制御、ユーザマットのブロック消去および書き込み制御、正常終了時およびエラー発生時の LED の表示制御 |

- 【注】 \*1 High-performance Embedded Workshop が自動生成するファイルですが、本アプリケーションノートでは、PowerON\_Reset\_PC 関数内の HardwareSetup 関数の呼び出しのコメントアウトを解除して、main.c ファイル内の HardwareSetup 関数を PowerON\_Reset\_PC 関数から呼び出すように変更しています。
- \*2 High-performance Embedded Workshop が自動生成するファイルですが、本アプリケーションノートでは、固定ベクタテーブルの特権命令例外、未定義命令例外、浮動小数点例外、ノンマスクブル割り込み、予約領域の Dummy 関数の定義をコメントアウトして、リセットベクタだけの定義に変更しています。
- \*3 High-performance Embedded Workshop が自動生成するファイルですが、本アプリケーションノートでは、Excep\_SuperVisorInst 関数、Dummy 関数、Excep\_UndefinedInst 関数、Excep\_FloatingPoint 関数、NonMaskableInterrupt 関数の定義をコメントアウトしています。
- \*4 High-performance Embedded Workshop が自動生成するファイルですが、本アプリケーションノートでは、Excep\_SuperVisorInst 関数、Dummy 関数、Excep\_UndefinedInst 関数、Excep\_FloatingPoint 関数、NonMaskableInterrupt 関数の記述をコメントアウトしています。

## 5.2 関数構成

スレーブの関数一覧を表 15 に、図 12 にスレーブの関数階層構造を示します。

表 15 スレーブの関数一覧

| 関数名                         | ファイル名      | 概要                     |
|-----------------------------|------------|------------------------|
| PowerON_Reset_PC            | resetprg.c | 初期設定関数                 |
| HardwareSetup               | main.c     | MCU 初期設定関数             |
| main                        | main.c     | メイン関数                  |
| Flash_Update                | main.c     | ユーザマツ書き込み/消去制御関数       |
| fcu_Interrupt_Disable       | main.c     | FCU 割り込み禁止制御関数         |
| fcu_Reset                   | main.c     | FCU 初期化関数              |
| fcu_Transfer_Firmware       | main.c     | FCU ファームウェア転送制御関数      |
| fcu_Transition_RomRead_Mode | main.c     | ROM リードモード遷移制御関数       |
| fcu_Transition_RomPE_Mode   | main.c     | ROM P/E モード遷移制御関数      |
| fcu_Notify_Peripheral_Clock | main.c     | FCU 周辺クロック通知コマンド発行制御関数 |
| fcu_Erase                   | main.c     | ユーザマツ消去制御関数            |
| fcu_Write                   | main.c     | ユーザマツ書き込み制御関数          |
| Indicate_Ending_LED         | main.c     | 正常終了処理関数               |
| Indicate_Error_LED          | main.c     | エラー終了処理関数              |
| SCI_Rcv1byte                | main.c     | 1 バイトデータ受信関数           |
| SCI_Rcvnbyte                | main.c     | n バイトデータ受信関数           |

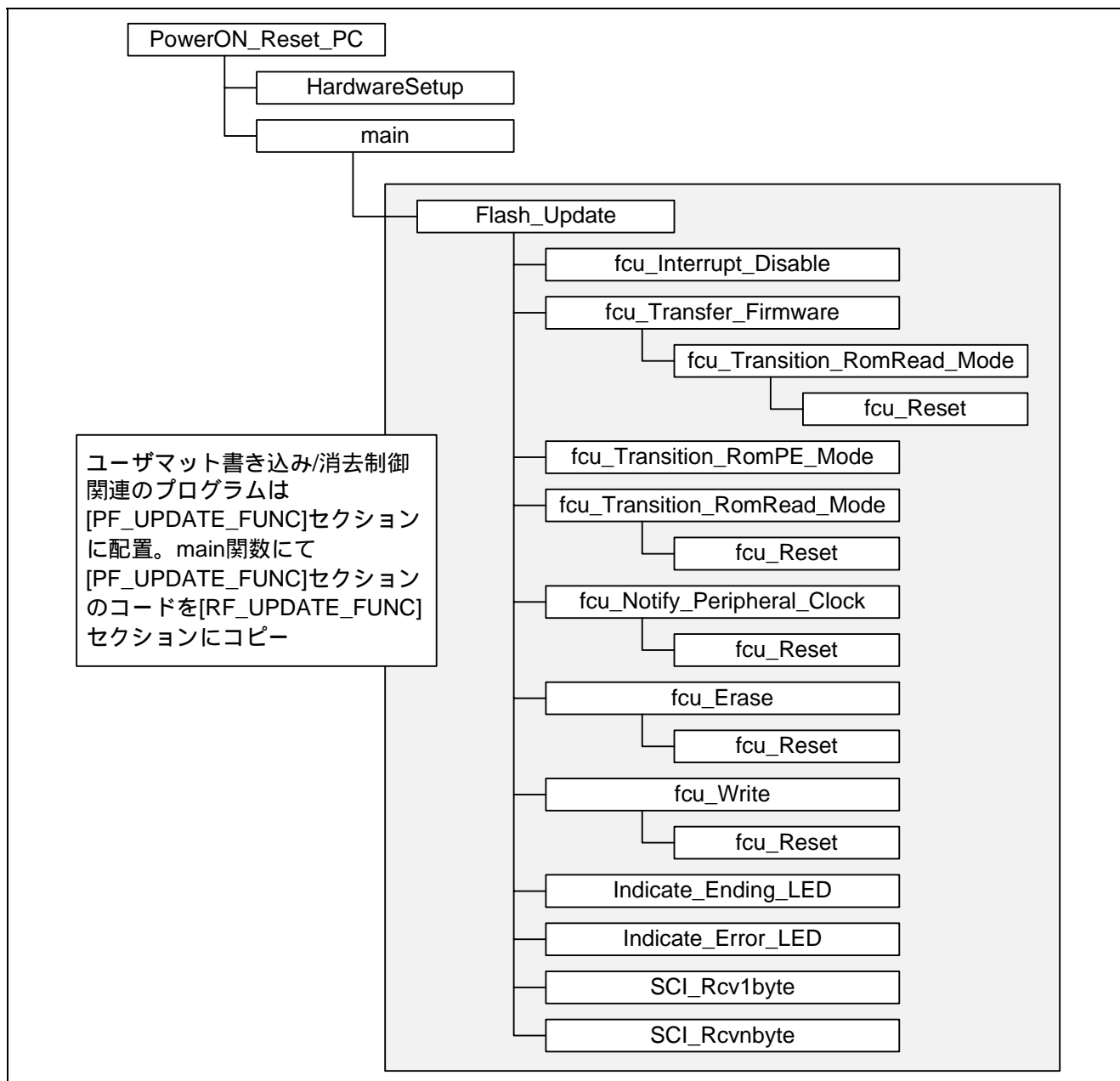


図 12 スレーブの関数階層構造

### 5.3 記号定数説明

スレーブが使用する記号定数を表 16 に示します。

表 16 スレーブの記号定数一覧

| 記号定数名          | 設定値              | 内容                    | 使用関数   |
|----------------|------------------|-----------------------|--|
| FSTART         | 0x10             | 書き込み/消去開始コマンド         | main   |
| ERASE          | 0x11             | 消去開始コマンド              | Flash_Update   |
| WRITE          | 0x12             | 書き込み開始コマンド            | Flash_Update   |
| LED_ON         | 0                | LED 点灯時の設定値           | main<br>Indicate_Ending_LED<br>Indicate_Error_LED                  |
| LED_OFF        | 1                | LED 消灯時の設定値           | HardwareSetup<br>main<br>Indicate_Ending_LED<br>Indicate_Error_LED |
| RSK_LED0       | PORT0.DR.BIT.B2  | 評価ボード搭載 LED0 の点灯/消灯制御 | HardwareSetup<br>main<br>Indicate_Ending_LED<br>Indicate_Error_LED |
| RSK_LED1       | PORT0.DR.BIT.B3  | 評価ボード搭載 LED1 の点灯/消灯制御 | HardwareSetup<br>main<br>Indicate_Ending_LED<br>Indicate_Error_LED |
| RSK_LED2       | PORT0.DR.BIT.B5  | 評価ボード搭載 LED2 の点灯/消灯制御 | HardwareSetup<br>main<br>Indicate_Ending_LED<br>Indicate_Error_LED |
| RSK_LED3       | PORT3.DR.BIT.B4  | 評価ボード搭載 LED3 の点灯/消灯制御 | HardwareSetup<br>main<br>Indicate_Ending_LED<br>Indicate_Error_LED |
| RSK_LED0_DDR   | PORT0.DDR.BIT.B2 | 評価ボード搭載 LED0 の入出力制御   | HardwareSetup  |
| RSK_LED1_DDR   | PORT0.DDR.BIT.B3 | 評価ボード搭載 LED1 の入出力制御   | HardwareSetup  |
| RSK_LED2_DDR   | PORT0.DDR.BIT.B5 | 評価ボード搭載 LED2 の入出力制御   | HardwareSetup  |
| RSK_LED3_DDR   | PORT3.DDR.BIT.B4 | 評価ボード搭載 LED3 の入出力制御   | HardwareSetup  |
| ASSERT         | 0                | Busy ポートアサート時の設定値     | main<br>Flash_Update   |
| NEGATE         | 1                | Busy ポートネゲート時の設定値     | main<br>Flash_Update   |
| SLAVE_BUSY     | PORT0.DR.BIT.B1  | Busy ポートの出力制御         | HardwareSetup<br>main<br>Flash_Update                              |
| SLAVE_BUSY_DDR | PORT0.DDR.BIT.B1 | Busy ポートの入出力制御        | HardwareSetup  |

表 16 スレーブの記号定数一覧 (つづき)

| 記号定数名            | 設定値        | 内容  | 使用関数                                      |
|------------------|------------|---|---|
| PCKA_48MHZ       | 0x0030     | PCKAR レジスタに設定する周辺モジュールクロック (PCLK) の周波数データ                 | fcu_Notify_Peripheral_Clock               |
| WAIT_TE16K       | 7603200    | タイムアウト (tE16K × 1.1) データ<br>tE16K: 16K バイトの消去ブロックに対する消去時間 | fcu_Transition_RomRead_Modefcu_Erase      |
| WAIT_TP256       | 345600     | タイムアウト (tP256 × 1.1) データ<br>tP256: 256 バイトデータの書き込み時間      | fcu_Write                                 |
| WAIT_TRESW2      | 2520       | ウェイト (tRESW2) データ<br>tRESW2: 書き込み/消去中のリセットパルス幅            | fcu_Reset                                 |
| WAIT_TPCKA       | 1636       | タイムアウト (tPCKA) データ  | fcu_Notify_Peripheral_Clock               |
| WAIT_LED         | 2000000    | スレーブのユーザマットの書き込み/消去が正常に終了した際に表示する LED の点灯間隔の時間データ         | Indicate_Ending_LED<br>Indicate_Error_LED |
| FCU_FIRM_TOP     | 0xFEFFE000 | FCU ファームウェア格納領域の先頭アドレス                                    | fcu_Transfer_Firmware                     |
| FCU_RAM_TOP      | 0x007F8000 | FCU RAM の先頭アドレス   | fcu_Transfer_Firmware                     |
| FCU_RAM_SIZE     | 0x2000     | FCU RAM のサイズ  | fcu_Transfer_Firmware                     |
| SIZE_WRITE_BLOCK | 128        | ユーザマットへの書き込みサイズ (ワードサイズ)                                  | Flash_Update<br>fcu_Program_Verify        |
| BUF_SIZE         | 256        | 書き込みデータ格納領域のサイズ   | —   |
| ERROR_NO_01      | 1          | エラー状態を示すデータ   | Flash_Update<br>Indicate_Error_LED        |
| ERROR_NO_02      | 2          |   |   |
| ERROR_NO_03      | 3          |   |   |
| ERROR_NO_04      | 4          |   |   |
| ERROR_NO_05      | 5          |   |   |
| ERROR_NO_06      | 6          |   |   |
| ERROR_NO_07      | 7          |   |   |
| ERROR_NO_08      | 8          |   |   |
| ERROR_NO_09      | 9          |   |   |
| ERROR_NO_10      | 10         |   |   |
| ERROR_NO_11      | 11         |   |   |
| ERROR_NO_12      | 12         |   |   |



表 16 スレーブの記号定数一覧 (つづき)

| 記号定数名      | 設定値  | 内容  | 使用関数         |
|------------|------|---|--------------|
| EB37_INDEX | 0x00 | スレーブの書き込み/消去を行う消去ブロックを指定するために送信する消去ブロック番号 | Flash_Update |
| EB36_INDEX | 0x01 |   |              |
| EB35_INDEX | 0x02 |   |              |
| EB34_INDEX | 0x03 |   |              |
| EB33_INDEX | 0x04 |   |              |
| EB32_INDEX | 0x05 |   |              |
| EB31_INDEX | 0x06 |   |              |
| EB30_INDEX | 0x07 |   |              |
| EB29_INDEX | 0x08 |   |              |
| EB28_INDEX | 0x09 |   |              |
| EB27_INDEX | 0x0A |   |              |
| EB26_INDEX | 0x0B |   |              |
| EB25_INDEX | 0x0C |   |              |
| EB24_INDEX | 0x0D |   |              |
| EB23_INDEX | 0x0E |   |              |
| EB22_INDEX | 0x0F |   |              |
| EB21_INDEX | 0x10 |   |              |
| EB20_INDEX | 0x11 |   |              |
| EB19_INDEX | 0x12 |   |              |
| EB18_INDEX | 0x13 |   |              |
| EB17_INDEX | 0x14 |   |              |
| EB16_INDEX | 0x15 |   |              |
| EB15_INDEX | 0x16 |   |              |
| EB14_INDEX | 0x17 |   |              |
| EB13_INDEX | 0x18 |   |              |
| EB12_INDEX | 0x19 |   |              |
| EB11_INDEX | 0x1A |   |              |
| EB10_INDEX | 0x1B |   |              |
| EB09_INDEX | 0x1C |   |              |
| EB08_INDEX | 0x1D |   |              |
| EB07_INDEX | 0x1E |   |              |
| EB06_INDEX | 0x1F |   |              |
| EB05_INDEX | 0x20 |   |              |
| EB04_INDEX | 0x21 |   |              |
| EB03_INDEX | 0x22 |   |              |
| EB02_INDEX | 0x23 |   |              |
| EB01_INDEX | 0x24 |   |              |
| EB00_INDEX | 0x25 |   |              |

## 5.4 const 変数説明

スレーブが使用する const 変数を表 17 に示します。

表 17 スレーブの const 変数一覧

| 変数名            | 型                            | 内容  |
|----------------|------------------------------|---|
| tbl_eb_adrs[ ] | ST_EB_ADRS (* <sup>1</sup> ) | 消去ブロック (EB00 ~ EB37) の書き込み/消去用開始アドレス/終了アドレス、読み出し用開始アドレス/終了アドレス、消去ブロックサイズの各データ (760 バイト) |

【注】 \*1 ST\_EB\_ADRS 型の詳細は「5.6.2 構造体 ST\_EB\_ADRS」を参照してください。

## 5.5 RAM 変数説明

スレーブが使用する RAM 変数を表 18 に示します。

表 18 スレーブの RAM 変数一覧

| 変数名                     | 型                             | 内容  |
|-------------------------|-------------------------------|---|
| wrdata_buffer[BUF_SIZE] | unsigned char                 | スレーブから受信した 256 バイトの書き込みデータを格納する配列 (256 バイト)         |
| fcu_info                | ST_FCU_INFO (* <sup>1</sup> ) | ユーザマットの書き込み/消去時に使用する FCU 関連のアドレス情報を格納する構造体 (28 バイト) |
| p_write_buffer          | unsigned short *              | ユーザマット書き込み時の書き込みデータ格納領域のアドレス: 4 バイト                 |
| p_command_adrs          | unsigned char *               | FCU コマンド発行先アドレス (書き込み/消去用アドレス): 4 バイト               |
| p_erase_adrs            | unsigned short *              | 消去時の対象消去ブロックの先頭アドレス (書き込み/消去用アドレス): 4 バイト           |
| p_write_adrs_top        | unsigned short *              | 書き込み時の対象消去ブロックの先頭アドレス (書き込み/消去用アドレス): 4 バイト         |
| p_write_adrs_end        | unsigned short *              | 書き込み時の対象消去ブロックの終了アドレス (書き込み/消去用アドレス): 4 バイト         |
| p_write_adrs_now        | unsigned short *              | 書き込み時の書き込み先アドレス (書き込み/消去用アドレス): 4 バイト               |
| eb_block_size           | unsigned long                 | 対象消去ブロックのブロックサイズ: 4 バイト                             |

【注】 \*1 ST\_FCU\_INFO 型の詳細は「5.6.1 構造体 ST\_FCU\_INFO」を参照してください。

## 5.6 構造体説明

### 5.6.1 構造体 ST\_FCU\_INFO

スレーブが使用する構造体 ST\_FCU\_INFO の仕様を表 19 に示します。

表 19 構造体 ST\_FCU\_INFO の仕様

| メンバ名             | 型   | 内容                                  |
|------------------|---|-------------------------------------|
| p_write_buffer   | unsigned short *                            | ユーザマツト書き込み時の書き込みデータ格納領域のアドレス        |
| p_command_adrs   | volatile<br>__evenaccess<br>unsigned char * | FCU コマンド発行先アドレス (書き込み/消去用アドレス)      |
| p_erase_adrs     | unsigned short *                            | 消去時の対象消去ブロックの先頭アドレス(書き込み/消去用アドレス)   |
| p_write_adrs_top | unsigned short *                            | 書き込み時の対象消去ブロックの先頭アドレス(書き込み/消去用アドレス) |
| p_write_adrs_end | unsigned short *                            | 書き込み時の対象消去ブロックの終了アドレス(書き込み/消去用アドレス) |
| p_write_adrs_now | unsigned short *                            | 書き込み時の書き込み先アドレス(書き込み/消去用アドレス)       |
| eb_block_size    | unsigned long                               | 対象消去ブロックのブロックサイズ                    |

