

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以って NEC エレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事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、海中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

SH7058F グループ

0.18 μ m プロセス F-ZTAT マイコン SH7058F オンボード書き込み

目次

1.	0.18 μ m プロセス F-ZTAT の特徴	2
2.	0.35 μ m プロセス F-ZTAT との相違点	3
3.	消去／書き込み手続きプログラム概要	5
4.	オンボードプログラミングモード (SH7058F の場合)	7
4.1	SH7058F のフラッシュの特徴	7
4.1.1	2種類 のフラッシュメモリ内蔵	7
4.1.2	3種類 のオンボードプログラミングモード	8
4.2	ブートモード概要	9
4.3	ユーザプログラムモード概要	11
4.4	ユーザブートモード概要	14
5.	フラッシュ書き込み手順～SH7058F の場合	18
5.1	ブートモード書き込み手順	18
5.1.1	ビットレート問い合わせステータス	18
5.1.2	問い合わせ選択ステータス	19
5.1.3	消去／書き込みステータス	19
5.2	ユーザプログラムモード書き込み手順	21
5.3	ユーザブートモード書き込み手順	28
6.	ユーザ作成プログラム例	31
6.1	ユーザプログラムモードフラッシュ消去／書き込み手続きアプリケーションプログラム例	31
6.1.1	概要	31
6.1.2	詳細仕様	34
6.1.3	メモリマップ	37
6.2	ユーザブートモードフラッシュ消去／書き込み手続きアプリケーションプログラム例	38
6.2.1	概要	38
6.2.2	詳細仕様	42
7.	付録	45

1. 0.18 μ m プロセス F-ZTAT の特徴

F-ZTAT*マイコンは、オンボード(機器組み込み状態)でプログラムの消去/書き込みが可能なフラッシュメモリ内蔵マイコンです。オンボードでプログラム書き換えが可能なので、市場出荷後のソフトウェアのバージョンアップ、組み込み機器ごとのパラメータの最適化、メンテナンスなどを容易に行うことができます。

0.18 μ m プロセス F-ZTAT には、専用の消去/書き込みプログラムが内蔵されています。

消去/書き込みを行うときは、それらの内蔵モジュールを内蔵RAMにダウンロード後、所定の引数パラメータをユーザシステムにあわせ設定し、呼び出すだけで、消去/書き込みが自動で行えるようになりました。

消去は、消去ブロック単位で行い、一度の消去プログラム呼び出しで1ブロックの消去ができます。

書き込みは、128バイト単位で行い、一度の書き込みプログラム呼び出しで、128バイトの書き込みができます。

【注】 * F-ZTAT (Flexible Zero Turn Around Time) は (株) ルネサステクノロジの商標です。

2. 0.35 μm プロセス F-ZTAT との相違点

図2.1に0.35 μm プロセスF-ZTATの消去／書き込み、図2.2に0.18 μm プロセスF-ZTATの消去／書き込み方法を示します。

0.35 μm プロセスF-ZTATでは、消去／書き込みを行う場合、消去プログラム、書き込みプログラム、それらを制御する手続きプログラムのすべてを作成しなければなりません。

それに対し、0.18 μm プロセスF-ZTATでは、消去プログラム、書き込みプログラムが、マイコンに内蔵されているため、手続きプログラムを作成すればよいことになりました。

消去／書き込みの際には、手続きプログラムをRAM上に展開し実行します。手続きプログラムの処理内で、内蔵プログラム(初期化、消去、書き込み)をRAM上にダウンロードし、それらをサブルーチン呼び出しして、消去／書き込みを行います。

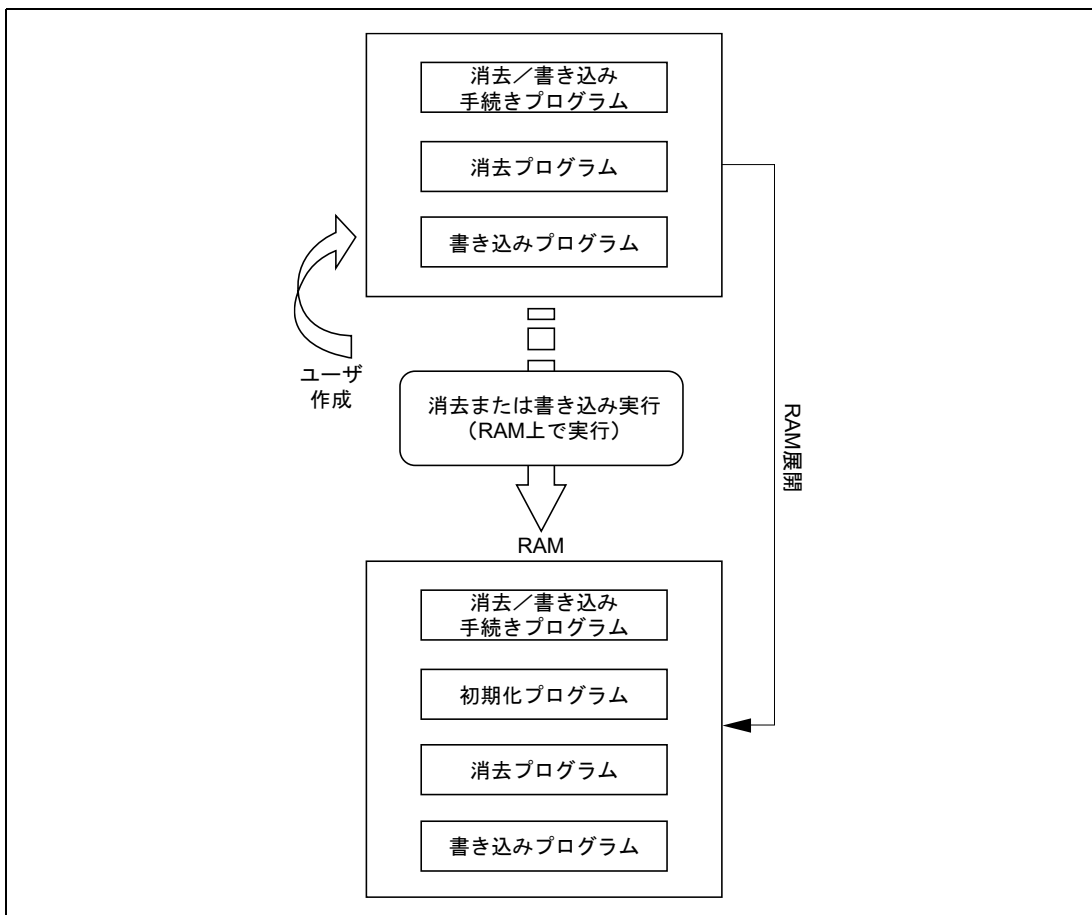


図 2.1 0.35 μm プロセス F-ZTAT の消去／書き込み

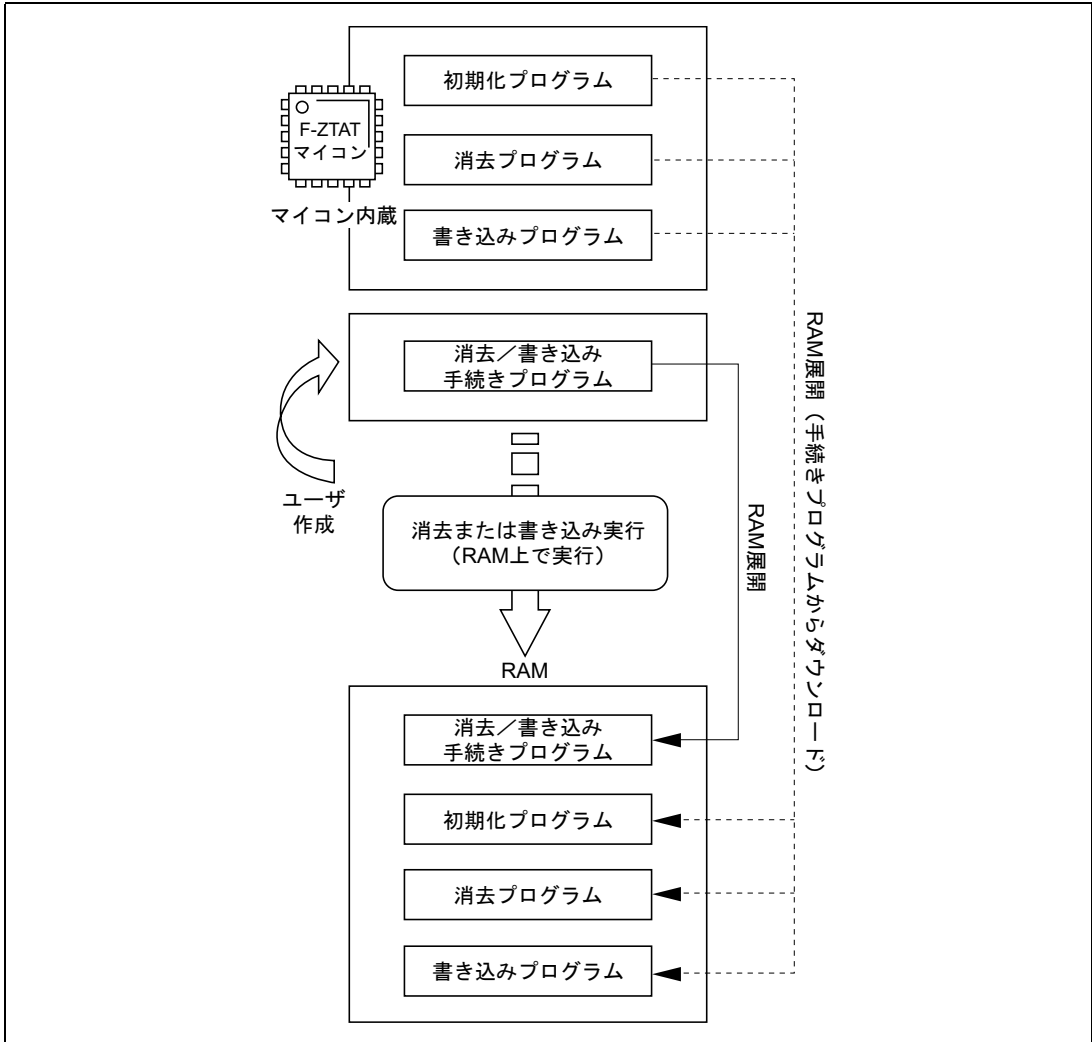


図 2.2 0.18 μ m プロセス F-ZTAT の消去/書き込み

3. 消去／書き込み手続きプログラム概要

0.18 μm プロセス F-ZTAT の消去／書き込み手続きプログラムは、図3.1に記載するシーケンスで作成します。

この図において、初期化の実行、消去実行、書き込み実行の各シーケンスはダウンロードした内蔵プログラムをサブルーチン呼び出しすることで実行します。

内蔵プログラムをダウンロードするときは、消去／書き込み、どちらの処理を実行するかにより、消去プログラムか、書き込みプログラムかを選択します。ただし、一度のダウンロードで、消去処理、書き込み処理をともに実行することはできません。

フラッシュを消去して書き込みを行う場合は、「消去プログラムのダウンロード～消去完了」、「書き込みプログラムのダウンロード～書き込み完了」と処理を2回に分けて行います。

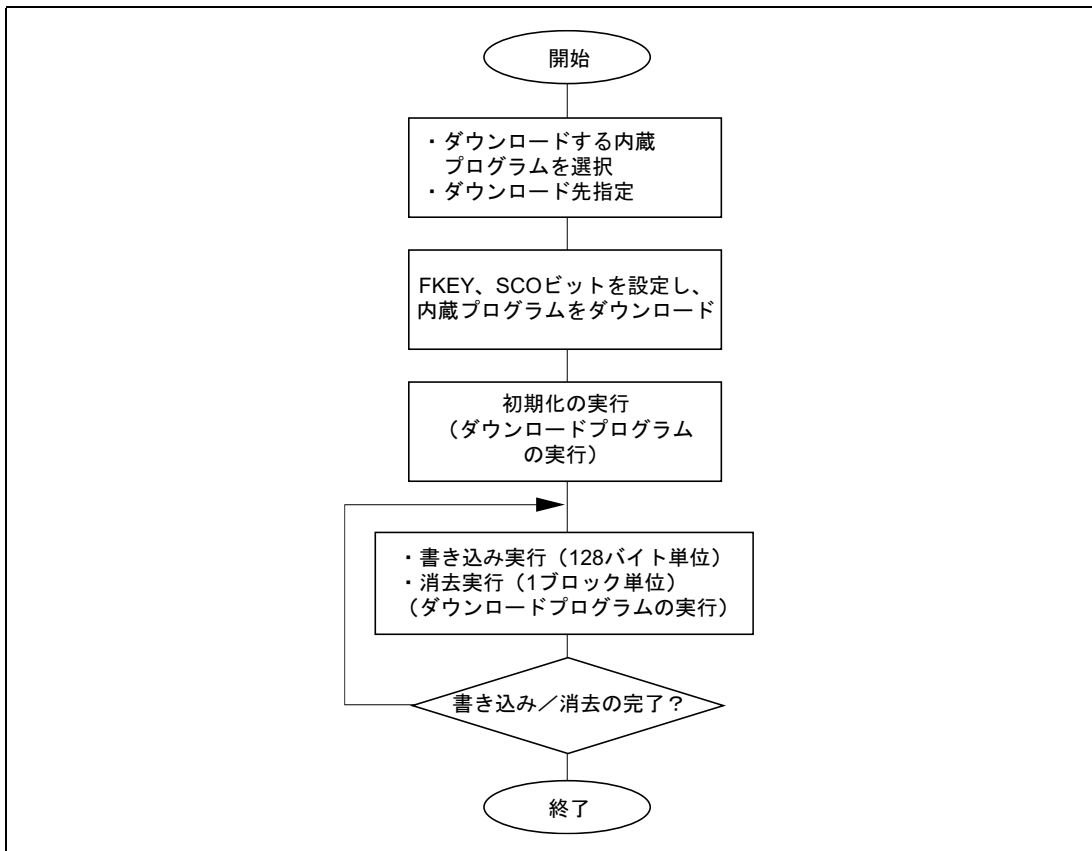


図 3.1 消去／書き込みプログラム概要

消去／書き込み手続きプログラムで実行される処理の概要を記述します。消去／書き込み手続きプログラムを作成する場合、これらの処理を組み込みます。

(1) ダウンロードのパラメータ指定

- ・ ダウンロード先の内蔵RAMアドレスの選択 (FTDAR)
- ・ ダウンロードプログラム(消去or書き込み)の選択 (FPCSのPPVSビット or FECSのEPVBビット)

(2) RAM 上への内蔵プログラムダウンロード

- フラッシュキーコードレジスタ(FKEY)にH'A5を設定
 - フラッシュコードコントロール・ステータスレジスタ(FCCS)を設定しダウンロード
- ダウンロード実行後、ダウンロードパス・フェイルリザルトパラメータ (DPFR) を参照し、エラーチェック

(3) 初期化の実行

ダウンロードした初期化プログラムを実行(サブルーチン呼び出し)します。初期化プログラムのエントリアドレスは、(FTDARで設定したアドレス+32)番地です。

初期化実行の際に設定するレジスタ、パラメータは、次のとおりです。

- CPUの動作周波数の設定 (FPEFEQ)
 - ユーザブランチアドレスの設定 (FUBRA) (未サポートLSIあり)
- 初期化実行後、フラッシュパス/フェイルリザルトパラメータ (FPFR) を参照し、エラーチェック

(4) 消去/書き込みの実行

ダウンロードした消去プログラム又は、書き込みプログラムを実行(サブルーチン呼び出し)することによって消去/書き込みは行われます。消去/書き込みプログラムのエントリアドレスは、(FTDARで設定したアドレス+16)番地です。

消去/書き込み実行の際に設定するレジスタ、パラメータは、次のとおりです。

1. 消去/書き込み共通のレジスタ設定
 - フラッシュキーコードレジスタ(FKEY)に H'5A を設定
 2. 書き込み時のパラメータ設定
 - フラッシュ ROM 書き込み先の先頭アドレス (FMPAR)
 - 書き込みデータ格納領域の先頭アドレス (FMPDR)
 3. 消去時のパラメータ設定
 - 消去ブロック番号 (FEBS)
- 消去/書き込み実行後、フラッシュパス/フェイルリザルトパラメータ (FPFR) を参照し、エラーチェック

4. オンボードプログラミングモード (SH7058F の場合)

0.18 μ m プロセス F-ZTAT マイコンの消去/書き込み方法を説明します。ここでは、0.18 μ m プロセス F-ZTAT マイコンである SH7058F を取り上げ、その特徴とマイコンが持つ 3 種類のオンボードプログラミングモードについて説明します。

4.1 SH7058F のフラッシュの特徴

初めに、0.18 μ m プロセス F-ZTAT マイコン SH7058F に関する特徴を述べます。

ここでは、SH7058F が持つ 2 種類のフラッシュメモリと、3 種類のオンボードプログラミングモードについて記述します。

4.1.1 2 種類のフラッシュメモリ内蔵

0.18 μ m プロセスの F-ZTAT は、フラッシュメモリの構成は、2 種類のメモリ空間からなっています。これらはメモリマットと呼ばれ、1MB のユーザマット、8kB のユーザブートマットから成り立ちます (図 4.1 参照)。

使用用途としては、ユーザマットは、通常、動作するアプリケーションプログラムを格納します。

ユーザブートマットは、後述するユーザブートモードで使用し、フラッシュ消去/書き込み手続きプログラムを格納します。

ユーザマット、ユーザブートマットの先頭アドレスは、同一アドレスであるため、2 つのマット間で実行したり、データアクセスがまたがる場合には必ずマットの切り替えを行わなければなりません。ユーザマットの書き込みは 128 バイト単位、消去は図 4.2 に示すブロック単位で行います。