### 5.6.2 構造体 ST\_EB\_ADRS

スレーブが使用する構造体 ST\_EB\_ADRS の仕様を表 20 に示します。

表 20 構造体 ST\_EB\_ADRS の仕様

| メンバ名              | 型             | 内容                       |
|-------------------|---------------|--------------------------|
| eb_write_adrs_top | unsigned long | 消去ブロックの開始アドレス (書き込み/消去用) |
| eb_write_adrs_end | unsigned long | 消去ブロックの終了アドレス (書き込み/消去用) |
| eb_read_adrs_top  | unsigned long | 消去ブロックの開始アドレス (読み出し用)    |
| eb_read_adrs_end  | unsigned long | 消去ブロックの終了アドレス (読み出し用)    |
| eb_block_size     | unsigned long | 消去ブロックのサイズ               |

## 5.7 enum 型説明

スレーブが使用する enum 型 FCU\_STATUS の構成を表 21 に示します。関数の戻り値としてステータスを示します。

表 21 enum 型 FCU\_STATUS の仕様

| メンバ名        | 型           | 値 | 内容    |
|-------------|-------------|---|-------|
| FCU_SUCCESS | signed long | 0 | 正常状態  |
| FCU_ERROR   | signed long | 1 | エラー状態 |

## 5.8 使用 I/O レジスタ説明

スレーブのプログラムで使用する I/O レジスタを以下に示します。なお、設定値は本アプリケーションノートで使用している値であり、初期値とは異なります。

### (1) クロック発生回路

- システムクロックコントロールレジスタ (SCKCR) ビット数: 32 ビット アドレス: 0008 0020h

| ビット     | シンボル     | 設定値  | ビット名                     | 機能   | R/W |
|---------|----------|------|--------------------------|--|-----|
| b11-b8  | PCK[3:0] | 0001 | 周辺モジュールクロック (PCLK) 選択ビット | 0001: ×4<br>PCLK = 48MHz<br>(EXTAL クロック = 12.0MHz 時) | R/W |
| b19-b16 | BCK[3:0] | 0010 | 外部バスクロック (BCLK) 選択ビット    | 0010: ×2<br>BCLK = 24MHz<br>(EXTAL クロック = 12.0MHz 時) | R/W |
| b23     | PSTOP1   | 0    | BCLK 出力停止ビット             | 0: BCLK 出力   | R/W |
| b27-b24 | ICK[3:0] | 0000 | システムクロック (ICLK) 選択ビット    | 0000: ×8<br>ICLK = 96MHz<br>(EXTAL クロック = 12.0MHz 時) | R/W |

### (2) I/O ポート

- ポート 0 データレジスタ (P0.DR) ビット数: 8 ビット アドレス: 0008 C020h

| ビット | シンボル | 設定値 | ビット名           | 機能           | R/W |
|-----|------|-----|----------------|--------------|-----|
| b2  | B2   | 0   | P02 出力データ格納ビット | 0: 出力データ = 0 | R/W |
|     |      | 1   |                | 1: 出力データ = 1 |     |
| b3  | B3   | 0   | P03 出力データ格納ビット | 0: 出力データ = 0 | R/W |
|     |      | 1   |                | 1: 出力データ = 1 |     |
| b5  | B5   | 0   | P05 出力データ格納ビット | 0: 出力データ = 0 | R/W |
|     |      | 1   |                | 1: 出力データ = 1 |     |

- ポート 3 データレジスタ (P3.DR) ビット数: 8 ビット アドレス: 0008 C023h

| ビット | シンボル | 設定値 | ビット名           | 機能           | R/W |
|-----|------|-----|----------------|--------------|-----|
| b4  | B4   | 0   | P34 出力データ格納ビット | 0: 出力データ = 0 | R/W |
|     |      | 1   |                | 1: 出力データ = 1 |     |

- ポートファンクションレジスタ F (PFSCI) ビット数: 8 ビット アドレス: 0008 C10Fh

| ビット | シンボル  | 設定値 | ビット名         | 機能  | R/W |
|-----|-------|-----|--------------|---|-----|
| b2  | SCI2S | 0   | SCI2 端子選択ビット | 0: P12 を RxD2-A 端子として設定<br>P11 を SCK2-A 端子として設定<br>P13 を TxD2-A 端子として設定 | R/W |

## RX62N グループ、RX621 グループ ユーザブートモードによる内蔵フラッシュメモリ書き換え (スレーブ)

- ポート0 データディレクションレジスタ (P0.DDR) ビット数: 8 ビット アドレス: 0008 C000h

| ビット | シンボル | 設定値 | ビット名           | 機能       | R/W |
|-----|------|-----|----------------|----------|-----|
| b2  | B2   | 1   | P02 入力/出力指定ビット | 1: 出力ポート | R/W |
| b3  | B3   | 1   | P03 入力/出力指定ビット | 1: 出力ポート | R/W |
| b5  | B5   | 1   | P05 入力/出力指定ビット | 1: 出力ポート | R/W |

- ポート3 データディレクションレジスタ (P3.DDR) ビット数: 8 ビット アドレス: 0008 C003h

| ビット | シンボル | 設定値 | ビット名           | 機能       | R/W |
|-----|------|-----|----------------|----------|-----|
| b4  | B4   | 1   | P34 入力/出力指定ビット | 1: 出力ポート | R/W |

- ポート1 入力バッファコントロールレジスタ (P1.ICR) ビット数: 8 ビット アドレス: 0008 C061h

| ビット | シンボル | 設定値 | ビット名            | 機能                | R/W |
|-----|------|-----|-----------------|-------------------|-----|
| b1  | B1   | 1   | P11 入力バッファ制御ビット | 1: P11 の入力バッファは有効 | R/W |
| b2  | B2   | 1   | P12 入力バッファ制御ビット | 1: P12 の入力バッファは有効 | R/W |

### (3) 低消費電力低減機能

- モジュールストップコントロールレジスタ B (MSTPCRB) ビット数: 32 ビット アドレス: 0008 0014h

| ビット | シンボル    | 設定値 | ビット名   | 機能                          | R/W |
|-----|---------|-----|--|-----------------------------|-----|
| b29 | MSTPB29 | 0   | シリアルコミュニケーション<br>インタフェース2 モジュール<br>ストップ設定ビット | 0: SCI2 のモジュールストップ状<br>態の解除 | R/W |

(4) シリアルコミュニケーションインタフェース 2 (SCI2)

- SCI2 シリアルコントロールレジスタ (SCI2.SCR) ビット数: 8 ビット アドレス: 0008 8252h  
(シリアルコミュニケーションインタフェースモードのとき (SCI2.SCMR.SMIF ビット = 0))

| ビット   | シンボル     | 設定値 | ビット名          | 機能  | R/W                      |
|-------|----------|-----|---------------|---|--------------------------|
| b1-b0 | CKE[1:0] | 10  | クロック許可ビット     | (クロック同期式の場合)<br>10:外部クロック<br>SCK2 端子はクロック入力端子 | R/W<br>(* <sup>1</sup> ) |
| b2    | TEIE     | 0   | 送信完了割り込み許可ビット | 0: TEI2 割り込み要求を禁止                             | R/W                      |
| b4    | RE       | 0   | 受信許可ビット       | 0: シリアル受信動作を禁止                                | R/W<br>(* <sup>2</sup> ) |
|       |          | 1   |               | 1: シリアル受信動作を許可                                |                          |
| b5    | TE       | 0   | 送信許可ビット       | 0: シリアル送信動作を禁止                                | R/W<br>(* <sup>2</sup> ) |
| b6    | RIE      | 0   | 受信割り込み許可ビット   | 0: RXI2、ERI2 割り込み要求を禁止                        | R/W                      |
|       |          | 1   |               | 1: RXI2、ERI2 割り込み要求を許可                        |                          |
| b7    | TIE      | 0   | 送信割り込み許可ビット   | 0: TXI2 割り込み要求を禁止                             | R/W                      |

【注】 \*1 TE ビット = 0、RE ビット = 0 の場合のみ書き込み可能です。

\*2 TE ビット = 0、RE ビット = 0 の場合のみ"1"を書き込み可能です。一旦、TE、RE ビットのいずれかを"1"に設定した後は、TE ビット = 0、RE ビット = 0 の書き込みのみ可能になります。

- SCI2 シリアルモードレジスタ (SCI2.SMR) ビット数: 8 ビット アドレス: 0008 8250h  
(シリアルコミュニケーションインタフェースモードのとき (SCI2.SCMR.SMIF ビット = 0))

| ビット   | シンボル     | 設定値 | ビット名                | 機能                                      | R/W                      |
|-------|----------|-----|---------------------|---|--------------------------|
| b1-b0 | CKS[1:0] | 00  | クロック選択ビット           | 00: PCLK クロック (n = 0) (* <sup>1</sup> ) | R/W<br>(* <sup>2</sup> ) |
| b7    | CM       | 1   | コミュニケーションモード<br>ビット | 1: クロック同期式モードで動作                        | R/W<br>(* <sup>2</sup> ) |

【注】 \*1 n の設定値については 7. 参考ドキュメント「ユーザーズマニュアル」を参照してください。

\*2 SCI2.SCR.TE ビット = 0、SCI2.SCR.RE ビット = 0 (シリアル送信動作を禁止、かつシリアル受信動作を禁止) の場合のみ書き込み可能です。

## RX62N グループ、RX621 グループ ユーザブートモードによる内蔵フラッシュメモリ書き換え (スレーブ)

- SCI2 スマートカードモードレジスタ (SCI2.SCMR) ビット数: 8 ビット アドレス: 0008 8256h

| ビット | シンボル | 設定値 | ビット名                   | 機能                         | R/W                   |
|-----|------|-----|------------------------|----------------------------|-----------------------|
| b0  | SMIF | 0   | スマートカードインタフェースモード選択ビット | 0: シリアルコミュニケーションインタフェースモード | R/W (* <sup>1</sup> ) |
| b3  | SDIR | 0   | ビットオーダー選択ビット           | 0: LSB ファーストで送受信           | R/W (* <sup>1</sup> ) |

【注】 \*1 SCI2.SCR.TE ビット = 0、SCI2.SCR.RE ビット = 0 (シリアル送信動作を禁止、かつシリアル受信動作を禁止) の場合のみ書き込み可能です。

- SCI2 シリアルステータスレジスタ (SCI2.SSR) ビット数: 8 ビット アドレス: 0008 8254h  
(シリアルコミュニケーションインタフェースモードのとき (SCI2.SCMR.SMIF ビット = 0))

| ビット | シンボル | 設定値                 | ビット名        | 機能                                   | R/W                   |
|-----|------|---------------------|-------------|--------------------------------------|-----------------------|
| b5  | ORER | — (* <sup>1</sup> ) | オーバランエラーフラグ | 0: オーバランエラーの発生なし<br>1: オーバランエラーの発生あり | R/W (* <sup>2</sup> ) |

【注】 \*1 本アプリケーションノートでは、ORER ビットは読み出しだけを行います。フラグをクリアするための"0"書き込みは行いません。

\*2 フラグをクリアするための"0"書き込みのみ可能です。

- SCI2 レシーブデータレジスタ (SCI2.RDR) ビット数: 8 ビット アドレス: 0008 8255h

| ビット   | シンボル | 設定値 | ビット名 | 機能       | R/W |
|-------|------|-----|------|----------|-----|
| b7-b0 | —    | —   | —    | 受信データを格納 | R   |

## RX62N グループ、RX621 グループ ユーザブートモードによる内蔵フラッシュメモリ書き換え (スレーブ)

### (5) 割り込みコントローラ (ICUa)

- 割り込み要因プライオリティレジスタ 82 (IPR82) ビット数: 8 ビット アドレス: 0008 7382h

| ビット   | シンボル     | 設定値  | ビット名                | 機能                   | R/W |
|-------|----------|------|---------------------|----------------------|-----|
| b3-b0 | IPR[3:0] | 0000 | SCI2 割り込み優先レベル設定ビット | 0000: レベル 0 (割り込み禁止) | R/W |

- 割り込み要求許可レジスタ 1B (IER1B) ビット数: 8 ビット アドレス: 0008 721Bh

| ビット | シンボル | 設定値 | ビット名               | 機能               | R/W |
|-----|------|-----|--------------------|------------------|-----|
| b7  | IEN7 | 0   | RX12 割り込み要求許可ビット 7 | 0: RX12 割り込み要求禁止 | R/W |

- 割り込み要求レジスタ 223 (IR223) ビット数: 8 ビット アドレス: 0008 70DFh

| ビット | シンボル | 設定値 | ビット名              | 機能                                   | R/W         |
|-----|------|-----|-------------------|--------------------------------------|-------------|
| b0  | IR   | 0   | RX12 割り込みステータスフラグ | 0: RX12 割り込み要求なし<br>1: RX12 割り込み要求あり | R/W<br>(*1) |

【注】 \*1 フラグをクリアするための"0"書き込みのみ可能。"1"書き込みは禁止です。

- 割り込み要因プライオリティレジスタ 01 (IPR01) ビット数: 8 ビット アドレス: 0008 7301h

| ビット   | シンボル     | 設定値  | ビット名                  | 機能                   | R/W |
|-------|----------|------|-----------------------|----------------------|-----|
| b3-b0 | IPR[3:0] | 0000 | FIFERR 割り込み優先レベル設定ビット | 0000: レベル 0 (割り込み禁止) | R/W |

- 割り込み要因プライオリティレジスタ 02 (IPR02) ビット数: 8 ビット アドレス: 0008 7302h

| ビット   | シンボル     | 設定値  | ビット名                 | 機能                   | R/W |
|-------|----------|------|----------------------|----------------------|-----|
| b3-b0 | IPR[3:0] | 0000 | FRDYI 割り込み優先レベル設定ビット | 0000: レベル 0 (割り込み禁止) | R/W |

- 割り込み要求許可レジスタ 02 (IER02) ビット数: 8 ビット アドレス: 0008 7202h

| ビット | シンボル | 設定値 | ビット名                 | 機能                 | R/W |
|-----|------|-----|----------------------|--------------------|-----|
| b5  | IEN5 | 0   | FIFERR 割り込み要求許可ビット 5 | 0: FIFERR 割り込み要求禁止 | R/W |
| b7  | IEN7 | 0   | FRDYI 割り込み要求許可ビット 7  | 0: FRDYI 割り込み要求禁止  | R/W |



(6) ROM (コード格納用フラッシュメモリ)

- フラッシュアクセスステータスレジスタ (FASTAT) ビット数: 8 ビット アドレス: 007F C410h

| ビット | シンボル   | 設定値 | ビット名                      | 機能   | R/W      |
|-----|--------|-----|---------------------------|--|----------|
| b0  | DFLWPE | 0   | データフラッシュ書き込み/消去プロテクト違反ビット | 0: DFLWE レジスタの設定に違反したデータフラッシュ書き込み/消去系コマンドの発行なし<br>1: DFLWE レジスタの設定に違反したデータフラッシュ書き込み/消去系コマンドの発行あり | R/W (*1) |
| b1  | DFLRPE | 0   | データフラッシュリードプロテクト違反ビット     | 0: DFLRE レジスタの設定に違反したデータフラッシュ読み出しなし<br>1: DFLRE レジスタの設定に違反したデータフラッシュ読み出しあり                       | R/W (*1) |
| b3  | DFLAE  | 0   | データフラッシュアクセス違反ビット         | 0: データフラッシュアクセス違反なし<br>1: データフラッシュアクセス違反あり   | R/W (*1) |
| b4  | CMDLK  | 1   | FCU コマンドロックビット            | 0: FCU はコマンドロック状態ではない<br>1: FCU はコマンドロック状態   | R        |
| b7  | ROMAE  | 0   | ROM アクセス違反ビット             | 0: ROM アクセスエラーなし<br>1: ROM アクセスエラーあり   | R/W (*1) |

【注】 \*1 フラグをクリアするために、"1"を読み出した後に"0"を書き込むことのみ可能です。

- フラッシュアクセスエラー割り込み許可レジスタ (FAEINT) ビット数: 8 ビット アドレス: 007F C411h

| ビット | シンボル     | 設定値 | ビット名                            | 機能   | R/W |
|-----|----------|-----|---------------------------------|--|-----|
| b0  | DFLWPEIE | 0   | データフラッシュ書き込み/消去プロテクト違反割り込み許可ビット | 0: FASTAT.DFLWPE ビット = 1 で FIFERR 割り込み要求を発生しない | R/W |
| b1  | DFLRPEIE | 0   | データフラッシュリードプロテクト違反割り込み許可ビット     | 0: FASTAT.DFLRPE ビット=1 で FIFERR 割り込み要求を発生しない   | R/W |
| b3  | DFLAEIE  | 0   | データフラッシュアクセス違反割り込み許可ビット         | 0: FASTAT.DFLAE ビット=1 で FIFERR 割り込み要求を発生しない    | R/W |
| b4  | CMDLKIE  | 0   | FCU コマンドロック割り込み許可ビット            | 0: FASTAT.CMDLK ビット=1 で FIFERR 割り込み要求を発生しない    | R/W |
| b7  | ROMAEIE  | 0   | ROM アクセス違反割り込み許可ビット             | 0: FASTAT.ROMAE ビット=1 で FIFERR 割り込み要求を発生しない    | R/W |