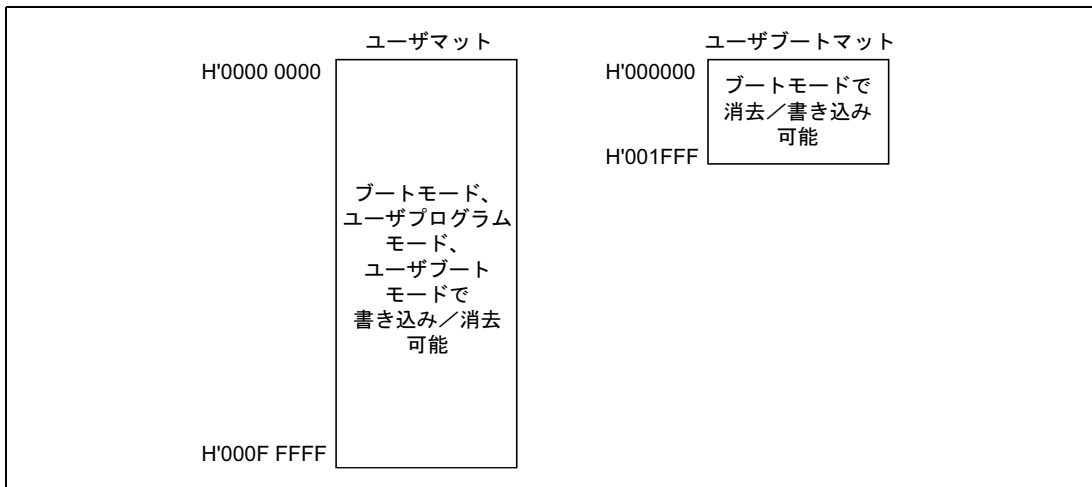


図 4.1 フラッシュメモリ構造

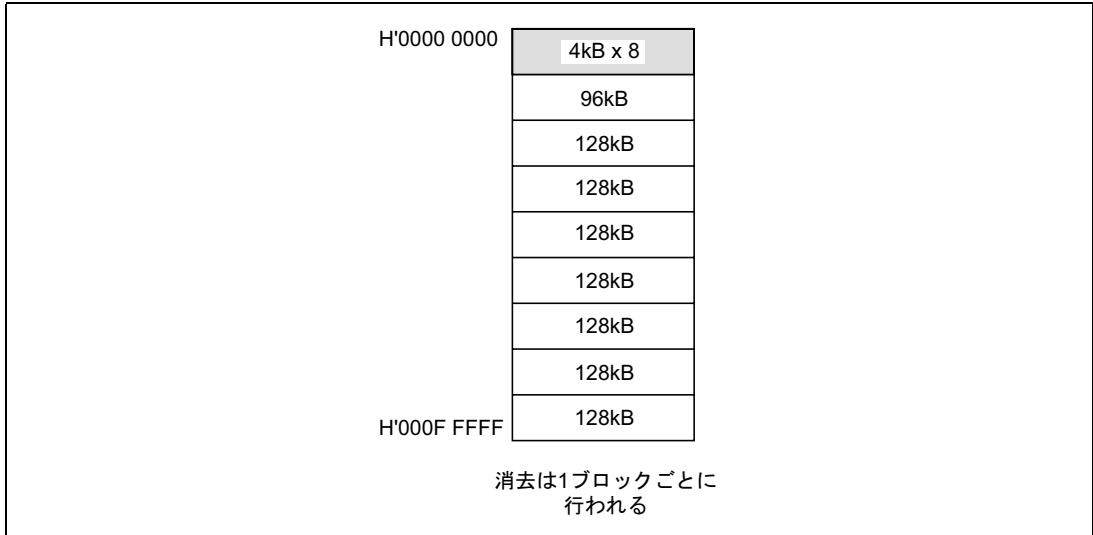


図 4.2 消去ブロック構造

4.1.2 3種類のオンボードプログラミングモード

フラッシュの消去／書き込みを行う方法には、オンボードプログラミングモードとして、次の3種類のモードが用意されています。これらのモードを選択するには、表4.1に示すとおり端子の設定を行うことで、任意のモードでの消去／書き込みを行うことができます。

- ブートモード
内蔵SCIインターフェースを使用します。ホストとF-ZTATマイコンに内蔵されたブートプログラムとの間でコマンド、データのやり取りを行い、ユーザマット、ユーザブートマットの消去／書き込みを行います。
- ユーザプログラムモード
任意のインターフェースでの、ユーザマットの書き換えが可能です。
- ユーザブートモード
ユーザ作成のユーザブートプログラムにより、任意のインターフェースでの、ユーザマットの書き換えが可能です。このモードは、ユーザブートマットから起動されるため、ユーザマットへの消去／書き込み処理では、マットの切り替えを行う必要があります。

表 4.1 モード別端子設定

端子	モード				
	リセット 状態	書き換えモード			アプリケーション 実行時
		ユーザプログラム モード	ユーザブート モード	ブートモード	
RES	0	1	1	1	1
FEW	0/1	1	1	1	0
MD0	0/1	1	1	1	0/1
MD1	0/1	0/1	0/1	0/1	0/1
MD2	0/1	1	0	0	0/1
NMI	0/1	0/1	0	1	0/1

4.2 ブートモード概要

ブートモードは、マイコン内蔵のSCIIを利用してホストから制御コマンドや書き込みデータを送受信する方式で、ユーザマットやユーザブートマットへの消去／書き込みを実行します。

ブートモード時のシステム構成を図4.3に示します。

表4.1に示したブートモードに各々端子設定し、リセットスタートすることでブートプログラムが起動します。ブートプログラムはSCIビットレートの自動調整後、ホストから送信される制御コマンドを受け、そのコマンドに対するレスポンスをホストに返信します。そして、受信コマンドが書き込みや消去などの処理を伴うコマンドであれば、それらのコマンド処理を実行していきます。

ホスト上には制御コマンドを送信するためのツールと書き込みデータを準備しておく必要があります。

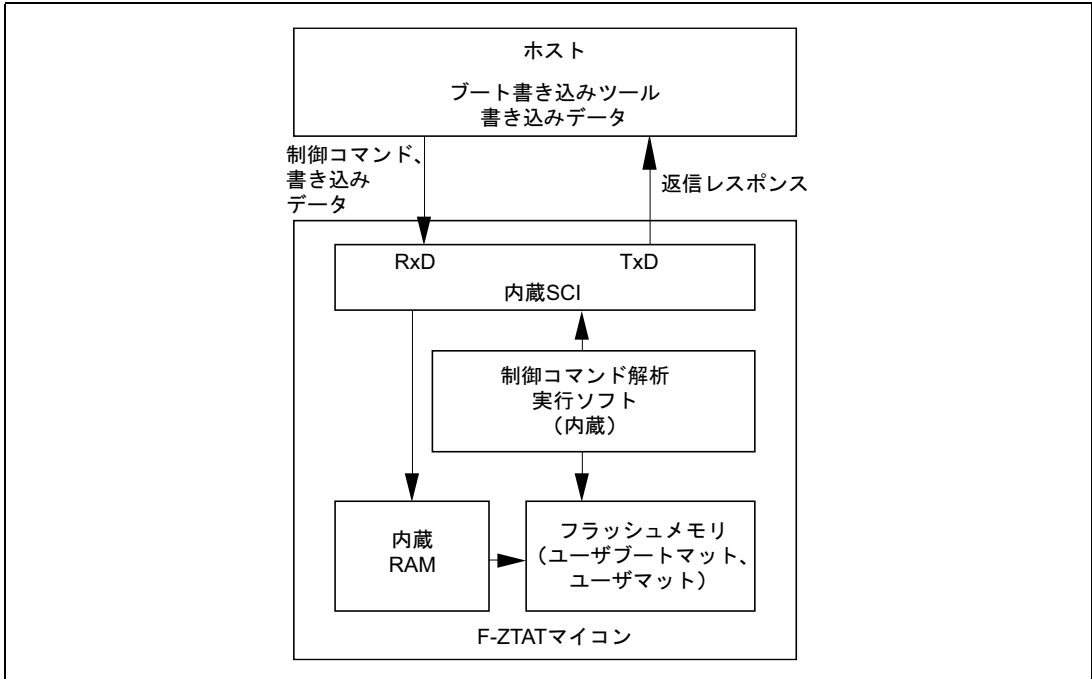


図 4.3 ブートモード時システム構成

ホストとブートプログラムのコマンドのやり取りを大きく分けると、次の3状態に分けることができます。1から3の順で遷移していきます。

1. ビットレート合せ込みステータス
2. 問い合わせステータス
3. 消去/書き込みステータス

これらのステータスの遷移については、図4.4を参照してください。

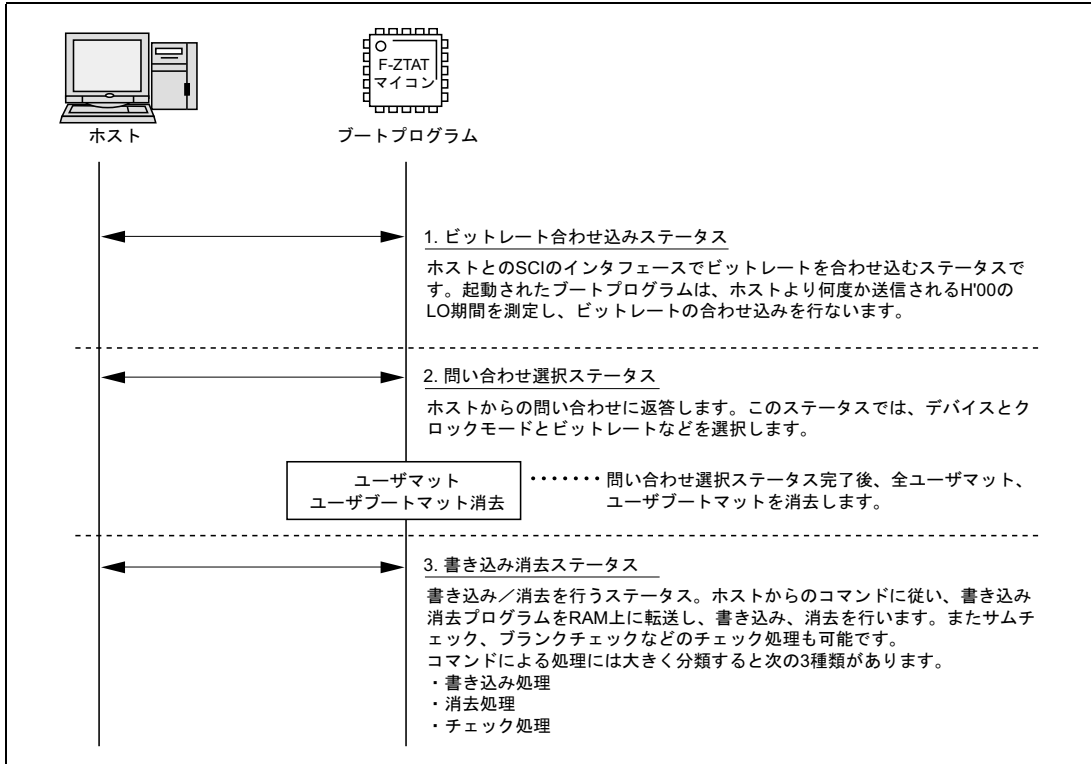


図 4.4 ブートモードステータス遷移

4.3 ユーザプログラムモード概要

ユーザプログラムモードでは、ユーザマツトの消去／書き込みが可能です。マイコンに内蔵されている消去／書き込みプログラムをダウンロードして使用します。内蔵プログラムのダウンロードの要求、消去／書き込みの制御、結果の判定など行う手続きプログラムをユーザマツト上に作成します。消去／書き込みを行う時には作成した手続きプログラムをRAM上に展開させ実行させます。ユーザプログラムモードの処理の流れを図4.5に示します。

(1) 手続きプログラムの RAM 展開

ユーザマツト上の手続きプログラムをRAM上に展開します。以降の制御は、すべてRAM上に展開した手続きプログラムにて行います。

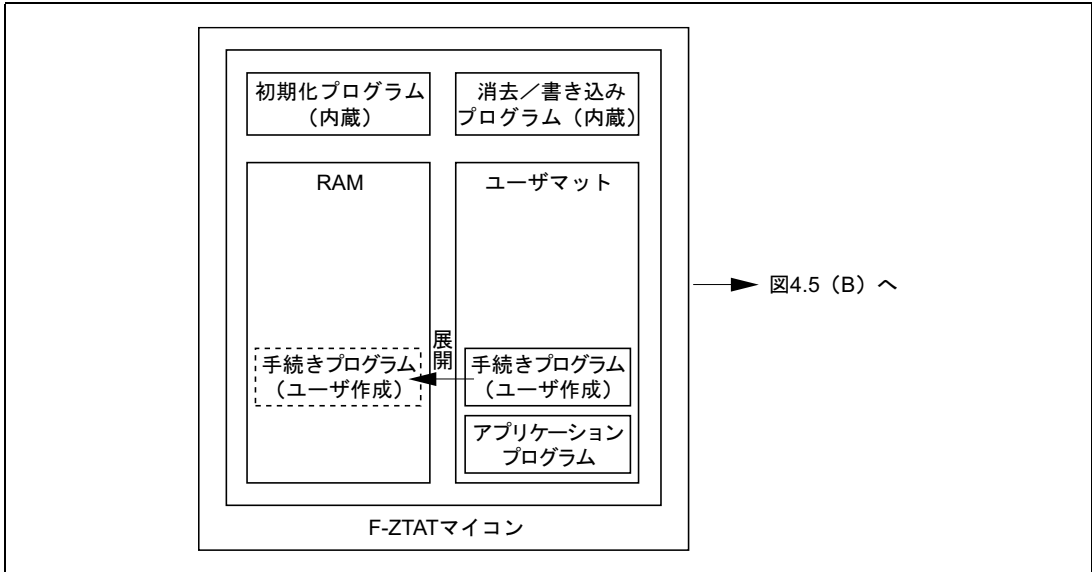


図 4.5 (A) 手続きプログラムの RAM 展開

(2) 内蔵プログラムのダウンロード

消去時には消去プログラムを、書き込み時には書き込みプログラムをRAM上にダウンロードします。同時に、初期化プログラムがダウンロードされます。

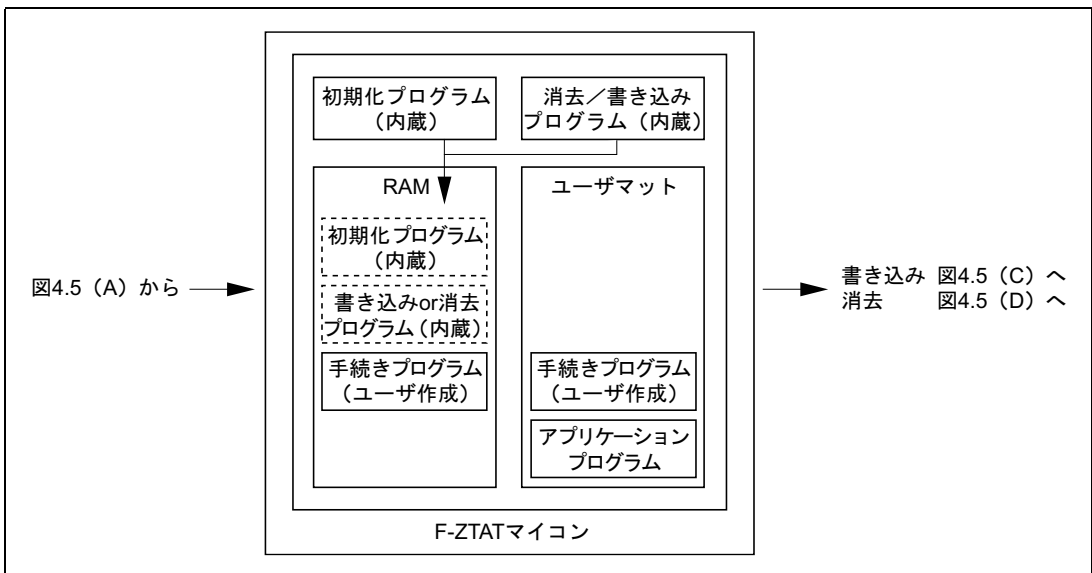


図 4.5 (B) 内蔵プログラムダウンロード

(3) ユーザマットの書き込み

RAM上にダウンロードした初期化プログラムをサブルーチン呼び出し、書き込みに必要な初期化処理を実行します。

↓ 初期化終了

RAM上にダウンロードした書き込みプログラムをサブルーチン呼び出しすることで、ユーザマットの書き込みを行います。このとき、ユーザマットへの書き込みは128バイト単位で行われるので、書き込みプログラムの呼び出しは128バイトごとに行い、すべての書き込みデータ(アプリケーションプログラム)を書き込み完了するまで続けます。

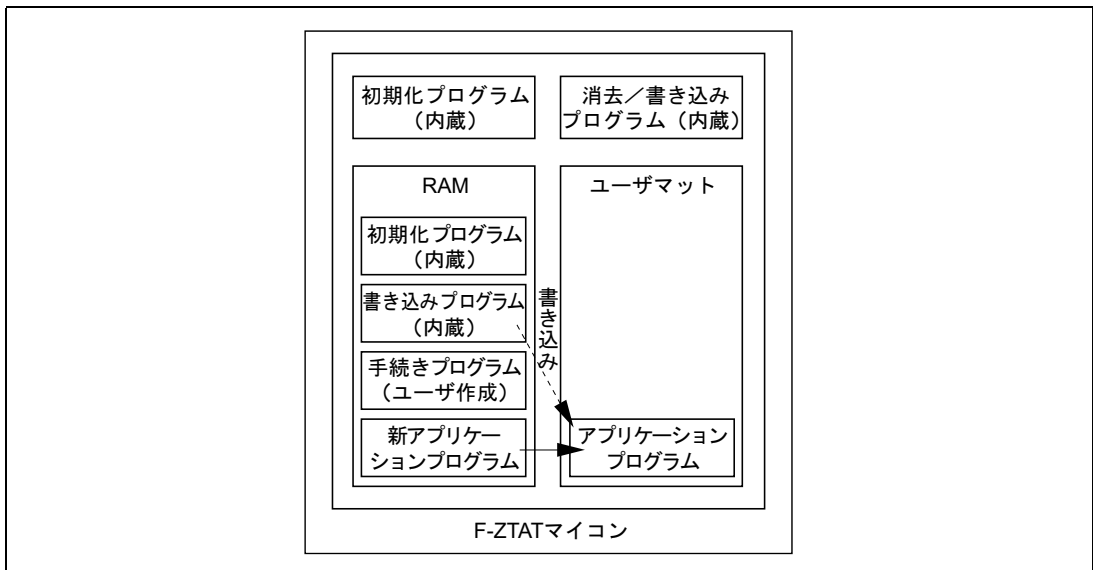


図 4.5 (C) ユーザマット書き込み

(4) ユーザマットの消去

RAM上にダウンロードした初期化プログラムをサブルーチン呼び出し、消去に必要な初期化処理を実行します。

↓ 初期化終了

RAM上にダウンロードした消去プログラムをサブルーチン呼び出しすることで、ユーザマットの消去を行います。このとき、ユーザマットの消去はブロック単位で行われるので、消去プログラムの呼び出しは1ブロックごとに行い、すべての消去対象ブロックが消去されるまで続けます。

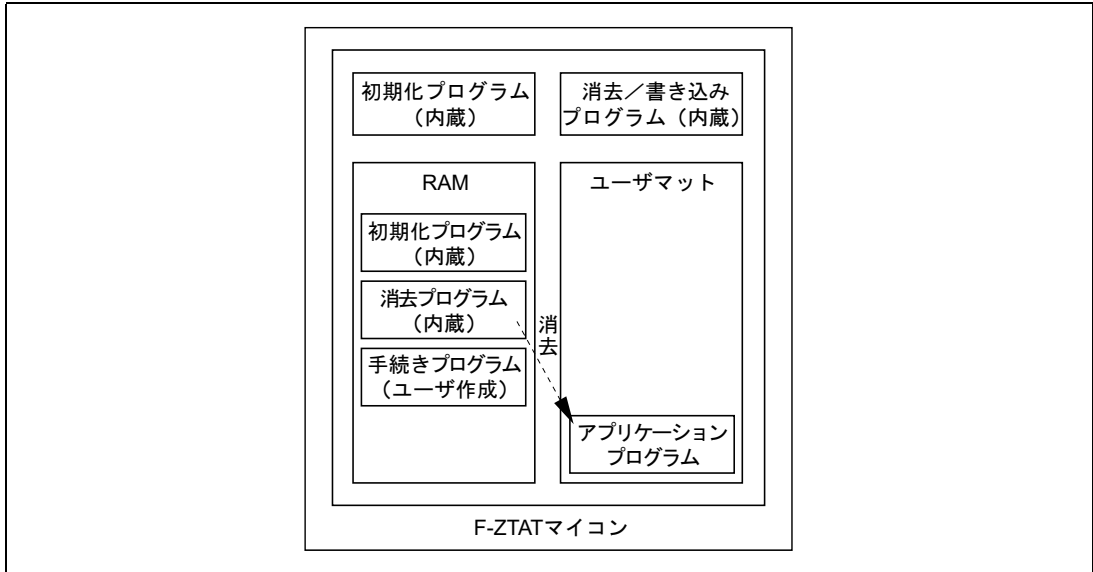


図 4.5(D) ユーザマット消去

4.4 ユーザブートモード概要

ユーザブートモードでは、内蔵SCII以外の任意のインターフェースでブートモードを実現できます。

ユーザブートモードの起動方法は、表4.1の端子設定を参照してください。ユーザブートモードでの消去/書き込みでは、手続きプログラムはユーザブートマット上に作成しておきます。また、ユーザブートモード起動時はユーザブートマットが選択されているため、ユーザマットの消去/書き込みを行う際には、ユーザマットへの切り替えを行う必要があります。