RX62N グループ、RX621 グループ ユーザブートモードによる内蔵フラッシュメモリ書き換え (スレーブ)

- FCU RAM イネーブルレジスタ (FCURAME) ビット数: 16 ビット アドレス: 007F C454h

| ビット    | シンボル     | 設定値      | ビット名          | 機能  | R/W      |
|--------|----------|----------|---------------|---|----------|
| b0     | FCRME    | 1        | FCU RAM 許可ビット | 0: FCU RAM へのアクセス禁止<br>1: FCU RAM へのアクセス許可  | R/W      |
| b15-b8 | KEY[7:0] | 11000100 | キーコード         | FCRME ビットの書き換えの可否を制御<br>C4h: ワードアクセスで KEY[7: 0] ビットが C4h の場合のみ、FCRME ビットへの書き込みが有効 | R/W (*1) |

【注】 \*1 書き込みデータは保持されません。

- フラッシュステータスレジスタ 0 (FSTATR0) ビット数: 8 ビット アドレス: 007F FFB0h

| ビット | シンボル    | 設定値 | ビット名            | 機能   | R/W |
|-----|---------|-----|-----------------|--|-----|
| b4  | PRGERR  | —   | 書き込みエラービット      | 0: 書き込み処理は正常終了<br>1: 書き込み処理中にエラー発生   | R   |
| b5  | ERSERR  | —   | 消去エラービット        | 0: 消去処理は正常終了<br>1: 消去処理中にエラー発生   | R   |
| b6  | ILGLERR | —   | イリーガルコマンドエラービット | 0: FCU は不正なコマンドや、ROM/データフラッシュアクセスを検出していない<br>1: FCU は不正なコマンドや、ROM/データフラッシュアクセスを検出        | R   |
| b7  | FRDY    | —   | フラッシュレディビット     | 0: 書き込み/消去処理中、書き込み/消去の中断処理中、ロックビットリード 2 コマンド処理中、データフラッシュのブランクチェック処理中<br>1: 上記の処理を実行していない | R   |

- フラッシュステータスレジスタ 1 (FSTATR1) ビット数: 8 ビット アドレス: 007F FFB1h

| ビット | シンボル   | 設定値 | ビット名       | 機能                                    | R/W |
|-----|--------|-----|------------|---------------------------------------|-----|
| b7  | FCUERR | —   | FCU エラービット | 0: FCU の処理でエラー未発生<br>1: FCU の処理でエラー発生 | R   |

RX62N グループ、RX621 グループ ユーザブートモードによる内蔵フラッシュメモリ書き換え (スレーブ)

• フラッシュプロテクトレジスタ (FPROTR) ビット数: 16 ビット アドレス: 007F FFB4h

| ビット    | シンボル       | 設定値      | ビット名                    | 機能   | R/W      |
|--------|------------|----------|-------------------------|--|----------|
| b0     | FPROTCN    | 1        | ロックビットプロテクト<br>キャンセルビット | 1: ロックビットによるプロテクト無効  | R/W      |
| b15-b8 | FPKEY[7:0] | 01010101 | キーコード                   | FPROTCN ビットの書き換えの可否を制御<br>55h: FENTYRY レジスタの値が 0000h 以外の状態で、ワードアクセスで FPKEY[7: 0] ビットが 55h の場合のみ、FPROTCN ビットへの書き込みが有効 | R/W (*1) |

【注】 \*1 書き込みデータは保持されません

• フラッシュリセットレジスタ (FRESETR) ビット数: 16 ビット アドレス: 007F FFB6h

| ビット    | シンボル       | 設定値      | ビット名         | 機能  | R/W      |
|--------|------------|----------|--------------|---|----------|
| b0     | FRESET     | 0        | フラッシュリセットビット | 0: FCU はリセットされない  | R/W      |
|        |            | 1        |              | 1: FCU はリセットされる   |          |
| b15-b8 | FRKEY[7:0] | 11001100 | キーコード        | FRESET ビットの書き換えの可否を制御<br>CCh: ワードアクセスで FRKEY[7: 0] ビットが CCh の場合のみ、FRESET ビットへの書き込みが有効 | R/W (*1) |

【注】 \*1 書き込みデータは保持されません。

RX62N グループ、RX621 グループ ユーザブートモードによる内蔵フラッシュメモリ書き換え (スレーブ)

- フラッシュ P/E モードエントリレジスタ (FENTRYR) ビット数: 16 ビット アドレス: 007F FFB2h

| ビット    | シンボル       | 設定値      | ビット名                        | 機能   | R/W         |
|--------|------------|----------|-----------------------------|--|-------------|
| b0     | FENTRY0    | 0        | ROM P/E モードエントリ<br>ビット 0    | 0: ROM 512K/384K/256K バイ<br>トは、ROM リードモード<br>1: ROM 512K/384K/256K バイ<br>トは、ROM P/E モード                                  | R/W         |
|        |            | 1        |                             |  |             |
| b7     | FENTRYD    | 0        | データフラッシュ P/E モー<br>ドエントリビット | 0: データフラッシュはリード<br>モード   | R/W         |
| b15-b8 | FEKEY[7:0] | 10101010 | キーコード                       | FENTRYD、FENTRY0 ビットの<br>書き換えの可否を制御<br>AAh: ワードアクセスで<br>FEKEY[7: 0]ビットが AAh<br>の場合のみ、FENTRY0、<br>FENTRYD ビットへの書き<br>込みが有効 | R/W<br>(*1) |

【注】 \*1 書き込みデータは保持されません。

- 周辺クロック通知レジスタ (PCKAR) ビット数: 16 ビット アドレス: 007F FFE8h

| ビット   | シンボル      | 設定値      | ビット名        | 機能                      | R/W |
|-------|-----------|----------|-------------|-------------------------|-----|
| b7-b0 | PCKA[7:0] | 00110000 | 周辺クロック通知ビット | 0x30: PCLK の周波数 = 48MHz | R/W |

- フラッシュライトイレースプロテクトレジスタ (FWEPROR) ビット数: 8 ビット アドレス: 0008 C289h

| ビット   | シンボル      | 設定値 | ビット名                | 機能                              | R/W |
|-------|-----------|-----|---------------------|---------------------------------|-----|
| b1-b0 | FLWE[1:0] | 01  | フラッシュ書き込み/消去<br>ビット | 01: 書き込み/消去可能<br>10: 書き込み/消去不可能 | R/W |
|       |           | 10  |                     |                                 |     |

## 5.9 関数仕様

スレープの各関数の仕様を以下に示します。

### (1) PowerON\_Reset\_PC 関数

#### (a) 機能説明

PowerON\_Reset\_PC関数は、スタックポインタの初期化 (PowerON\_Reset\_PC関数に対して#pragma entryを宣言することによりコンパイラが自動的にISP/USPの初期化コードを関数先頭に生成)、INTBの設定 (set\_intb関数: 組み込み関数)、FPSWの初期化 (set\_fpsw関数: 組み込み関数)、RAM領域セクションの初期化 (\_INITISCT関数: 標準ライブラリ関数)、HardwareSetup関数の呼び出し、PSWの初期化 (set\_psw関数: 組み込み関数)、プロセッサモードをユーザモードに設定します。その後、main関数を呼び出します。

#### (b) 引数

なし

#### (c) 戻り値

なし

#### (d) フローチャート

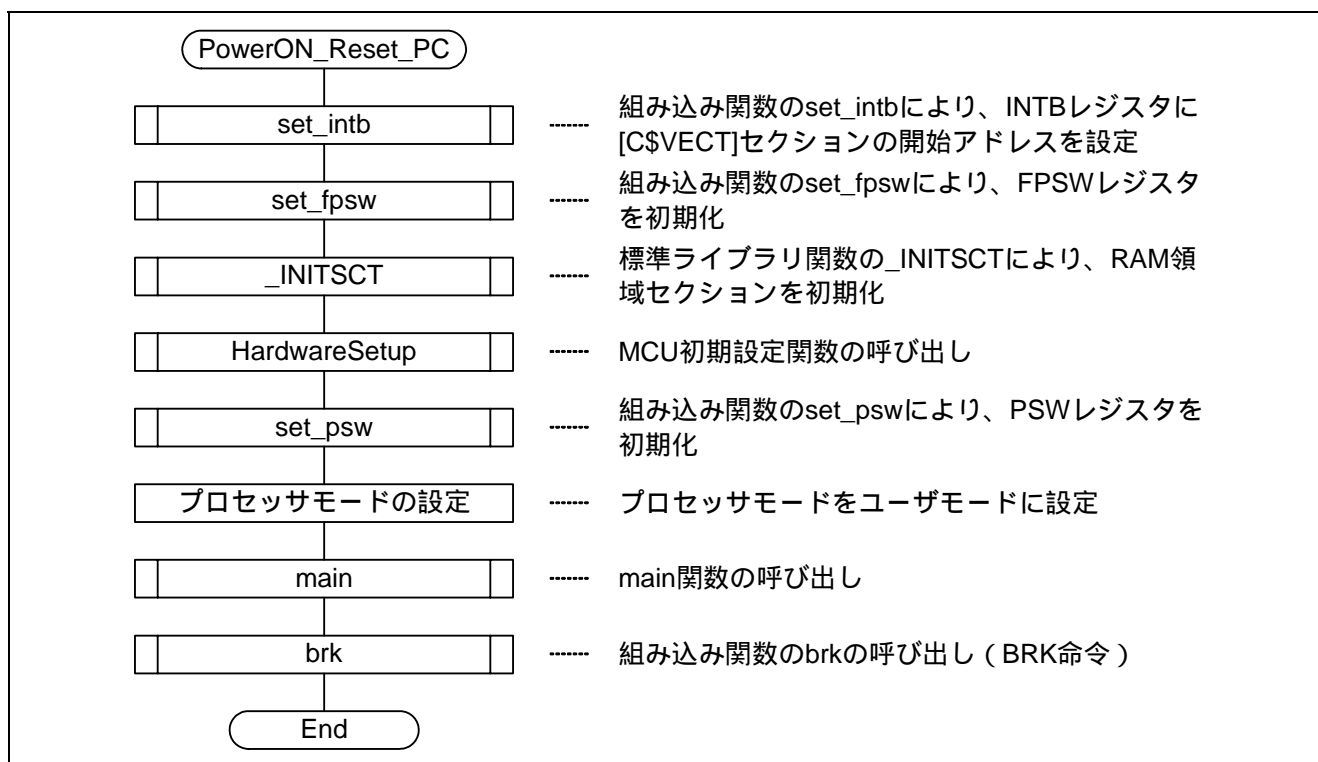


図 13 フローチャート (PowerON\_Reset\_PC) (スレープ)

(2) HardwareSetup 関数

(a) 機能説明

HardwareSetup 関数はMCUの初期設定を行います。クロック (システムクロック (ICLK)、周辺モジュールクロック (PCLK)、および外部バスクロック (BCLK)) の初期設定、LED0~LED3 を接続している I/O ポート (P02、P03、P05、および P34) の初期出力設定、Busy ポート (P01) の初期出力設定、および SCI2 の初期設定を行います。

(b) 引数

なし

(c) 戻り値

なし

(d) フローチャート

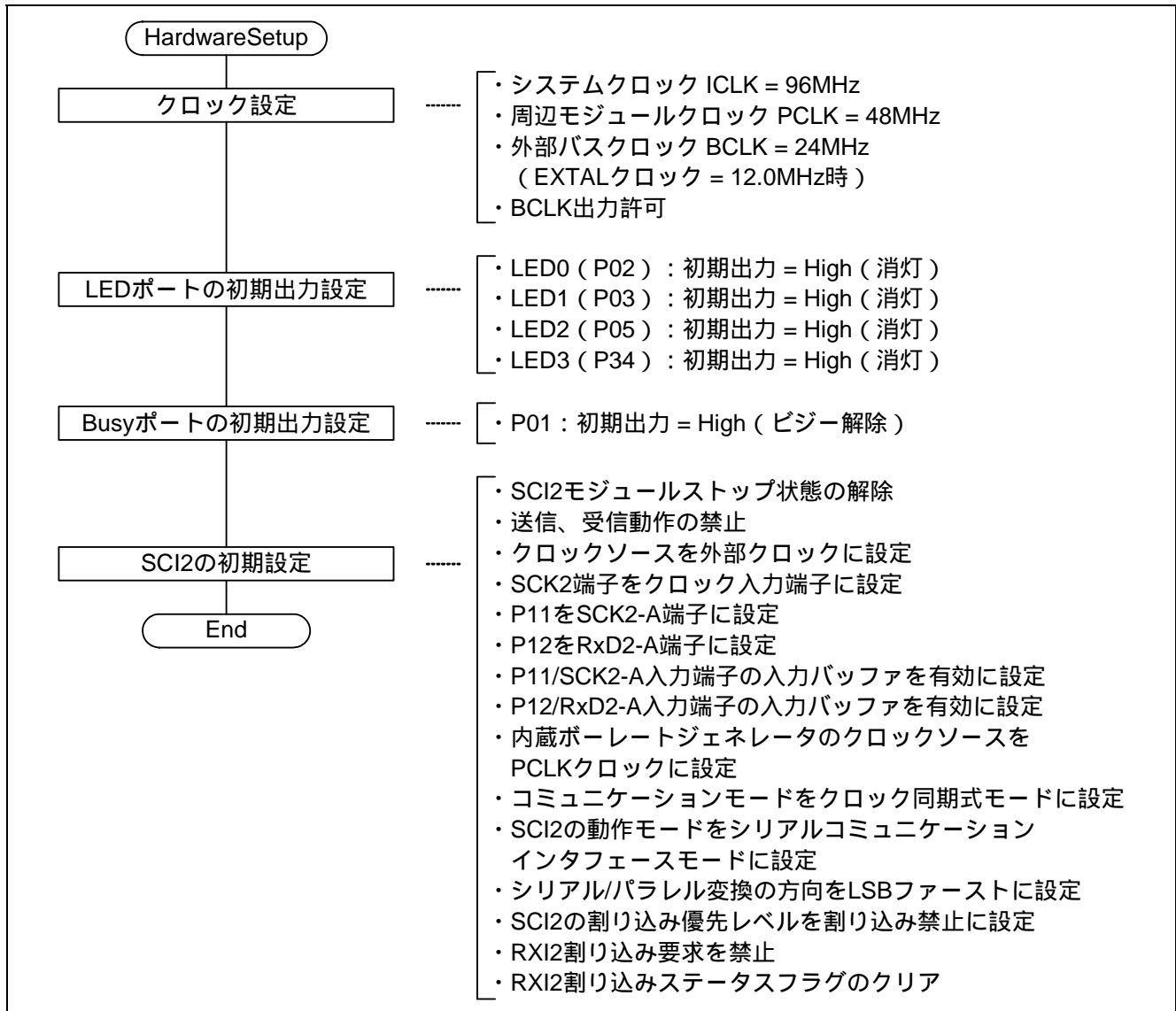


図 14 フローチャート (HardwareSetup) (スレーブ)

(3) main 関数

(a) 機能説明

main 関数は、マスタからの 1 バイトデータの受信制御、ユーザマット書き込み/消去制御プログラムをユーザマット (PF\_UPDATE\_FUNC セクション) から内蔵 RAM (RF\_UPDATE\_FUNC セクション) にコピー、内蔵 RAM 上のユーザマット書き込み/制御プログラム (Flash\_Update 関数) の呼び出しを行います。

(b) 引数

なし

(c) 戻り値

なし

(d) フローチャート

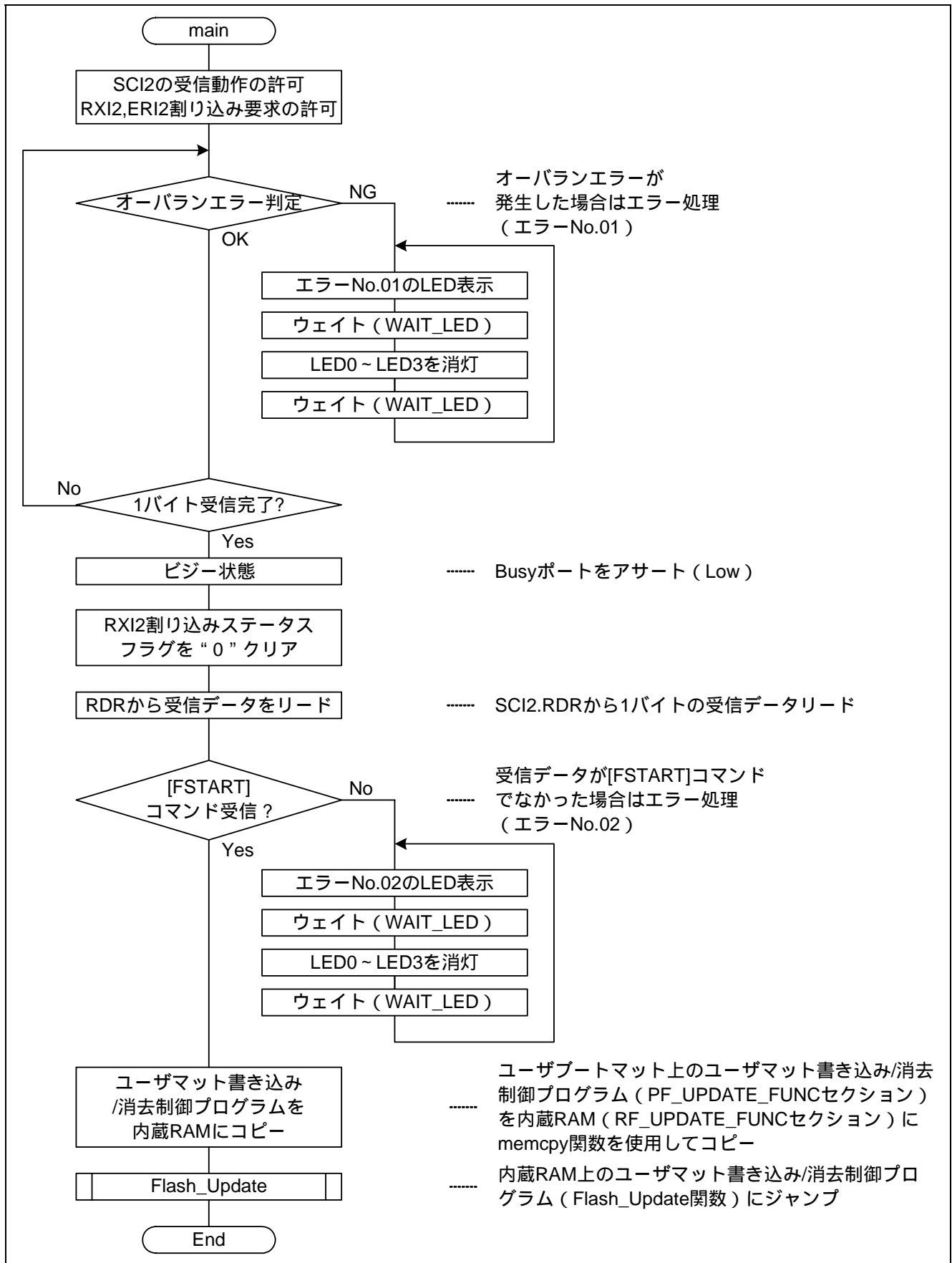


図 15 フローチャート (main) (スレーブ)