ユーザプログラムモードの処理の流れを図4.6に示します。

(1) 手続きプログラムのRAM展開

ユーザブートマット上の手続きプログラムをRAM上に展開します。以降の制御は、すべてRAM上に展開した手続きプログラムにて行います。

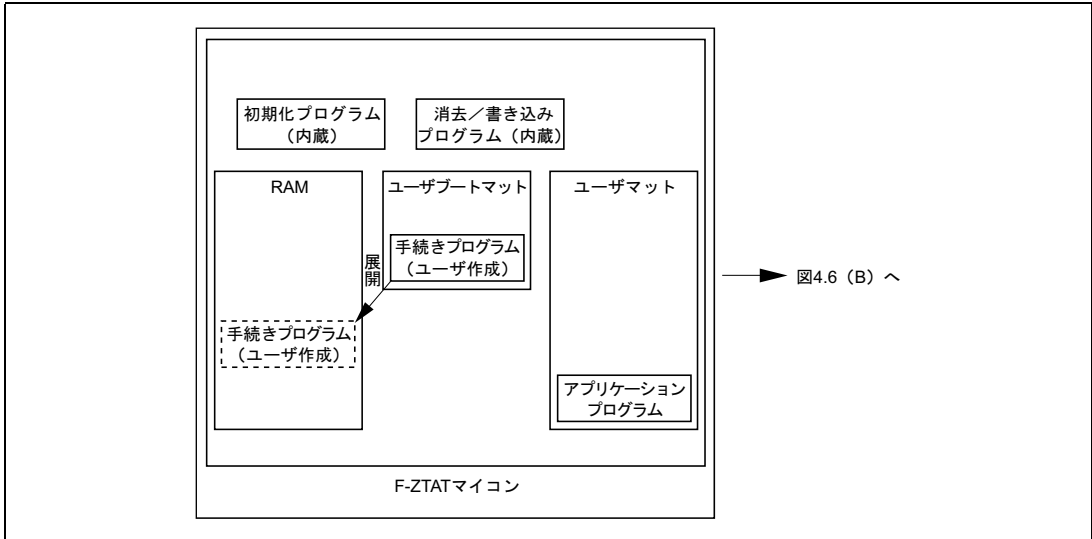


図 4.6 (A) 手続きプログラムの RAM 展開

(2) 内蔵プログラムのダウンロード

消去時には消去プログラムを、書き込み時には書き込みプログラムをRAM上にダウンロードします。

同時に、初期化プログラムがダウンロードされます。

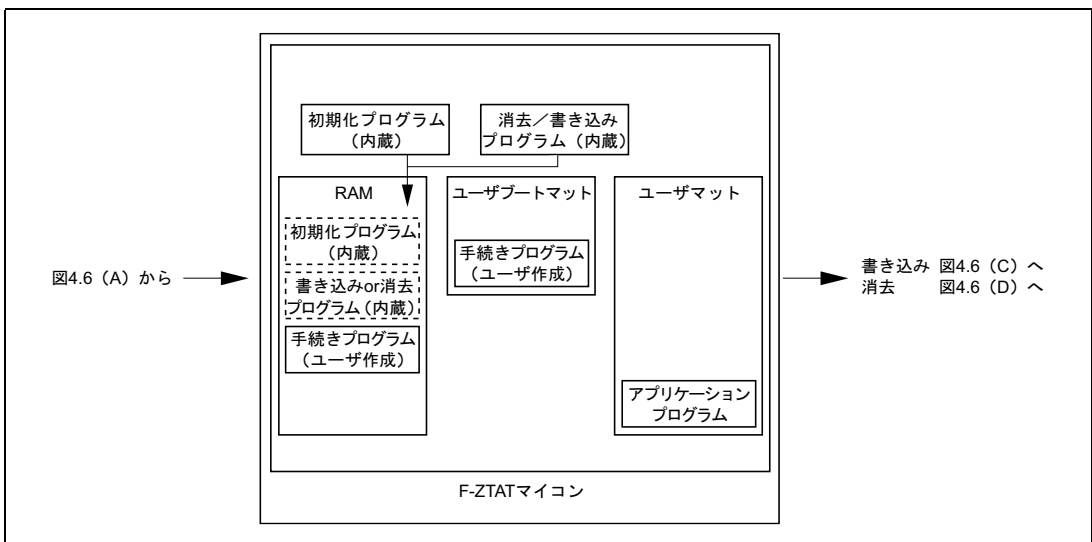


図 4.6 (B) 内蔵プログラムのダウンロード

(3) ユーザマットの書き込み

RAM上にダウンロードした初期化プログラムをサブルーチン呼び出し、書き込みに必要な初期化処理を実行します。

↓ 初期化終了

RAM上にダウンロードした書き込みプログラムをサブルーチン呼び出しすることで、ユーザマットの書き込みを行います。このとき、ユーザマットへの書き込みは128バイト単位で行われるので、書き込みプログラムの呼び出しは128バイトごとに行い、すべての書き込みデータ(アプリケーションプログラム)を書き込み完了するまで続けます。

【注意】 ユーザマット書き込み時には、ユーザブートマットからユーザマットへの切り替えが必要です。

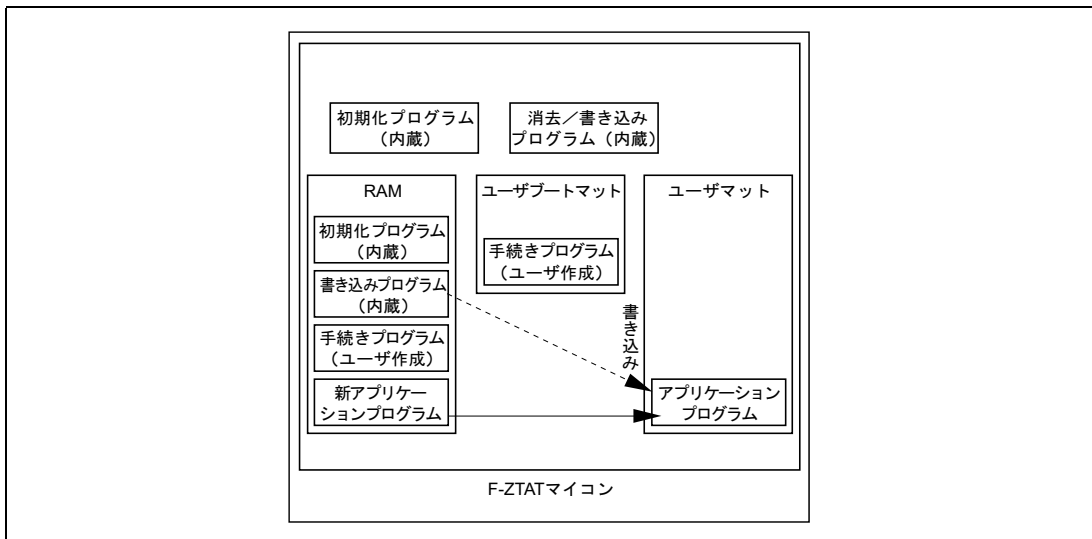


図 4.6 (C) ユーザマット書き込み

(4) ユーザマットの消去

RAM上にダウンロードした初期化プログラムをサブルーチン呼び出し、消去に必要な初期化処理を実行します。

↓ 初期化終了

RAM上にダウンロードした消去プログラムをサブルーチン呼び出しすることで、ユーザマットの消去を行います。このとき、ユーザマットの消去はブロック単位で行われるので、消去プログラムの呼び出しは1ブロックごとに行い、すべての消去対象ブロックが消去されるまで続けます。

【注意】 ユーザマット消去時には、ユーザブートマットからユーザマットへの切り替えが必要です。

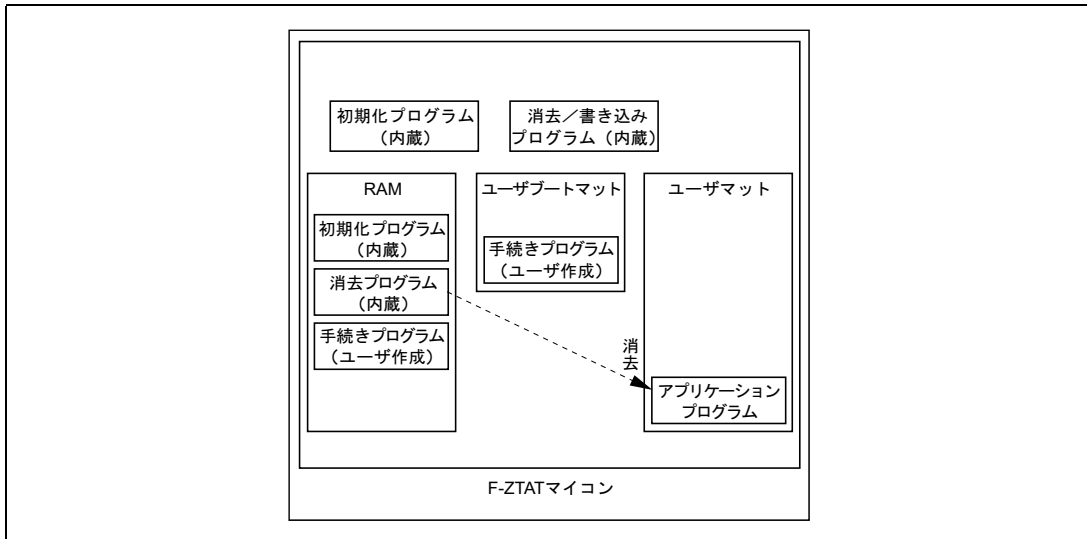


図 4.6 (D) ユーザマット消去

5. フラッシュ書き込み手順～SH7058F の場合

SH7058Fを例にして、0.18 μ m プロセスF-ZTATのフラッシュ書き込み手順を3種類のオンボードプログラミングモードごとに説明します。

5.1 ブートモード書き込み手順

ブートモードでリセットスタート後、ブートプログラムは次の3ステータスを持ちます。各ステータスの概要については、4.2のブートモード概要を参照してください。

各ステータスにおいて、ホストとブートプログラムのインターフェイスについて述べます。

5.1.1 ビットレート問い合わせステータス

ビットレート問い合わせステータスでは、ホストからSCI調歩同期式でH'00を送信し続けます。ブートプログラムでは、送信データのLOW期間を測定します。LOW期間の測定から、ホストの送信するビットレートを計算し、合わせ込み終了ステータス(H'00)をホストへ送信します。

ホストは、H'00を正常に受信したことを確認し、ブートプログラムに対し、H'55を送信します。

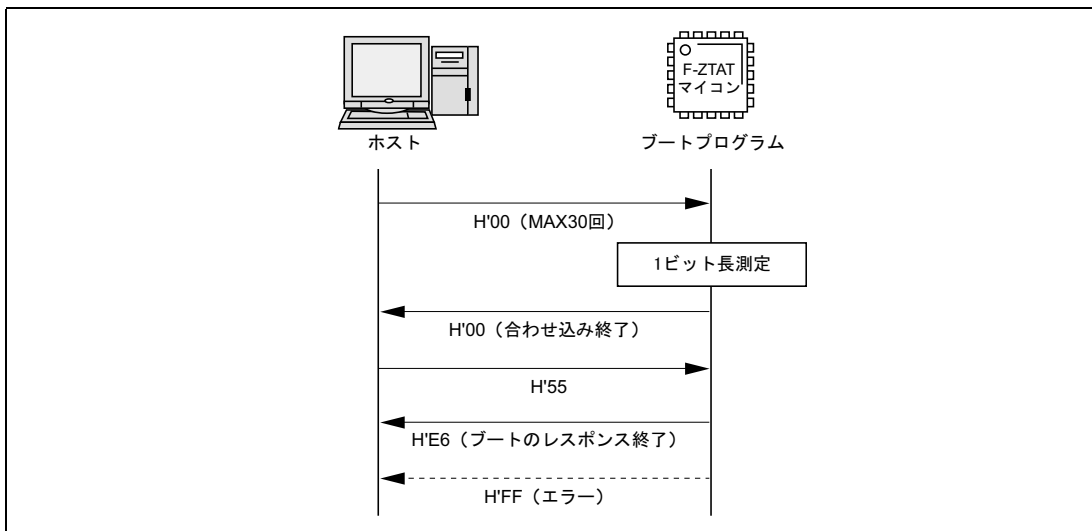


図 5.1 ビットレート問い合わせシーケンス

5.1.2 問い合わせ選択ステータス

問い合わせ選択ステータスでは、ブートプログラムはホストからの問い合わせコマンドに対してフラッシュROMの情報を応答し、選択コマンドに対しては、デバイス、クロックモード、ビットレートを選択します。

ホストは送信した問い合わせコマンドに対するレスポンスのチェックを行います。入力周波数、通倍比、動作周波数、ビットレートに対するチェック方法は、ハードウェアマニュアルを参照してください。

消去／書き込みステータスに遷移するには、必ず「デバイス選択(H'20)」、「クロックモード選択(H'11)」、「新ビットレート選択(H'3F)」の順に発行した後、ホストから「書き込み消去ステータス遷移 (H'40)」を発行してください。

ブートプログラムは、自動的にユーザマット、ユーザブートマットのデータを消去し、消去が終わると、ACK (H'06) を応答し、消去／書き込みステータスに遷移します。

問い合わせ／選択コマンドについては表5.1を参照ください。各コマンドの詳細は、ハードウェアマニュアルを参照してください。

表 5.1 問い合わせ／選択コマンド一覧

コマンド	コマンド名	機能
H'20	サポートデバイス問い合わせ	デバイスコードと F-ZTAT 品名の問い合わせ
H'10	デバイス選択	デバイスコードの選択
H'21	クロックモード問い合わせ	クロックモード数とそれぞれの値の問い合わせ
H'11	クロックモード選択	選択されているクロックモードの通知
H'22	通倍比問い合わせ	通倍比または分周比の種類数とそれぞれの個数とその値の問い合わせ
H'23	動作周波数問い合わせ	メインクロックとペリフェラルクロックの最大値／最小値の問い合わせ
H'24	ユーザブートマット情報問い合わせ	ユーザブートマットの個数とそれぞれの先頭／最終アドレスの問い合わせ
H'25	ユーザマット情報問い合わせ	ユーザマットの個数とそれぞれの先頭／最終アドレスの問い合わせ
H'26	消去ブロック情報問い合わせ	消去ブロック数とそれぞれの先頭／最終アドレスの問い合わせ
H'27	書き込みサイズ問い合わせ	書き込み時のデータ長の問い合わせ
H'3F	新ビットレート問い合わせ	新ビットレートの選択
H'40	書き込み消去ステータス遷移	ユーザブートマット、ユーザマットを消去し、書き込み消去ステータスに遷移
H'4F	ブートプログラムステータス問い合わせ	ブート処理状態の問い合わせ

5.1.3 消去／書き込みステータス

- 書き込み処理

書き込みを行うには、最初にホストから書き込み選択コマンドを発行して、書き込み方式と書き込みマットを選択します。次に「128バイト書き込みコマンド (H'40)」を発行し、データを書き込みます。

- 消去処理

消去を行うには、最初にホスト側から消去コマンドを発行します。次に「ブロック消去コマンド (H'58)」を発行して、指定したブロックを消去します。

消去／書き込みコマンドについては表5.2を参照ください。各コマンドの詳細は、ハードウェアマニュアルを参照してください。

表 5.2 書き込み消去コマンド一覧

コマンド	コマンド名	機能
H'42	ユーザブートマット書き込み選択	ユーザブートマット書き込みプログラムの転送
H'43	ユーザマット書き込み選択	ユーザマット書き込みプログラムの転送
H'50	128 バイト書き込み	128 バイト書き込み
H'48	消去選択	消去プログラムの転送
H'58	ブロック消去	ブロックデータの消去
H'52	メモリリード	メモリの読み出し
H'4A	ユーザブートマットのサムチェック	ユーザブートマットのサムチェック
H'4B	ユーザマットのサムチェック	ユーザマットのサムチェック
H'4C	ユーザブートマットのブランクチェック	ユーザブートマットのブランクチェック
H'4D	ユーザマットのブランクチェック	ユーザマットのブランクチェック
H'4F	ブートプログラムステータス問い合わせ	ブート処理状態の問い合わせ

5.2 ユーザプログラムモード書き込み手順

ユーザプログラムモードでは、ユーザでユーザマットの消去／書き込みの制御を行うプログラムを作成します。消去／書き込みは、RAM上で動作させます。

ユーザが作成する必要があるのは次のプログラムです。

1. RAM上で動作させる消去／書き込み手続きプログラム
2. 1.のプログラムをRAM上に展開するプログラム

128バイト単位の書き込みプログラム、1ブロック単位の消去プログラムは、マイコンに内蔵されています。ユーザはそれらの内蔵プログラムをRAM上へのダウンロードする処理、そして、書き込みプログラム又は、消去プログラムの呼び出しを行う手続きプログラムを作成する必要があります。初期化プログラムは、消去／書き込みプログラムのダウンロード時に同時にダウンロードされます。

書き込みを行う前には、書き込むエリアが消去されている必要があります。消去されていない場合には、必ず消去処理実行後、書き込み処理を行ってください。

図5.2にフラッシュメモリ(ユーザマット)の消去／書き込みの概略図を示します。

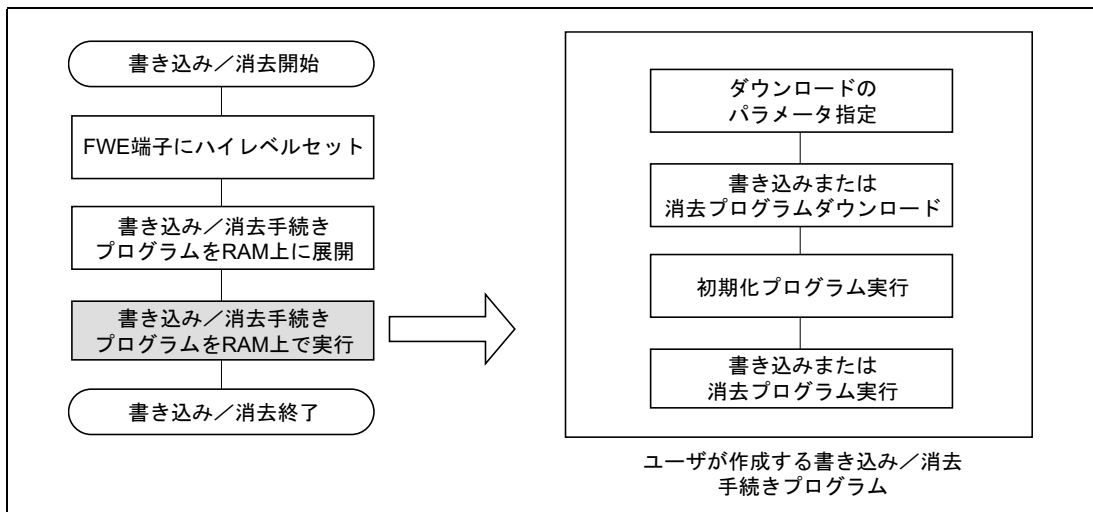


図 5.2 ユーザプログラムモード書き込み／消去処理概略

消去／書き込み手続きプログラムで設定を行うレジスタについて説明します。図5.2内の右枠に示した4処理 [(1) ダウンロードのパラメータ指定、(2) ダウンロードの実行、(3) 初期化の実行、(4) 消去／書き込みの実行] 別に記述します。

(1) ダウンロードのパラメータ指定

消去／書き込みを行うには、まず、内蔵プログラム等をRAMにダウンロードします。ダウンロード後のRAMの状態は図5.3のような構成になります。ダウンロード実施前に設定が必要なレジスタは次のとおりです。

- ダウンロード先の内蔵RAMアドレス指定
 - フラッシュトランスファディステーションレジスタ (FTDAR)
内蔵プログラムやその関連情報をダウンロードするときに、ダウンロード先の内蔵RAMのアドレスを指定します。このレジスタの設定でダウンロード先の先頭アドレスを幾通りかの決められたアドレスの中