(4) Flash\_Update 関数

(a) 機能説明

Flash\_Update 関数は、マスタとのクロック同期式シリアル通信による通信コマンドの受信制御、消去ブロック番号の受信制御、書き込みデータサイズの受信制御、書き込みデータの受信制御、シリアル通信の Busy ポート制御、ユーザマットの書き込み/消去制御、ユーザマットの書き込み/消去の正常終了時に Indicate\_Ending\_LED 関数の呼び出し、エラー終了時に Indicate\_Error\_LED 関数の呼び出しを行います。

(b) 引数

なし

(c) 戻り値

なし

(d) フローチャート

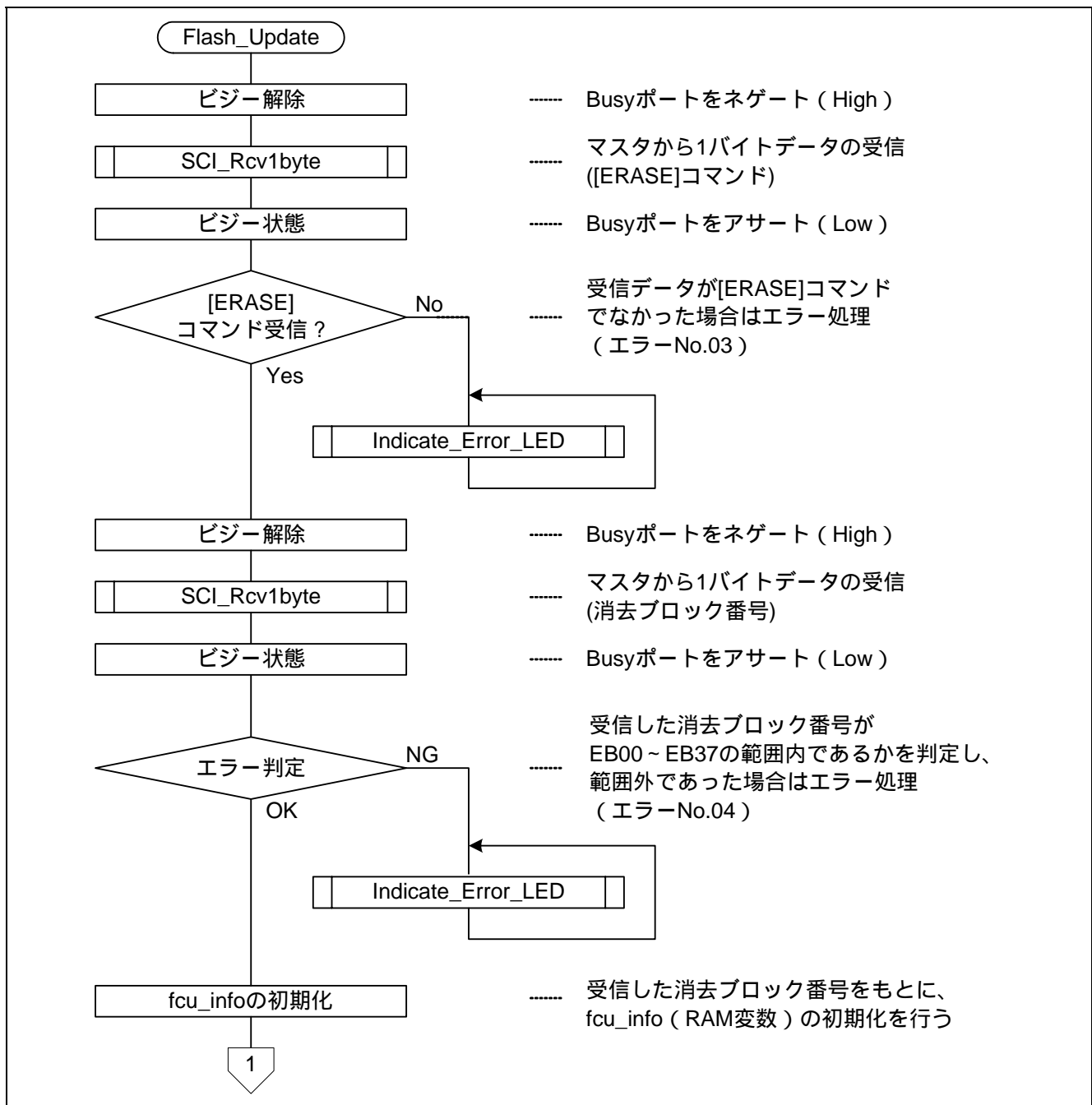


図 16 フローチャート (Flash\_Update) (1) (スレーブ)

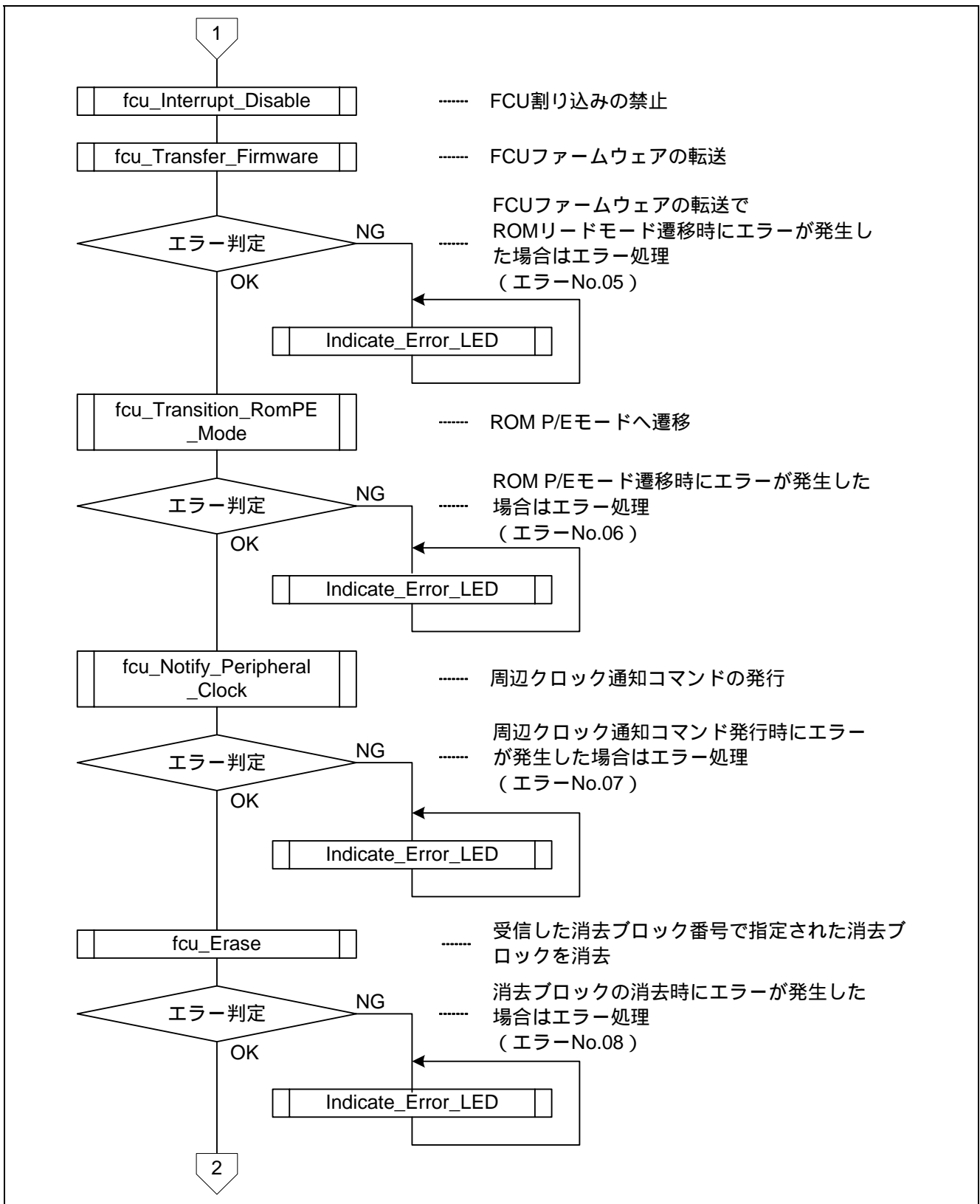


図 17 フローチャート (Flash\_Update) (2) (スレーブ)

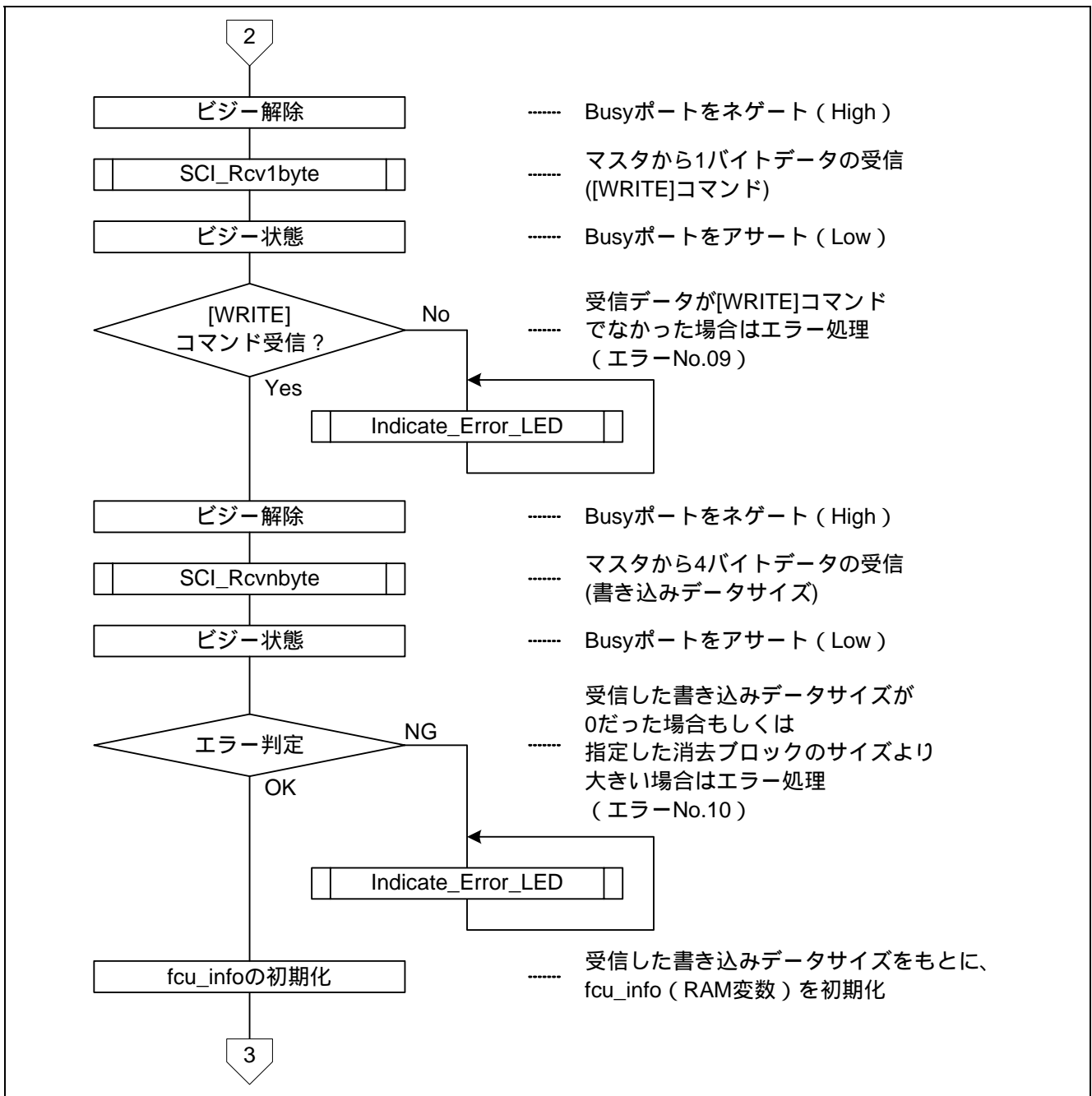


図 18 フローチャート (Flash\_Update) (3) (スレーブ)

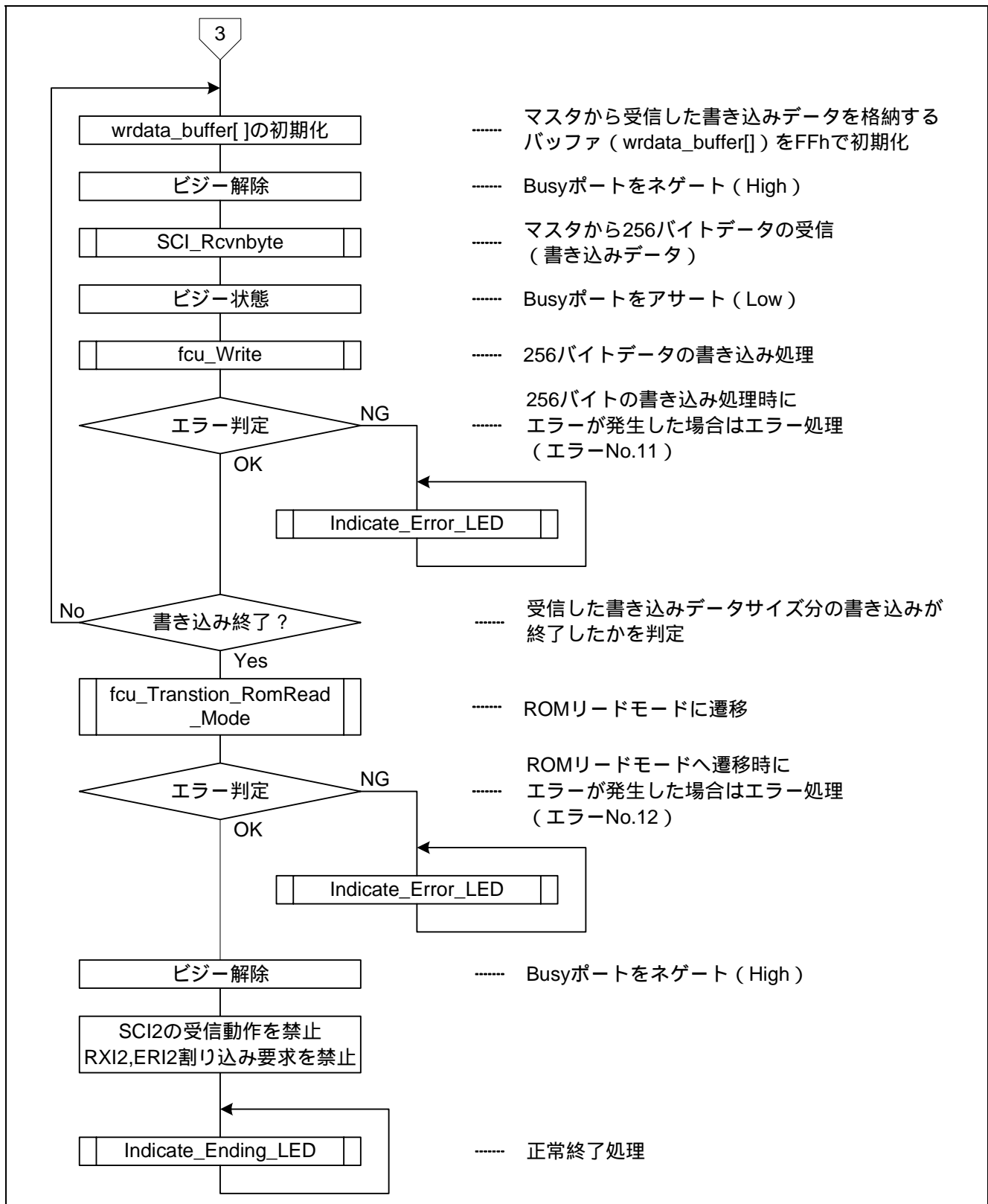


図 19 フローチャート (Flash\_Update) (4) (スレーブ)