から選択することができます。

- ダウンロードするプログラム(消去プログラム、書き込みプログラム)の選択
 - フラッシュプログラムコードセレクトレジスタ (FPCS)
内蔵書き込みプログラムのダウンロードをする場合に使用します。書き込み時には、本レジスタのPPVS(ビット0)を“1”にセットします。
 - フラッシュイレースコードセレクトレジスタ (FECS)
内蔵消去プログラムのダウンロードをする場合に使用します。消去時には、本レジスタのEPVB(ビット0)を“1”にセットします。

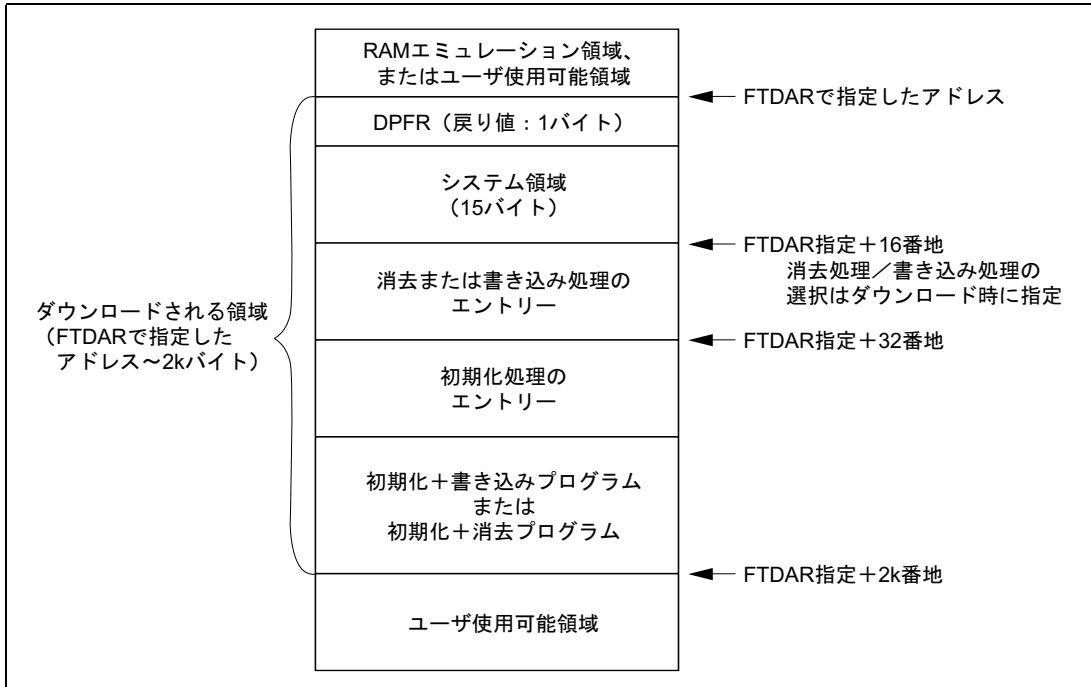


図 5.3 内蔵プログラムダウンロード実行後の RAM マップ

(2) ダウンロードの実行

ダウンロード実行時には、次のレジスタを設定します。

- フラッシュキーコードレジスタ(FKEY)
ダウンロード時には、本レジスタにH'A5をセットして下さい。それ以外の値がセットされていると、ソフトウェアプロテクト状態となり、FCCSのSCOに書き込むことができません。ダウンロード後はH'00にセットし、ソフトウェアプロテクト状態として下さい。
- フラッシュコードコントロール・ステータスレジスタ(FCCS)
本レジスタのSCO(ビット0)に“1”を書き込むことで、指定した内蔵RAM領域に指定されたプログラムをダウンロードします。

(3) 初期化の実行

消去／書き込みの実行前には初期化処理を行います。これはダウンロードした初期化プログラムを実行(サブルーチン呼び出し)することによって行います。初期化プログラムのエントリアドレスは、FTDARで設定したアドレス+32番地に格納されています。初期化実行時の参考プログラムを図5.4に示します。

また、初期化実行の際に設定するパラメータは次のとおりです。

- フラッシュプログラム/イレース周波数パラメータ(FPEFEQ:R4)
 CPUの動作周波数を設定します。
 (設定例) 40MHz動作の場合は、40.00(少数第3位は四捨五入) \times 100=4000をR4に設定
- フラッシュユーザブランチアドレス設定パラメータ(FUBRA:R5)
 消去/書き込み実行時のある決まった処理単位ごとに、設定したユーザプログラムを実行することができます。
- フラッシュパス/フェイルリザルトパラメータ(FPFR:R0)
 初期化結果がFPFR(R0)に格納されます。R0がH'00以外の場合エラーとなります。

繰り返し消去/書き込みを実行する場合、初期化処理は消去/書き込み前にそれぞれ一度実行するだけでかまいません。

```

<<初期化処理実行>> ...ENTRYはFTDARレジスタで指定したアドレス
MOV.L ENTRY+32, R0 ; 初期化プログラム格納アドレス指定
JSR @R0 ; 初期化プログラム実行
    
```

図 5.4 初期化プログラム実行例

(4) 消去/書き込みの実行

消去/書き込みの実行時にはダウンロードした消去プログラム/書き込みプログラムを実行(サブルーチン呼び出し)することによって行います。消去/書き込みプログラムのエントリアドレスは、FTDARで設定したアドレス+16番地に格納されています。消去/書き込み実行の参考プログラムを図5.5に示します。

ここで、ダウンロードした消去プログラムは1ブロックの消去、書き込みプログラムは128バイト単位の書き込みとなっています。したがって、複数のブロック消去、128バイトを超えた書き込みを行う場合には、繰り返し消去/書き込みプログラムを実行する必要があります。

また、消去/書き込み実行の際に設定するパラメータは次のとおりです。

- 消去/書き込み時、共通のパラメータ設定
 - フラッシュキーコードレジスタ(FKEY)
 消去/書き込み時には、本レジスタに**H'5A**をセットして下さい。それ以外の値がセットされていると、ソフトウェアプロテクト状態となり、消去/書き込みは実行されません。消去/書き込み後はH'00にセットし、ソフトウェアプロテクト状態として下さい。
 - フラッシュマットセレクトレジスタ(FMATS)
 ユーザマット選択/ユーザブートマットの切り替えを行うときに設定します。
 ユーザマットはH'00、ユーザブートマットはH'AAを指定します。
- 書き込み時のパラメータ設定
 - フラッシュマルチバースアドレスエリアパラメータ(FMPAR : R5)
 ユーザマット書き込む先の先頭アドレスを指定します。
 - フラッシュマルチバースデータデスティネーションパラメータ(FMPDR : R4)
 ユーザマットに書き込むデータが格納されている領域の先頭アドレスを設定します。
 - フラッシュパス/フェイルリザルトパラメータ(FPFR : R0)
 書き込み結果がFPFR(R0)に格納されます。R0がH'00以外の場合エラーとなります。
- 消去時のパラメータ設定
 - フラッシュイレースブロックセレクトパラメータ(FEBS : R4)
 消去するブロック番号を指定します。

- フラッシュバス/フェイルリザルトパラメータ(FPFR : R0)
消去結果がFPFR(R0)に格納されます。R0がH'00以外の場合エラーとなります。

```

<<消去/書き込み処理実行>>...ENTRYはFTDARレジスタで指定したアドレス
MOV.L ENTRY+16,R0          ;消去/書き込みプログラム格納アドレス指定
JSR    @R0                 ;消去/書き込みプログラム実行
    
```

図 5.5 消去/書き込みプログラム実行例

ユーザプログラムモードにおける書き込み手順を次に示します。図5.6に書き込み手続きプログラムのフローを示します。

1. FWE端子をハイレベルに設定
2. 消去/書き込み手続きプログラムを内蔵RAMに転送し、RAM上で実行
3. FTDARによりダウンロードするRAMのアドレスを指定
4. ダウンロードする内蔵プログラムを選択
書き込みを行う場合、FPCSレジスタのPPVSビットを1に設定して、書き込みプログラムを選択します。
5. FKEYレジスタにH'A5の書き込み
6. FCCSレジスタのSCOビットに1を書き込みます。これによりダウンロードが実行されます。
SCOビット書き込み後、プロテクトのため、FKEYレジスタはH'00にクリアします。
7. ダウンロード結果のチェック
DFPRパラメータのSSビット、FKビット、SFビットが0であることを確認します。1が立っていた場合エラー処理を行います。
8. 初期化のためのパラメータセット
FPEFEQパラメータ(R4)にCPUクロック周波数、FUBRAパラメータ(R5)にユーザブランチ先頭アドレスを設定します。
9. 初期化実行
書き込みプログラムのダウンロード時、初期化プログラムもいっしょにダウンロードされています。初期化プログラムのエントリーポイントは、(FTDAR設定値+32)番地のアドレスに展開されています。当アドレスをサブルーチンコールし、実行させることで、初期化が行われます。
10. 初期化結果のチェック
FPFRパラメータのBRビット、FQビット、SFビットが0であることを確認します。1が立っていた場合エラー処理を行います。
11. すべての割り込み、CPU以外のバスマスタの動作を禁止
12. FKEYレジスタにH'5Aの書き込み
13. ユーザマットの書き込み先の先頭アドレスをFMPARパラメータ(R5)にユーザマットに書き込むデータが格納されている領域の先頭アドレスを、FMPDRパラメータ(R4)に設定。
14. 書き込み実行
(FTDAR設定値+16)番地のアドレスに書き込みプログラムのエントリーポイントがあります。当アドレスをサブルーチンコールし、実行させることで、128バイトの書き込みが行われます。
15. 書き込み結果のチェック
FPFRパラメータのMDビット、EEビット、FKビット、WDビット、WAビット、SFビットが0であることを確認します。1が立っていた場合、エラー処理を行います。
16. 書き込み対象のデータがすべて書き込まれていなければ (13)に戻り、繰り返す。
17. FKEYレジスタをH'00にクリア