(5) fcu\_Interrupt\_Disable 関数

(a) 機能説明

fcu\_Interrupt\_Disable 関数は、ユーザマットの書き込み/消去処理の前に FCU の各種割り込み (FRDYI 割り込み、データフラッシュ書き込み/消去プロテクト違反割り込み、データフラッシュリードプロテクト違反割り込み、データフラッシュアクセス違反割り込み、FCU コマンドロック割り込み、ROM アクセス違反割り込み、および FIFERR 割り込み) の禁止を行います。

(b) 引数

なし

(c) 戻り値

なし

(d) フローチャート

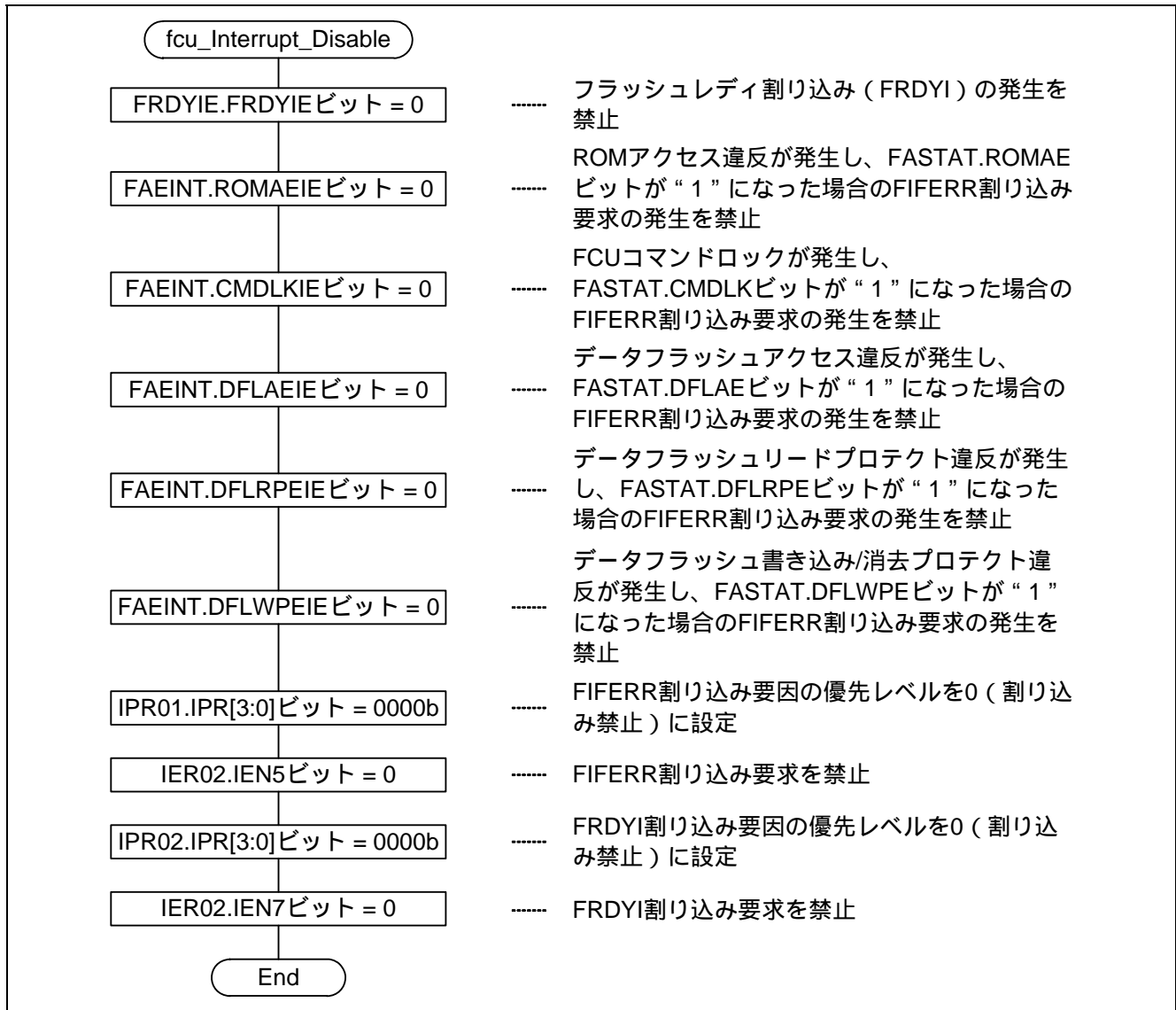


図 20 フローチャート (fcu\_Interrupt\_Disable) (スレーブ)

(6) fcu\_Reset 関数

(a) 機能説明

fcu\_Reset 関数は、FRESETR.FRESETR ビットによる FCU の初期化を行います。

(b) 引数

なし

(c) 戻り値

なし

(d) フローチャート

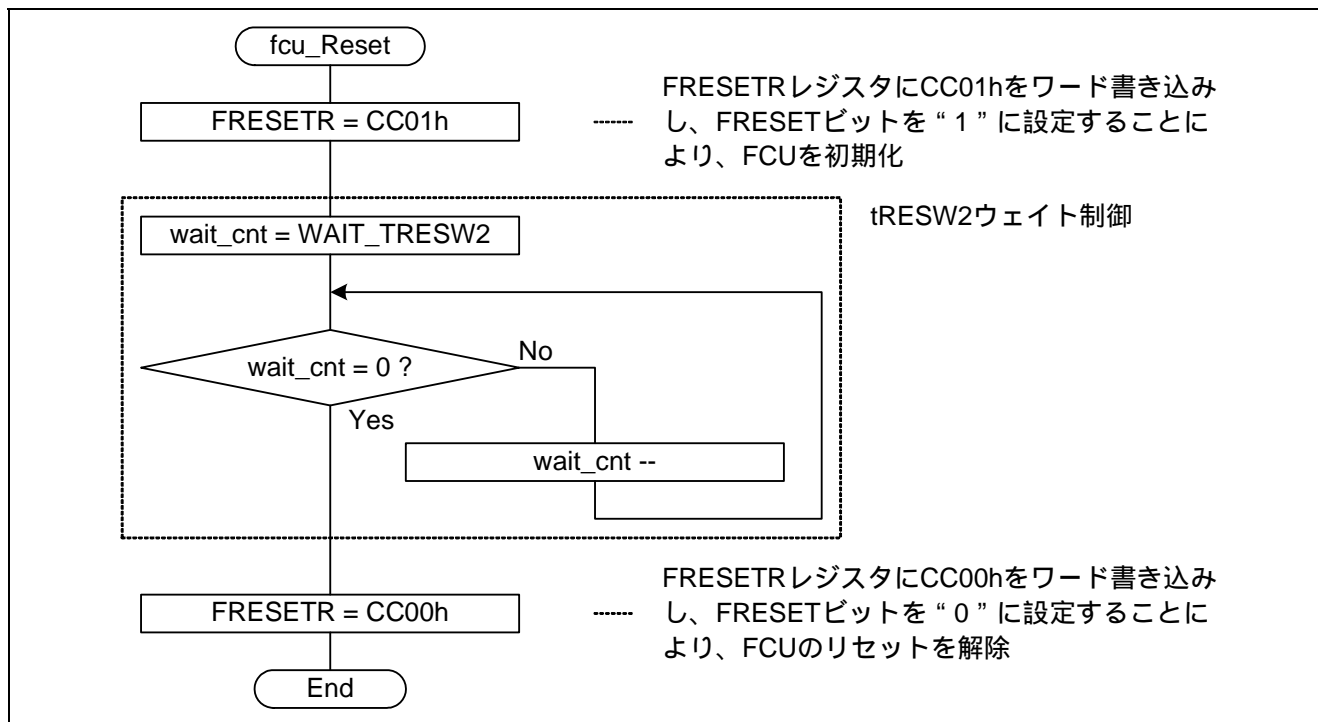


図 21 フローチャート (fcu\_Reset) (スレープ)

(7) fcu\_Transfer\_Firmware 関数

(a) 機能説明

fcu\_Transfer\_Firmware 関数は、FCU ファームウェア格納領域 (FEFF E000h ~ FEFF FFFFh) に格納された FCU ファームウェアを FCU RAM 領域 (007F 8000h ~ 007F 9FFFh) にコピーします。

(b) 引数

表 22 に本関数で使用する引数を示します。

表 22 fcu\_Transfer\_Firmware 関数の引数一覧

| 引数     | 型                                  | 説明  |
|--------|------------------------------------|---|
| 第 1 引数 | ST_FCU_INFO *<br>(* <sup>1</sup> ) | ユーザマットの書き込み/消去時に使用する FCU 関連のアドレス情報を格納している構造体のアドレス |

【注】 \*1 ST\_FCU\_INFO 型の詳細は「5.6.1 構造体 ST\_FCU\_INFO」を参照してください。

(c) 戻り値

表 23 に本関数で使用する戻り値を示します。

表 23 fcu\_Transfer\_Firmware 関数の戻り値一覧

| 型                            | 説明          |
|------------------------------|-------------|
| FCU_STATUS (* <sup>2</sup> ) | 関数実行時のステータス |

【注】 \*2 FCU\_STATUS 型の詳細は「5.7 enum 型説明」を参照してください。

(d) フローチャート

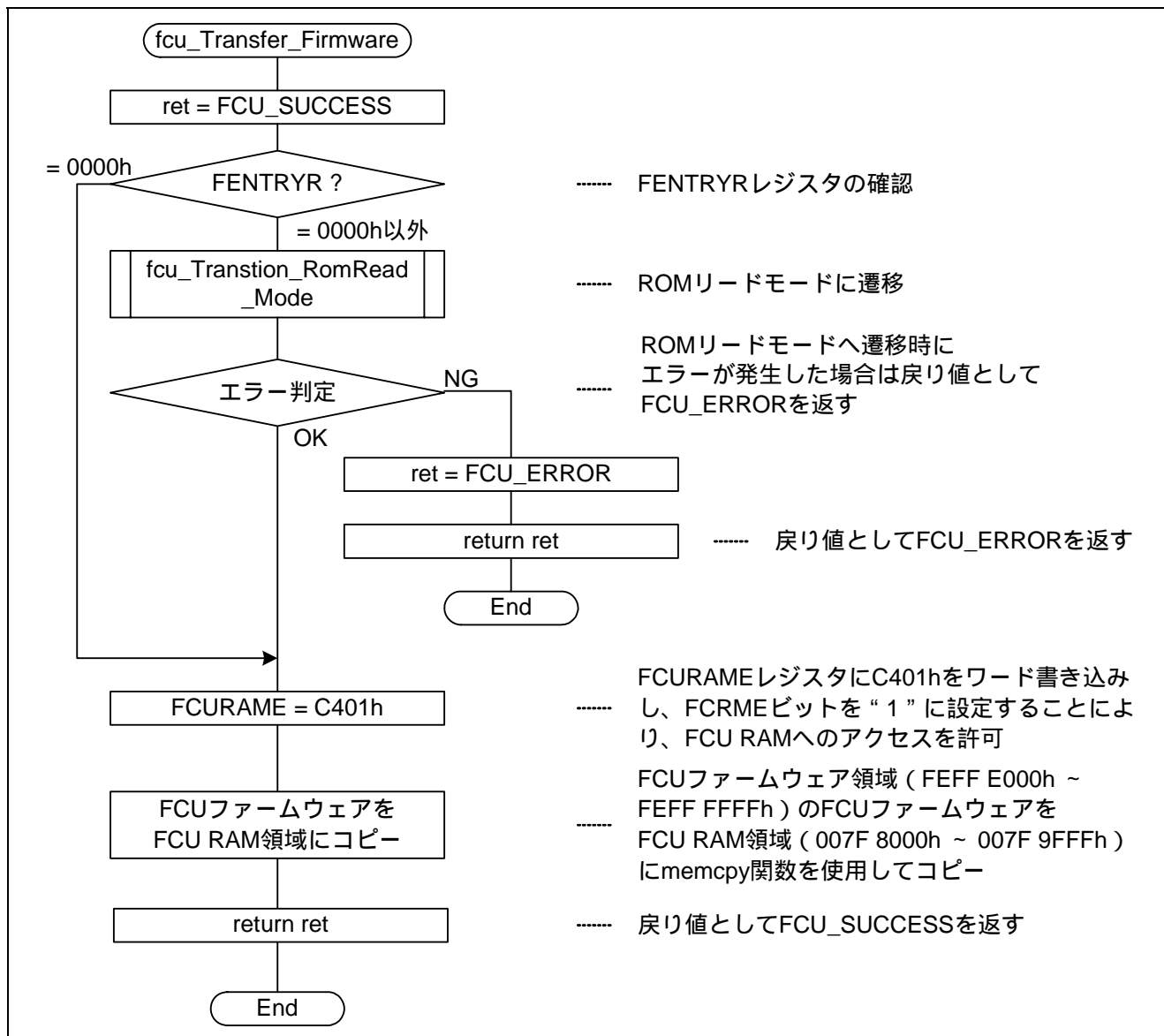


図 22 フローチャート (fcu\_Transfer\_Firmware) (スレーブ)



(8) fcu\_Transition\_RomRead\_Mode 関数

(a) 機能説明

fcu\_Transition\_RomRead\_Mode 関数は、FCU を ROM リードモードに遷移させます。

(b) 引数

表 24 に本関数で使用する引数を示します。

表 24 fcu\_Transition\_RomRead\_Mode 関数の引数一覧

| 引数     | 型                                  | 説明  |
|--------|------------------------------------|---|
| 第 1 引数 | ST_FCU_INFO *<br>(* <sup>1</sup> ) | ユーザマットの書き込み/消去時に使用する FCU 関連のアドレス情報を格納している構造体のアドレス |

【注】 \*1 ST\_FCU\_INFO 型の詳細は「5.6.1 構造体 ST\_FCU\_INFO」を参照してください。

(c) 戻り値

表 25 に本関数で使用する戻り値を示します。

表 25 fcu\_Transition\_RomRead\_Mode 関数の戻り値一覧

| 型                            | 説明          |
|------------------------------|-------------|
| FCU_STATUS (* <sup>2</sup> ) | 関数実行時のステータス |

【注】 \*2 FCU\_STATUS 型の詳細は「5.7 enum 型説明」を参照してください。

(d) フローチャート

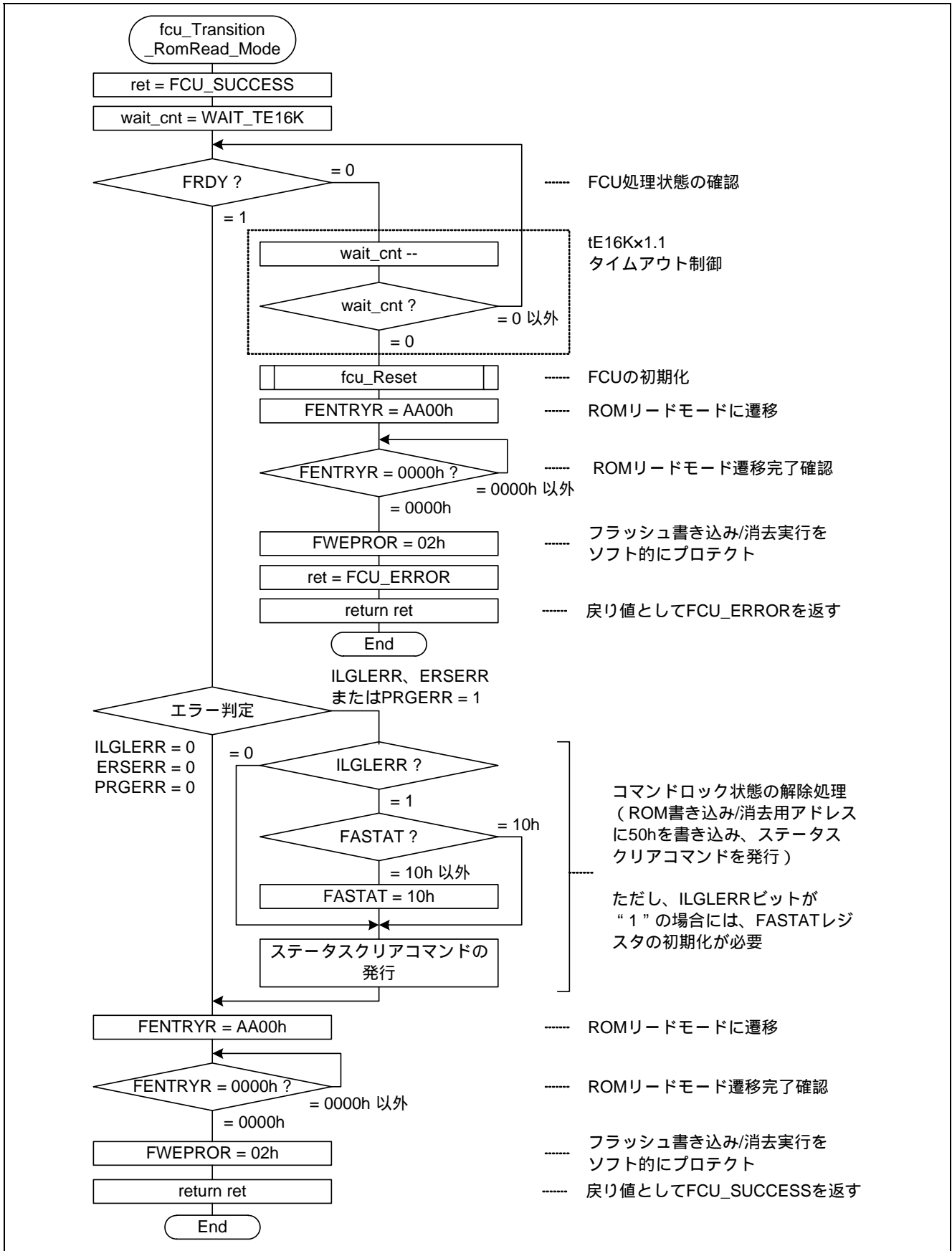


図 23 フローチャート (fcu\_Transition\_RomRead\_Mode) (スレーブ)

(9) fcu\_Transition\_RomPE\_Mode 関数

(a) 機能説明

fcu\_Transition\_RomPE\_Mode 関数は、FCU を ROM P/E モードに遷移させます。

(b) 引数

表 26 に本関数で使用する引数を示します。

表 26 fcu\_Transition\_RomPE\_Mode 関数の引数一覧

| 引数     | 型                                  | 説明  |
|--------|------------------------------------|---|
| 第 1 引数 | ST_FCU_INFO *<br>(* <sup>1</sup> ) | ユーザマットの書き込み/消去時に使用する FCU 関連のアドレス情報を格納している構造体のアドレス |

【注】 \*1 ST\_FCU\_INFO 型の詳細は「5.6.1 構造体 ST\_FCU\_INFO」を参照してください。

(c) 戻り値

表 27 に本関数で使用する戻り値を示します。

表 27 fcu\_Transition\_RomPE\_Mode 関数の戻り値一覧

| 型                            | 説明          |
|------------------------------|-------------|
| FCU_STATUS (* <sup>2</sup> ) | 関数実行時のステータス |

【注】 \*2 FCU\_STATUS 型の詳細は「5.7 enum 型説明」を参照してください。

(d) フローチャート

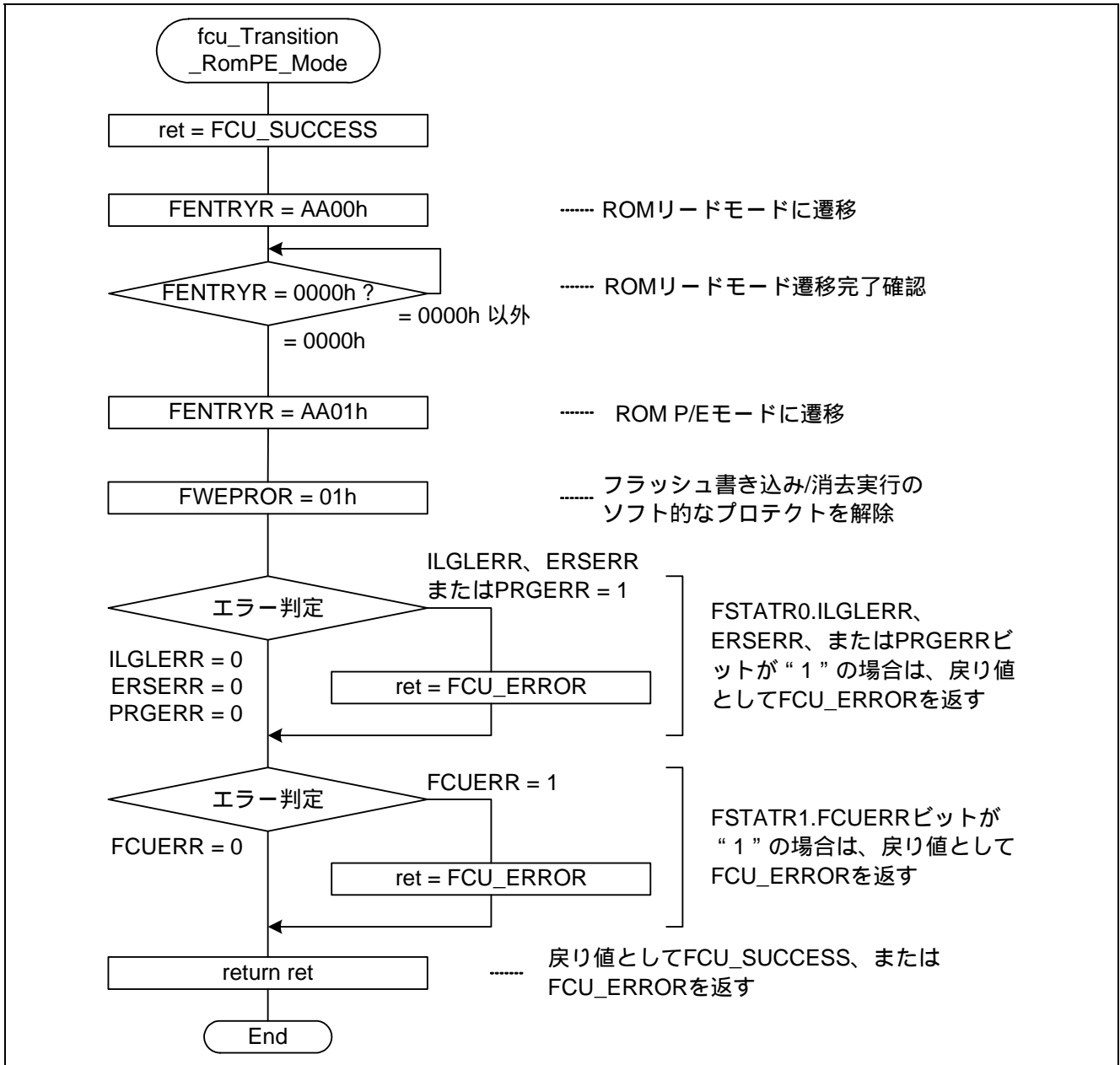


図 24 フローチャート (fcu\_Transition\_RomPE\_Mode) (スレーブ)

(10) fcu\_Notify\_Peripheral\_Clock 関数

(a) 機能説明

fcu\_Notify\_Peripheral\_Clock 関数は、周辺モジュールクロック (PCLK) の周波数を PCKAR レジスタに設定し、周辺クロック通知コマンドの発行を行います。

(b) 引数

表 28 に本関数で使用する引数を示します。

表 28 fcu\_Notify\_Peripheral\_Clock 関数の引数一覧

| 引数     | 型                                  | 説明  |
|--------|------------------------------------|---|
| 第 1 引数 | ST_FCU_INFO *<br>(* <sup>1</sup> ) | ユーザマットの書き込み/消去時に使用する FCU 関連のアドレス情報を格納している構造体のアドレス |

【注】 \*1 ST\_FCU\_INFO 型の詳細は「5.6.1 構造体 ST\_FCU\_INFO」を参照してください。

(c) 戻り値

表 29 に本関数で使用する戻り値を示します。

表 29 fcu\_Notify\_Peripheral\_Clock 関数の戻り値一覧

| 型                            | 説明          |
|------------------------------|-------------|
| FCU_STATUS (* <sup>2</sup> ) | 関数実行時のステータス |

【注】 \*2 FCU\_STATUS 型の詳細は「5.7 enum 型説明」を参照してください。

(d) フローチャート

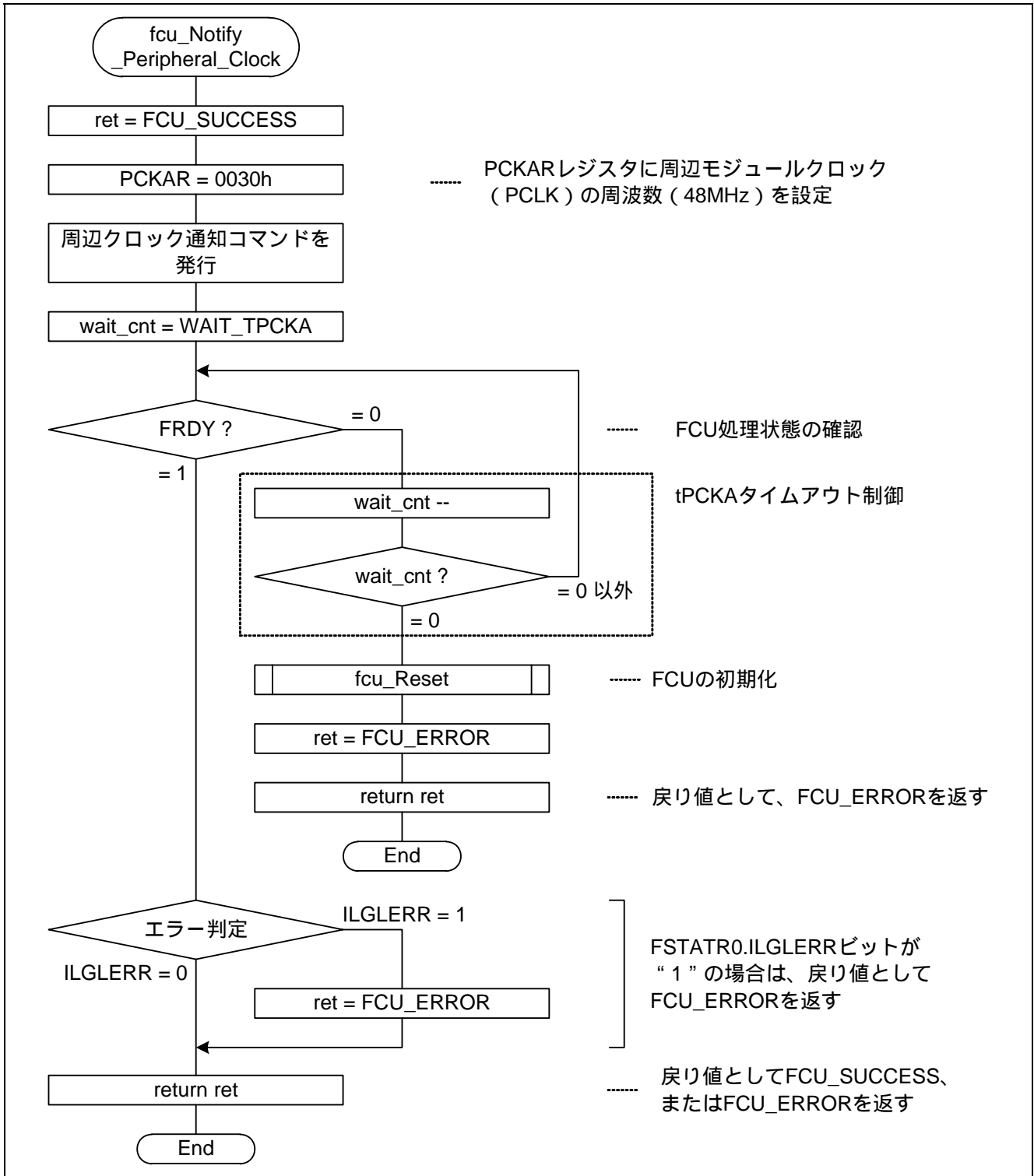


図 25 フローチャート (fcu\_Notify\_Peripheral\_Clock) (スレーブ)

(11) fcu\_Erase 関数

(a) 機能説明

fcu\_Erase 関数は、ブロックイレーズコマンドによりユーザマットの消去 (消去ブロック単位) を行います。

(b) 引数

表 30 に本関数で使用する引数を示します。

表 30 fcu\_Erase 関数の引数一覧

| 引数     | 型                                  | 説明  |
|--------|------------------------------------|---|
| 第 1 引数 | ST_FCU_INFO *<br>(* <sup>1</sup> ) | ユーザマットの書き込み/消去時に使用する FCU 関連のアドレス情報を格納している構造体のアドレス |

【注】 \*1 ST\_FCU\_INFO 型の詳細は「5.6.1 構造体 ST\_FCU\_INFO」を参照してください。

(c) 戻り値

表 31 に本関数で使用する戻り値を示します。

表 31 fcu\_Erase 関数の戻り値一覧

| 型                            | 説明          |
|------------------------------|-------------|
| FCU_STATUS (* <sup>2</sup> ) | 関数実行時のステータス |

【注】 \*2 FCU\_STATUS 型の詳細は「5.7 enum 型説明」を参照してください。

(d) フローチャート

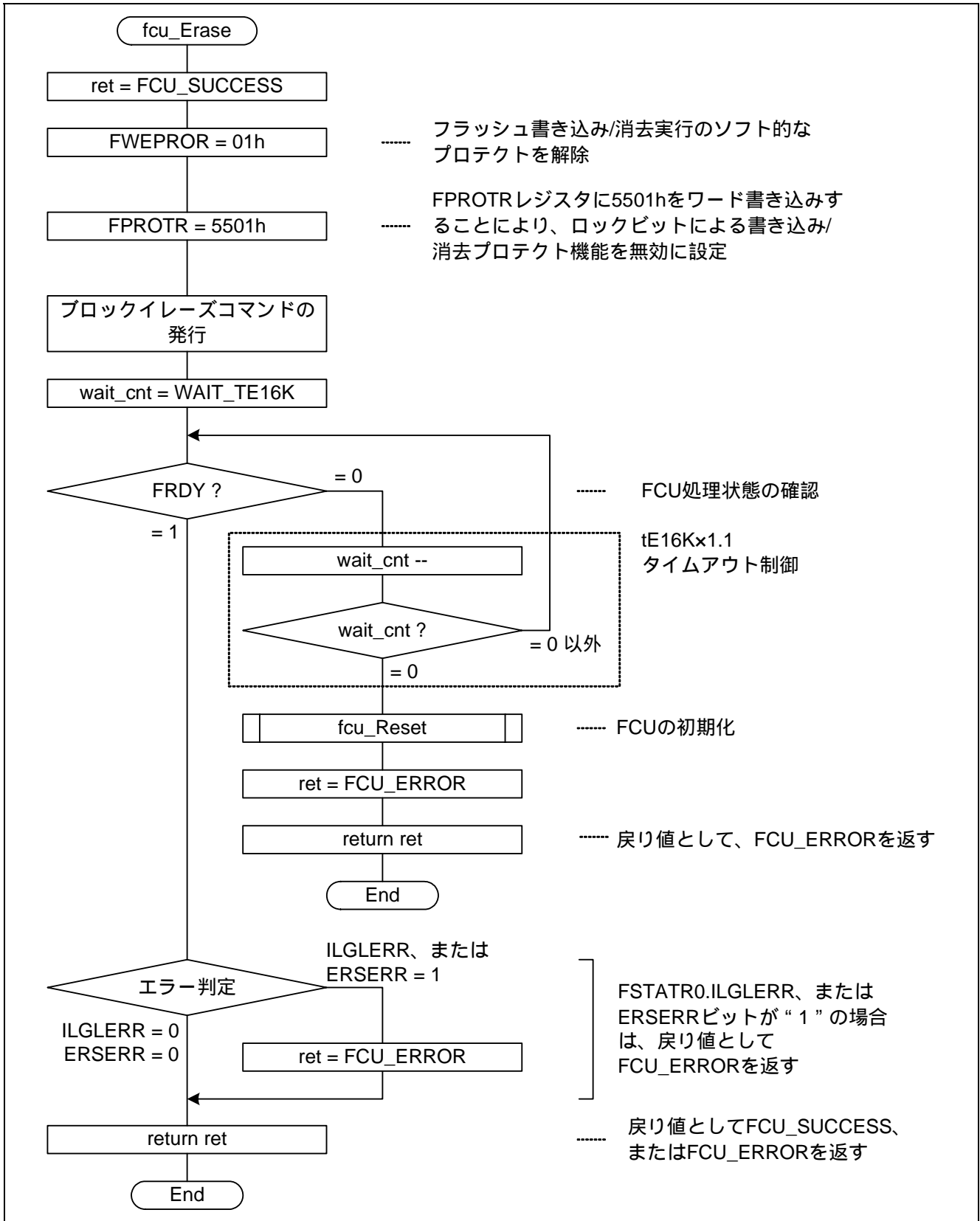


図 26 フローチャート (fcu\_Erase) (スレーブ)



(12) fcu\_Write 関数

(a) 機能説明

fcu\_Write 関数は、プログラムコマンドによりユーザマットの書き込み (256 バイト単位) を行います。

(b) 引数

表 32 に本関数で使用する引数を示します。

表 32 fcu\_Write 関数の引数一覧

| 引数     | 型                                 | 説明  |
|--------|-----------------------------------|---|
| 第 1 引数 | ST_FCU_INFO*<br>(* <sup>1</sup> ) | ユーザマットの書き込み/消去時に使用する FCU 関連のアドレス情報を格納している構造体のアドレス |

【注】 \*1 ST\_FCU\_INFO 型の詳細は「5.6.1 構造体 ST\_FCU\_INFO」を参照してください。

(c) 戻り値

表 33 に本関数で使用する戻り値を示します。

表 33 fcu\_Write 関数の戻り値一覧

| 型                            | 説明          |
|------------------------------|-------------|
| FCU_STATUS (* <sup>2</sup> ) | 関数実行時のステータス |

【注】 \*2 FCU\_STATUS 型の詳細は「5.7 enum 型説明」を参照してください。

(d) フローチャート

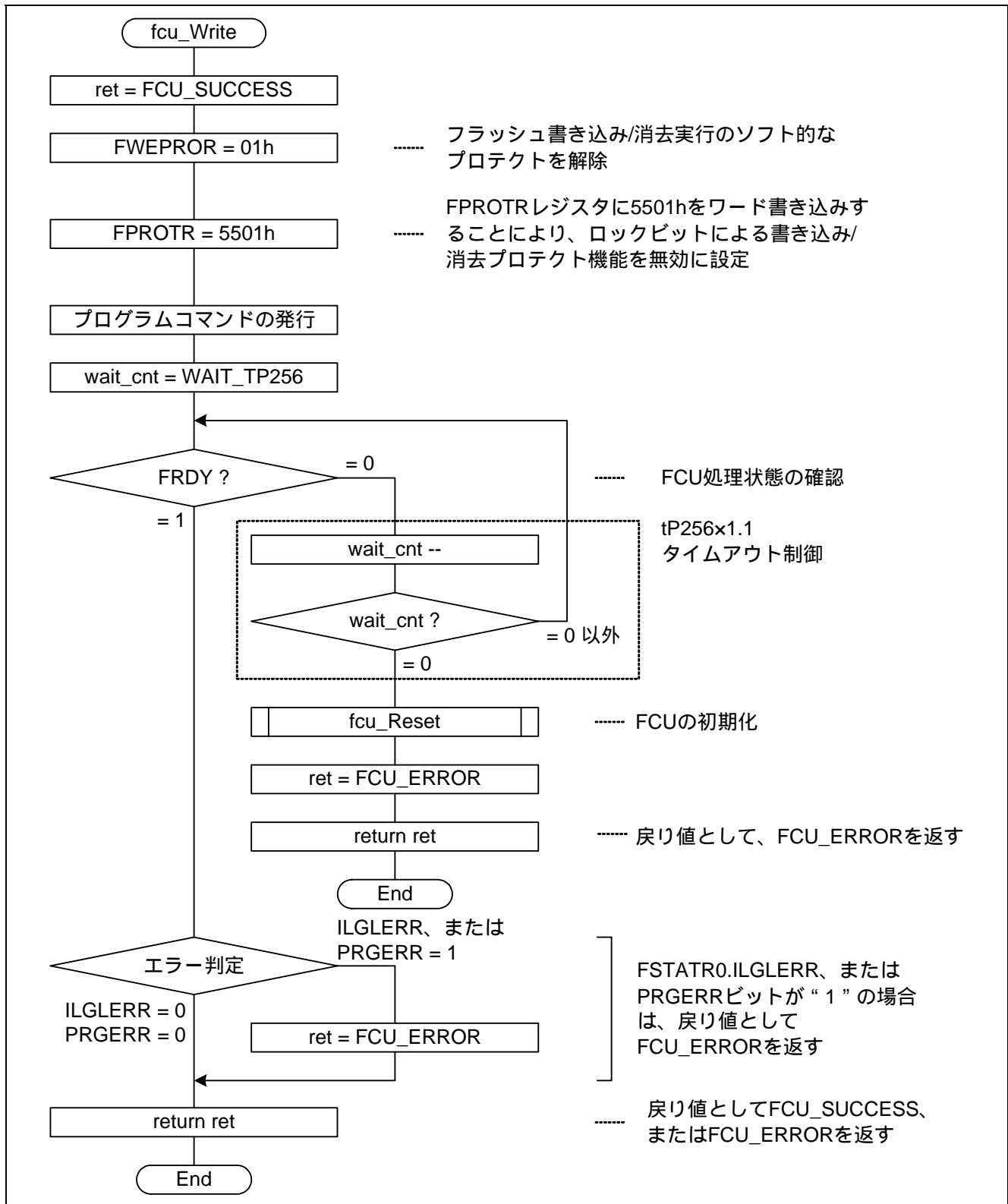


図 27 フローチャート (fcu\_Write) (スレーブ)

(13) Indicate\_Ending\_LED 関数

(a) 機能説明

Indicate\_Ending\_LED 関数は、書き込み/消去が正常に終了した場合に、LED0~LED3 に正常終了を示す表示を行います。LED0~LED3 を順番に1つずつ点灯させます。

(b) 引数

なし

(c) 戻り値

なし

(d) フローチャート

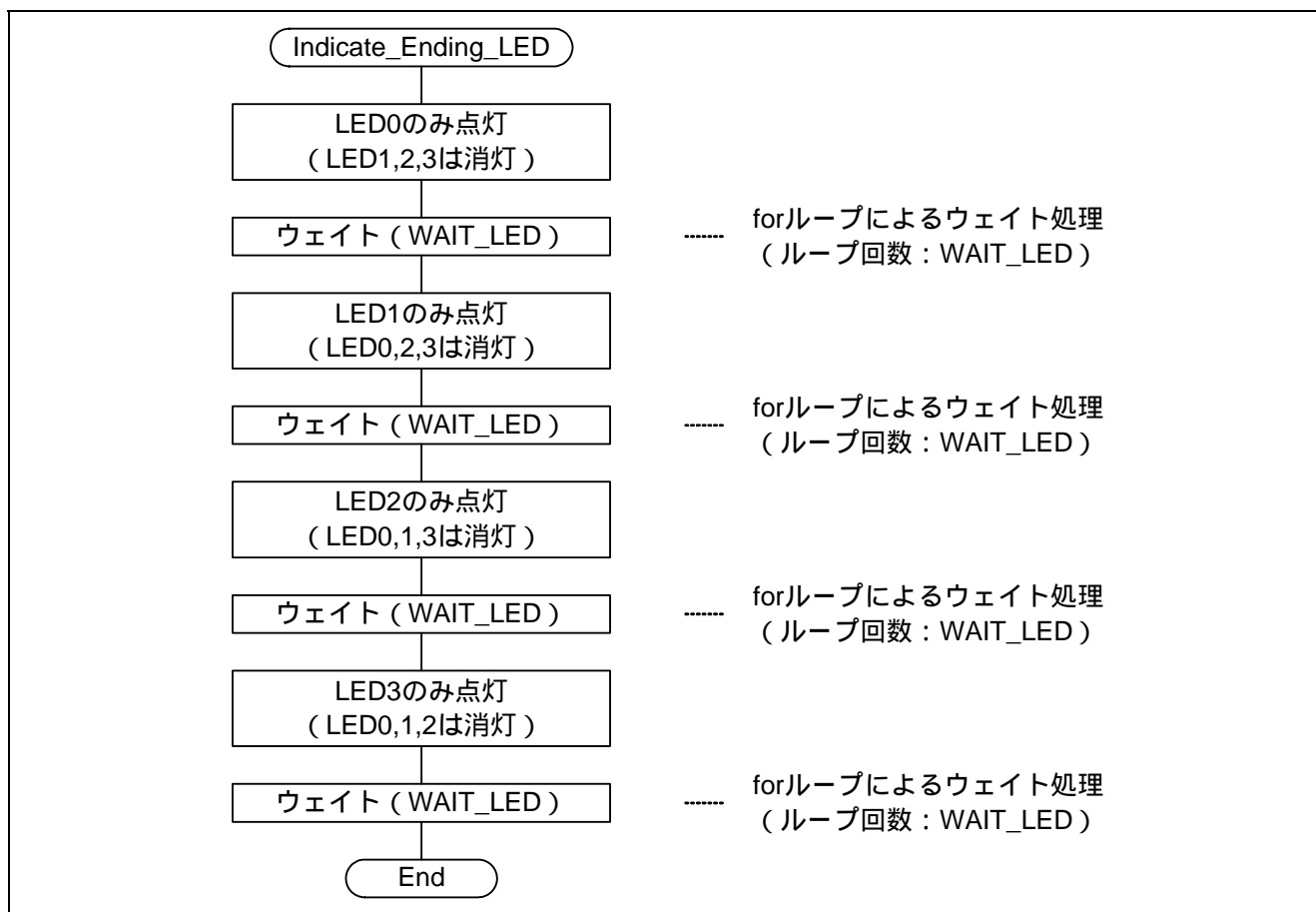


図 28 フローチャート (Indicate\_Ending\_LED) (スレーブ)

(14) Indicate\_Error\_LED 関数

(a) 機能説明

Indicate\_Error\_LED 関数は、ユーザマットの書き込み/消去の処理中にエラーが発生した場合に、LED0 ~ LED3 に発生したエラー番号の表示を行います。表示はエラー番号の表示と全消灯を繰り返します。

(b) 引数

表 34 に本関数で使用する引数を示します。

表 34 Indicate\_Error\_LED 関数の引数一覧

| 引数     | 型             | 説明                             |
|--------|---------------|--------------------------------|
| 第 1 引数 | unsigned char | ユーザマットの書き込み/消去中に発生したエラー番号 (*1) |

【注】 \*1 エラー番号は、「4.6 エラー処理」を参照してください。

(c) 戻り値

なし

(d) フローチャート

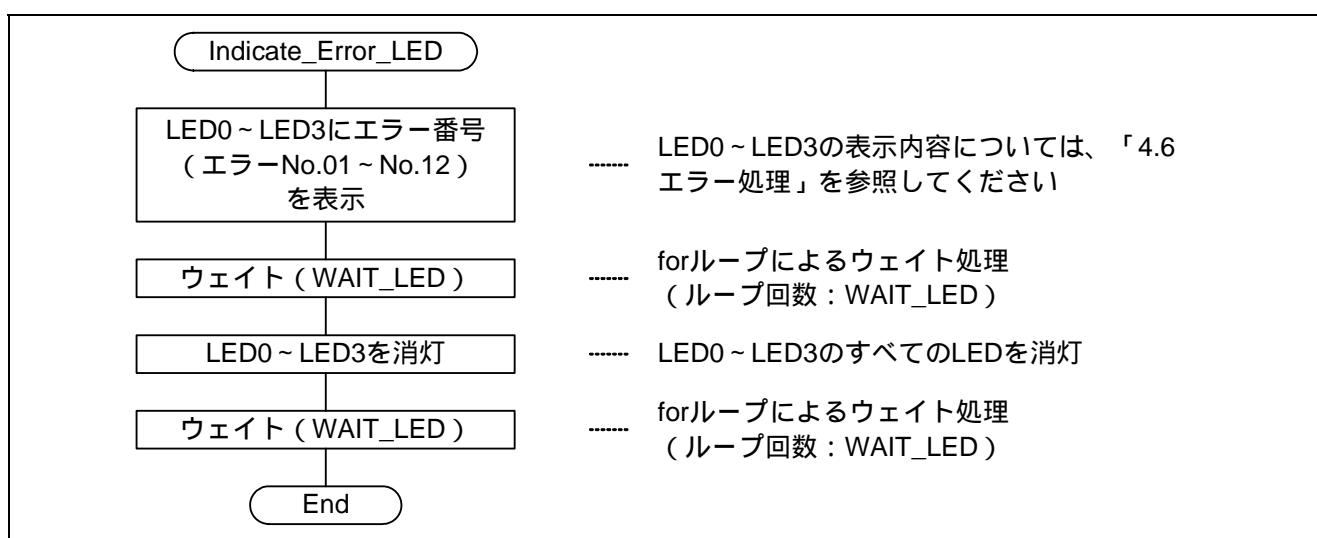


図 29 フローチャート (Indicate\_Error\_LED) (スレーブ)

(15) SCI\_Rcv1byte 関数

(a) 機能説明

SCI\_Rcv1byte 関数は、SCI2 のクロック同期式シリアル通信による 1 バイトデータの受信制御を行います。

(b) 引数

なし

(c) 戻り値

表 35 に本関数で使用する戻り値を示します。

表 35 SCI\_Rcv1byte 関数の戻り値一覧

| 型             | 説明                                |
|---------------|-----------------------------------|
| unsigned char | SCI2 のクロック同期式シリアル通信による 1 バイト受信データ |

(d) フローチャート

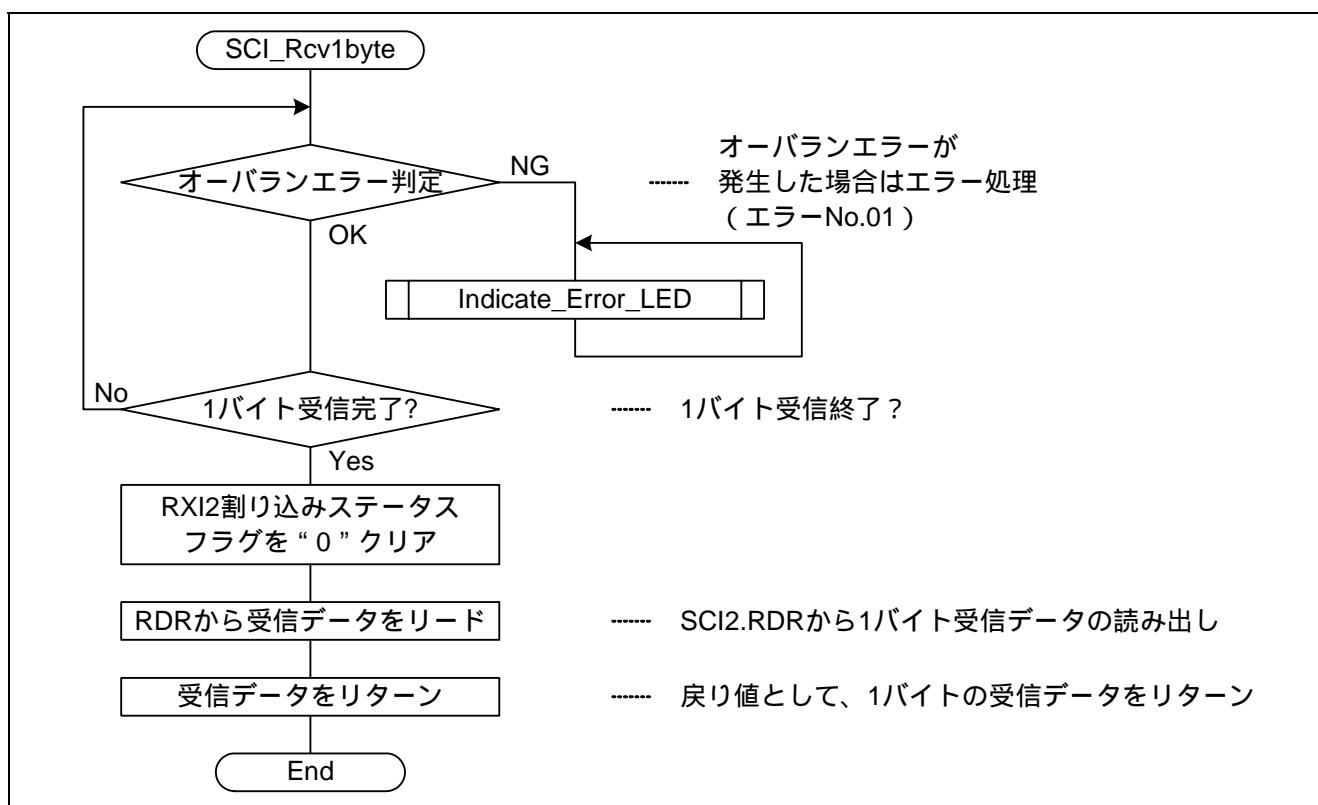


図 30 フローチャート (SCI\_Rcv1byte) (スレーブ)

(16) SCI\_Rcvnbyte 関数

(a) 機能説明

SCI\_Rcvnbyte 関数は、SCI2 のクロック同期式シリアル通信による n バイトデータ (n は unsigned short 型の第 1 引数) の受信制御を行います。

(b) 引数

表 36 に本関数で使用する引数を示します。

表 36 SCI\_Rcvnbyte 関数の引数一覧

| 引数     | 型               | 説明                              |
|--------|-----------------|---------------------------------|
| 第 1 引数 | unsigned short  | SCI2 のクロック同期式シリアル通信による受信データバイト数 |
| 第 2 引数 | unsigned char * | 受信データ格納場所の先頭アドレス                |

(c) 戻り値

なし

(d) フローチャート

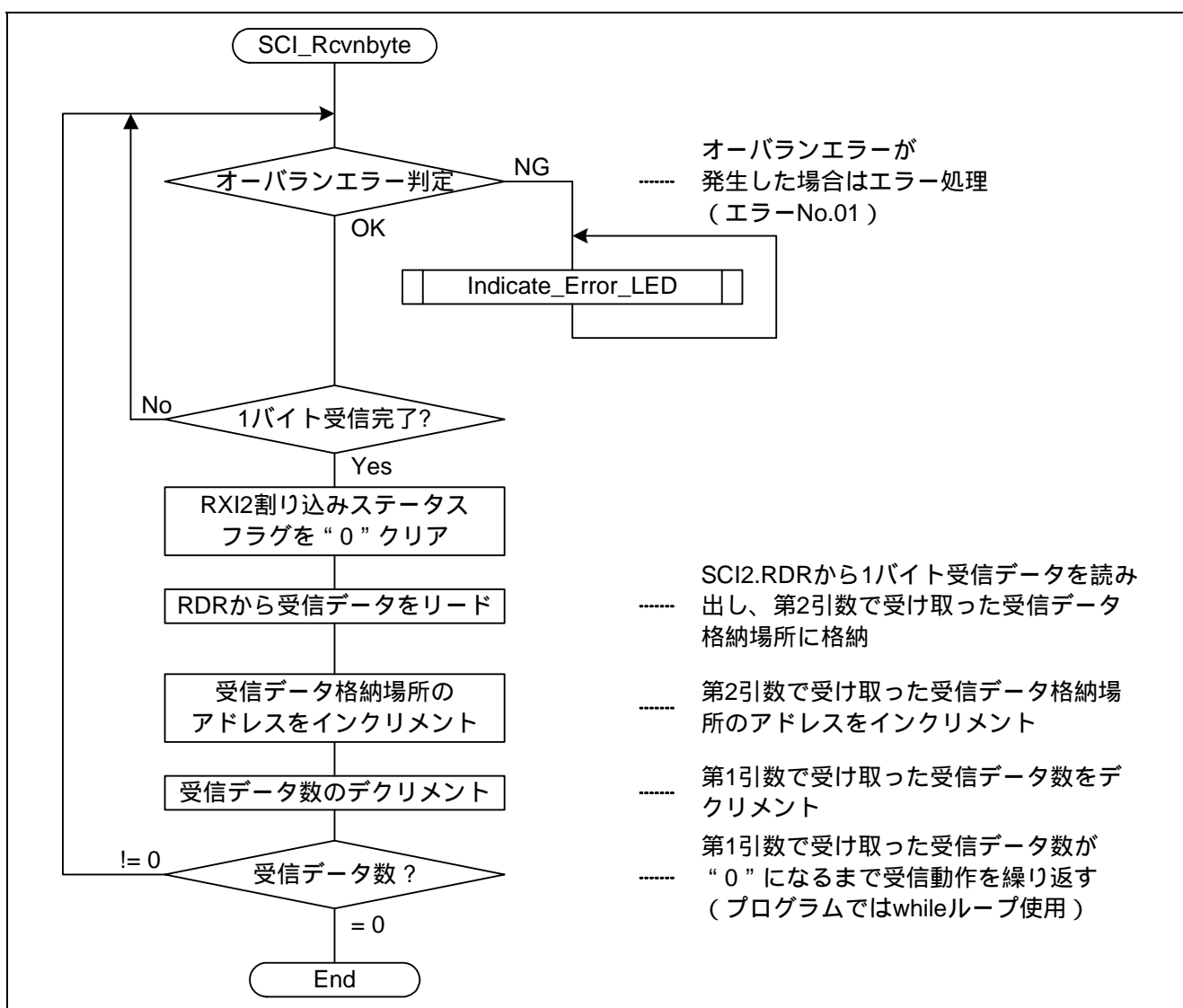


図 31 フローチャート (SCI\_Rcvnbyte) (スレーブ)

## 6. 使用上の注意事項

### 6.1 タイムアウトについて

本アプリケーションノートでは、ユーザマットの書き込み/消去中にタイムアウト制御を行っており、その時間計測にソフトウェアタイマを使用しています。

以下に使用しているタイムアウト制御について説明します。

#### 6.1.1 $t_{PCKA}$ タイムアウト制御

$t_{PCKA}$  タイムアウト制御は、FCU の周辺クロック通知コマンド発行時に使用します。本アプリケーションノートでは周辺クロック通知コマンド発行後、FSTATR0.FRDY ビットが"1"にセットされるまでに  $t_{PCKA}$  以上時間が経過すると、FCU の初期化を行い、エラー処理を実行します。

本アプリケーションノートでは、 $t_{PCKA}$  の待機時間を記号定数 WAIT\_TPCKA で定義したループ回数を while ループで回す処理を行っています。while ループを 1 回まわるサイクル数を 11 サイクル (コンパイラが出力するアセンブリ言語で確認できます) とすると、

$$\text{while ループの回る回数} = \text{待機時間} / (\text{while ループを 1 回まわるサイクル数} \times \text{ICLK サイクル時間})$$

で計算できます。なお、CPU の命令処理時間はパイプライン処理によって変動するため、上記の while ループを 1 回まわるサイクル数 (11 サイクル) は概算の命令処理時間になります。

$t_{PCKA}$  は PCLK = 50MHz で 60[ $\mu$ s]、PCLK = 25MHz で 120[ $\mu$ s]になります。本アプリケーションノートの  $t_{PCKA}$  は、PCLK = 48MHz のため 62.5[ $\mu$ s]になります。

本アプリケーションノートでは、マージンを取って待機時間を 187.5[ $\mu$ s]として計算しているため、

$$\text{while ループの回る回数} = \text{WAIT\_TPCKA} = 187.5[\mu\text{s}] / (11 \times 10.41666[\text{ns}]) = 1636 \text{ (ICLK} = 96\text{MHz 時)}$$

となります。したがって、記号定数 WAIT\_TPCKA を 1636 と定義して使用しています。

ユーザにて本アプリケーションノートをご使用の際は、CPU の命令処理時間の評価を十分に行っていたり、タイマを使用して時間を計測するようにしてください。

#### 6.1.2 $t_{RESW2}$ ウェイト制御

$t_{RESW2}$  ウェイト制御は、FCU の初期化時、FRESETR.FRESET ビットを"1"にセットしてから、"0"にクリアするまでの書き込み/消去中のリセットパルス幅 ( $t_{RESW2}$ ) をソフトウェアタイマにより制御しています。

表 37 に書き込み/消去中のリセットパルス幅を示します。

表 37 書き込み/消去中のリセットパルス幅

| 項目                         | 記号                            | min | max | 単位      | 測定条件 |
|----------------------------|-------------------------------|-----|-----|---------|------|
| 内部リセット時間 (* <sup>2</sup> ) | $t_{RESW2}$ (* <sup>1</sup> ) | 35  | —   | $\mu$ s | なし   |

【注】 \*1 本項目は、FCU リセット、WDT リセットに対する規定となります。

\*2 詳細は 7. 参考ドキュメント「ユーザズマニュアル」の「制御信号タイミング」を参照してください。

$t_{RESW2}$  の待機時間は、記号定数 WAIT\_TRESW2 で定義したループ回数を while ループで回す処理を行っています。while ループを 1 回まわるサイクル数を 4 サイクル (コンパイラが出力するアセンブリ言語で確認できます) とすると、

$$\text{while ループの回る回数} = \text{待機時間} / (\text{while ループを 1 回まわるサイクル数} \times \text{ICLK サイクル時間})$$

で計算できます。

なお、CPU の命令処理時間はパイプライン処理によって変動するため、上記の while ループを 1 回まわるサイクル数 (4 サイクル) は概算の命令処理時間となります。

本アプリケーションノートでは、マージンをとって待機時間 ( $t_{RESW2}$ ) を 105[ $\mu$ s]として計算しているため、  

$$\text{while ループの回る回数} = \text{WAIT\_TRESW2} = 105[\mu\text{s}] / (4 \times 10.41666 [\text{ns}]) = 2520 \text{ (ICLK} = 96\text{MHz 時)}$$

となります。したがって、記号定数 WAIT\_TRESW2 を 2520 と定義して使用しています。

ユーザにて本アプリケーションノートをご使用の際には、CPU の命令処理時間の評価を十分に行っていただくか、タイマを使用して時間を計測するようにしてください。

### 6.1.3 $t_{E16K} \times 1.1$ タイムアウト制御

$t_{E16K} \times 1.1$  タイムアウト制御は、FCU の ROM リードモード遷移時、およびユーザマットの消去時に使用しています。ROM リードモード遷移時は、FENTRYR レジスタに AA00h を書き込んで ROM リードモードに遷移する前に FSTATR0.FRDY ビットが"1"になるまでの 16K バイトの消去ブロックに対する消去時間をソフトウェアタイマにより計測します。また消去時は、ブロックイレースコマンドを発行してから FSTATR0.FRDY ビットが"1"になるまでの 16K バイトの消去ブロックに対する消去時間をソフトウェアタイマにより計測します。

表 38 に 16K バイトの消去ブロックに対する消去時間を示します。

表 38 16K バイトの消去ブロックに対する消去時間

| 項目        | 記号                 | min | typ | max | 単位 | 測定条件                                 |
|-----------|--------------------|-----|-----|-----|----|--------------------------------------|
| 消去時間 (*1) | 16KB<br>$t_{E16K}$ | —   | 100 | 240 | ms | PCLK = 50MHz 時<br>ブロックごとの消去回数 100 回時 |

【注】 \*1 詳細は 7. 参考ドキュメント「ユーザーズマニュアル」の「ROM (コード格納用フラッシュメモリ) 特性」を参照してください。

$t_{E16K} \times 1.1$  の待機時間は、記号定数 WAIT\_TE16K で定義したループ回数を while ループで回す処理を行っています。while ループを 1 回まわるサイクル数を 10 サイクル (コンパイラが出力するアセンブリ言語で確認できます) とすると、

$$\text{while ループの回る回数} = \text{待機時間} / (\text{while ループを 1 回まわるサイクル数} \times \text{ICLK サイクル時間})$$

で計算できます。なお、CPU の命令処理時間はパイプライン処理によって変動するため、上記の while ループを 1 回まわるサイクル数 (10 サイクル) は概算の命令処理時間となります。

本アプリケーションノートでは、マージンをとって待機時間 ( $t_{E16K} \times 1.1$ ) を 793[ms]として計算しているため、

$$\text{while ループの回る回数} = \text{WAIT\_TE16K} = 793[\text{ms}] / (10 \times 10.41666 [\text{ns}]) = 7603200 \text{ (ICLK} = 96\text{MHz 時)}$$

となります。したがって、WAIT\_TE16K を 7603200 と定義して使用しています。

ユーザにて本アプリケーションノートをご使用の際には、CPU の命令処理時間の評価を十分に行っていただくか、タイマを使用して時間を計測するようにしてください。



### 6.1.4 $t_{p256} \times 1.1$ タイムアウト制御

$t_{p256} \times 1.1$  タイムアウト制御は、ユーザマットの書き込み時に使用しています。プログラムコマンドを発行してから FSTAT0.FRDY ビットが"1"になるまでの 256 バイトの書き込み時間をソフトウェアタイマにより計測します。

表 39 に 256 バイトの書き込み時間を示します。

表 39 256 バイトの書き込み時間

| 項目          | 記号                 | min | typ | max | 単位 | 測定条件                                 |
|-------------|--------------------|-----|-----|-----|----|--------------------------------------|
| 書き込み時間 (*1) | 256B<br>$t_{p256}$ | —   | 2   | 12  | ms | PCLK = 50MHz 時<br>ブロックごとの消去回数 100 回時 |

【注】 \*1 詳細は 7. 参考ドキュメント「ユーザーズマニュアル」の「ROM (コード格納用フラッシュメモリ) 特性」を参照してください。

$t_{p256} \times 1.1$  の待機時間は、記号定数 WAIT\_TP256 で定義したループ回数を while ループで回す処理を行っています。while ループを 1 回まわるサイクル数を 11 サイクル (コンパイラが出力するアセンブリ言語で確認できます) とすると、

$$\text{while ループの回る回数} = \text{待機時間} / (\text{while ループを 1 回まわるサイクル数} \times \text{ICLK サイクル時間})$$

で計算できます。なお、CPU の命令処理時間はパイプライン処理によって変動するため、上記の while ループを 1 回まわるサイクル数 (11 サイクル) は概算の命令処理時間となります。

本アプリケーションノートでは、マージンをとって待機時間 ( $t_{p256} \times 1.1$ ) を 39.6[ms]として計算しているため、

$$\text{while ループの回る回数} = \text{WAIT\_TP256} = 39.6[\text{ms}] / (11 \times 10.41666 [\text{ns}]) = 345600 (\text{ICLK} = 96\text{MHz 時})$$

となります。したがって、WAIT\_TP256 を 345600 と定義して使用しています。

ユーザにて本アプリケーションノートをご使用の際は、CPU の命令処理時間の評価を十分に行っていたく、タイマを使用して時間を計測するようにしてください。

## 6.2 ユーザブートモード時の固定ベクタについて

MD1、MD0 端子によりユーザブートモードに設定しリセット解除を行うと、ユーザブートモードに遷移します。このときのリセットベクタは、ユーザブートマットの FF7F FFFCh 番地になります。その他のベクタテーブルは、通常のベクタテーブルを参照します。

固定ベクタテーブルは、テーブルの配置アドレスが固定されたベクタテーブルです。FFFF FFD0h ~

FFFF FFFCh 番地に、特権命令例外、未定義命令例外、浮動小数点例外、ノンマスカブル割り込み、リセットの各ベクタを配置しています。

図 32 に本アプリケーションノートにおける固定ベクタテーブルの設定方法を示します。

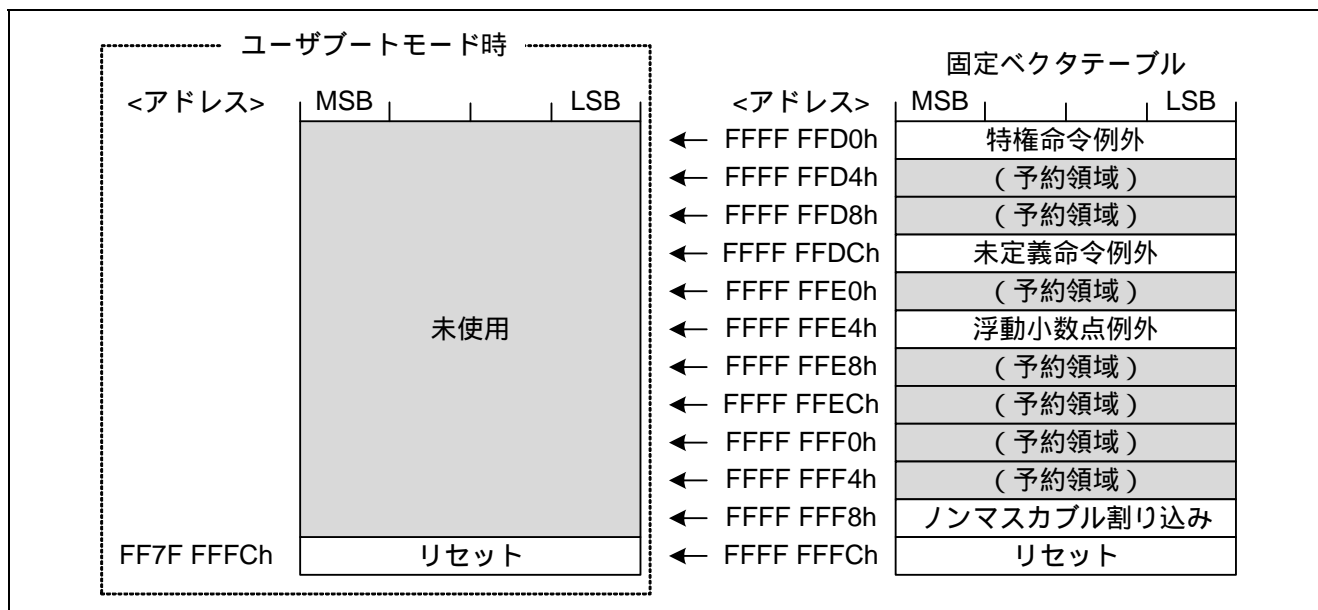


図 32 本アプリケーションノートにおける固定ベクタテーブル設定方法

本アプリケーションノートのスレーブは、ユーザブートモードで動作します。スレーブの固定ベクタの設定は、High-performance Embedded Workshop が自動生成するファイル (vecttbl.c、vect.h、intprg.c) の固定ベクタテーブルの特権命令例外 (Excep\_SuperVisorInst 関数)、未定義命令例外 (Excep\_UndefinedInst 関数)、浮動小数点例外 (Excep\_FloatingPoint 関数)、ノンマスクابل割り込み (NonMaskableInterrupt)、および予約領域の Dummy 関数をコメントアウトして、リセットだけの固定ベクタ (4 バイト) として、ユーザブートマットの FF7F FFFC 番地に配置しています。

なお、本アプリケーションノートのスレーブプログラムは、ユーザブートマット上のリセットベクタの設定のみを行っており、ユーザマットの固定ベクタの設定は行っておりません。

ユーザにて本アプリケーションノートのスレーブプログラムを使用される場合には、ユーザマットの固定ベクタ (特権命令例外、未定義命令例外、浮動小数点例外、ノンマスクابل割り込み、予約領域)、および各例外ハンドラを設定する必要がありますので、ご注意ください。

### 6.3 消去ブロック EB00 の書き換え時の注意事項

消去ブロック EB00 (書き込み/消去用アドレス: 00FF F000h ~ 00FF FFFF、読み出し用アドレス: FFFF F000h ~ FFFF FFFh) には、固定ベクタ (FFFF FF80h ~ FFFF FFFh)、ID コードプロテクト (FFFF FFA0h ~ FFFF FFAFh) などが配置されています。

消去ブロック番号を EB00\_INDEX に設定して、EB00 の書き込み/消去を行うと、上記の固定ベクタ、ID コードプロテクトは一度消去されてしまいます。したがって、EB00 の消去後に再度、固定ベクタ、ID コードプロテクトの設定を行う必要がありますので、ご注意ください。

ID コードプロテクトは、ホストからの読み出し/書き込み/消去を禁止するための機能で、ROM 上に書かれている制御コードおよび ID コードを使い、ID コードプロテクトの判定を行います。ID コードプロテクトの詳細については、7. 参考ドキュメント「ユーザーズマニュアル」を参照してください。

## 7. 参考ドキュメント

- ユーザーズマニュアル  
RX62N グループ、RX621 グループ ユーザーズマニュアル ハードウェア編 (R01UH0033JJ)  
(最新版をルネサス エレクトロニクスホームページから入手してください。)
- RX ファミリ ユーザーズマニュアル ソフトウェア編 (RJJ09B0465)  
(最新版をルネサス エレクトロニクスホームページから入手してください。)
- 開発環境マニュアル  
RX ファミリ用 C/C++コンパイラパッケージ ユーザーズマニュアル (RJJ10J2570)  
(最新版をルネサス エレクトロニクスホームページから入手してください。)
- アプリケーションノート  
RX62N グループ、RX621 グループ  
ユーザブートモードによる内蔵フラッシュメモリ書き換え (マスタ) (R01AN0185JJ)  
(最新版をルネサス エレクトロニクスホームページから入手してください。)
- テクニカルアップデート  
(最新版をルネサス エレクトロニクスホームページから入手してください。)

## 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://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

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。

## 改訂記録

| Rev. | 発行日        | 改訂内容          |   |
|------|------------|---------------|---|
|      |            | ページ           | ポイント  |
| 1.00 | 2010.12.17 | —             | 初版発行  |
| 1.01 | 2011.09.02 | 17<br>27<br>— | volatile __evenaccess 宣言追加<br>4.9.3 FCU コマンド修正<br>表 19 修正<br>ソースファイル(main.c)修正          |
| 1.02 | 2012.2.29  | 50<br>52      | 図 23 修正 (ROM リードモード遷移完了確認処理を追加)<br>図 24 修正 (ROM リードモード遷移完了確認処理を追加)<br>ソースファイル(main.c)修正 |
|      |            |               |   |
|      |            |               |   |
|      |            |               |   |
|      |            |               |   |
|      |            |               |   |
|      |            |               |   |

## 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. 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（予約領域）のアクセス禁止

【注意】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、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）がありません。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っていると、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けますとともに、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事務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、誤りがな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、海中継器、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。



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/contact/>