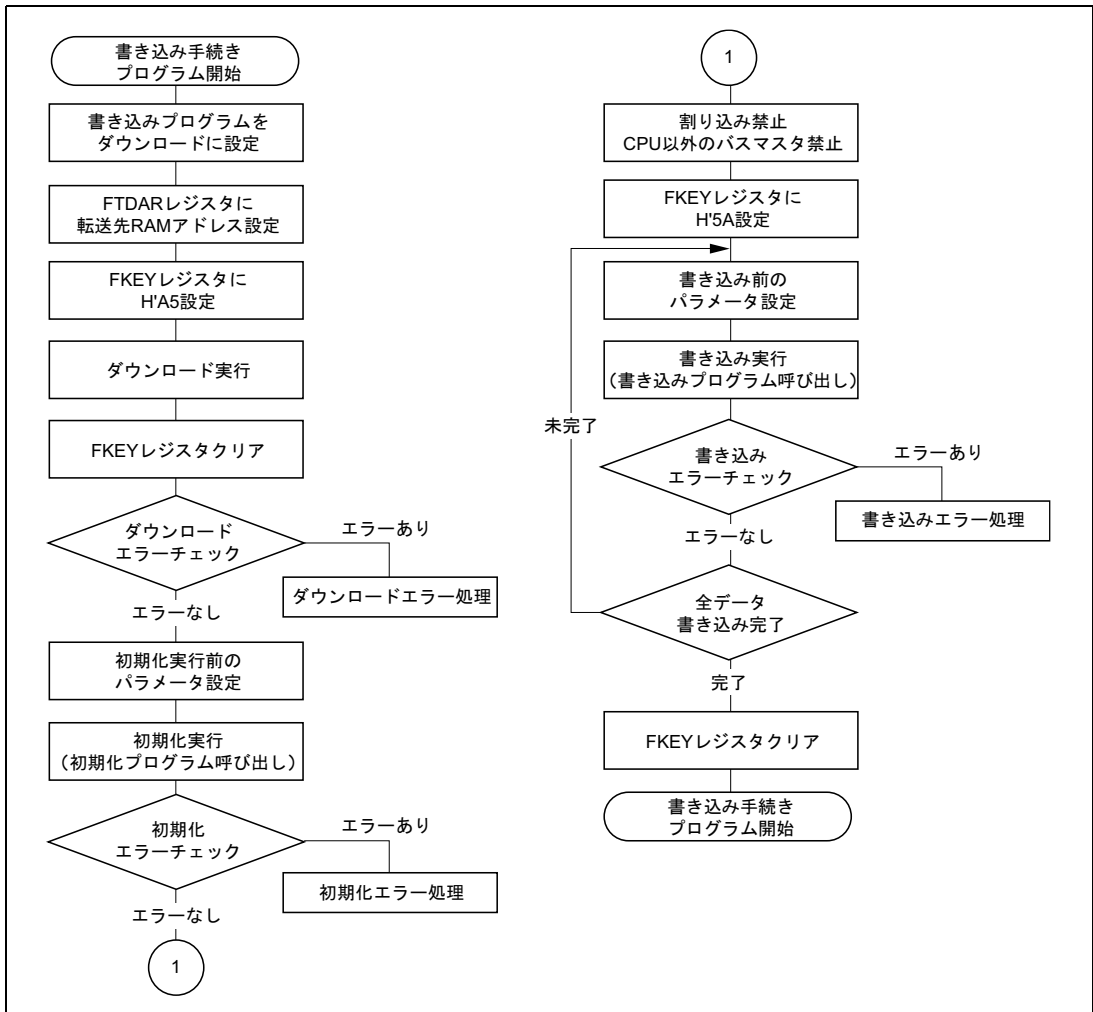


図 5.6 ユーザプログラムモード書き込み手続きプログラムフロー

ユーザプログラムモードにおける消去手順を次に示します。図5.7に消去手続きプログラムのフロー図を示します。

1. FWE端子をハイレベルに設定
2. 消去／書き込み手続きプログラムを内蔵RAMに転送し、RAM上で実行
3. FTDARによりダウンロードするRAMのアドレスを指定
4. ダウンロードする内蔵プログラムを選択
消去を行う場合、FECSレジスタのEPVBビットを1に設定して、消去プログラムを選択します。
5. FKEYレジスタにH'A5の書き込み
6. FCCSレジスタのSCOビットに1を書き込みます。これによりダウンロードが実行されます。
SCOビット書き込み後、プロテクトのため、FKEYレジスタはH'00にクリアします。
7. ダウンロード結果のチェック

- DPPFRパラメータのSSビット、FKビット、SFビットが0であることを確認します。1が立っていた場合エラー処理を行います。
8. 初期化のためのパラメータセット
FPEFEQパラメータ(R4)にCPUクロック周波数、FUBRAパラメータ(R5)にユーザブランチ先先頭アドレスを設定します。
 9. 初期化実行
消去プログラムのダウンロード時、初期化プログラムもいっしょにダウンロードされています。初期化プログラムのエントリーポイントは、(FTDAR設定値+32)番地のアドレスに展開されています。当アドレスをサブルーチンコールし、実行させることで、初期化が行われます。
 10. 初期化結果のチェック
FPPFRパラメータのBRビット、FQビット、SFビットが0であることを確認します。1が立っていた場合エラー処理を行います。
 11. すべての割り込み、CPU以外のバスマスタの動作を禁止
 12. FKEYレジスタにH'5Aの書き込み
 13. FEBSパラメータの設定
ユーザマットの消去ブロック番号をFEBS (R4)に設定します。
 14. 消去実行
(FTDAR設定値+16)番地のアドレスに消去プログラムのエントリーポイントがあります。当アドレスをサブルーチンコールし、実行させることで、指定した1ブロックの消去が行われます。
 15. 消去結果のチェック
FPPFRパラメータのMDビット、EEビット、FKビット、EBビット、SFビットが0であることを確認します。1が立っていた場合エラー処理を行います。
 16. 消去対象のブロックがすべて消去されていなければ (13) に戻り、繰り返す。
 17. FKEYレジスタをH'00にクリア

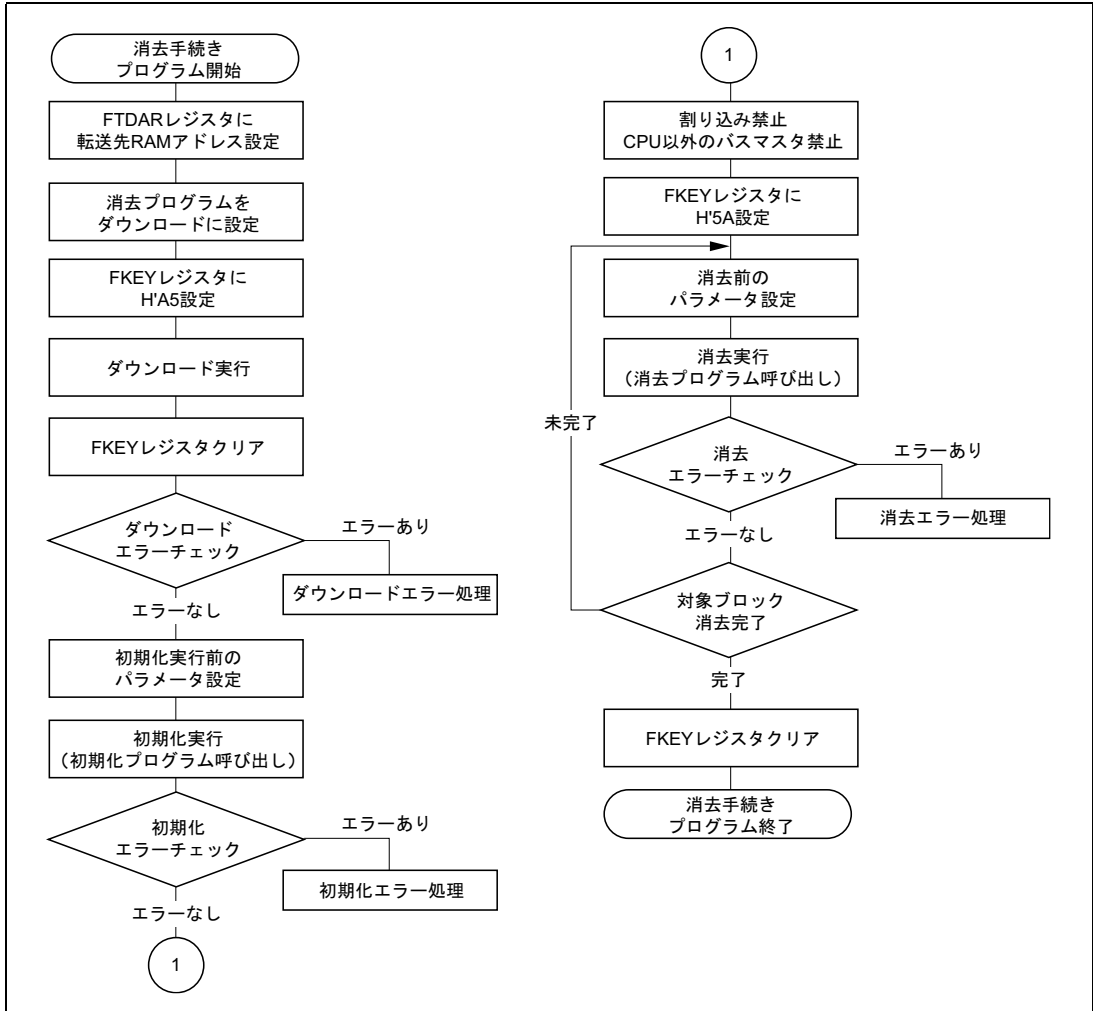


図 5.7 ユーザプログラムモード消去手続きプログラムフロー

5.3 ユーザブートモード書き込み手順

ユーザブートモードでの消去／書き込み時の処理の流れを図5.8に示します。

ユーザプログラムモードと同様、ユーザは手続きプログラムを作成する必要があります。

消去／書き込み手続きプログラムの処理において、ユーザプログラムモードとの違いは、マットの切り替えを行うか否かです。ユーザブートモードでは、リセットスタート後には、ユーザブートマットが選択されています。

ユーザマットに書き込みを行う場合には、ユーザブートマットからユーザマットへのマット切り替えが必要となり、書き込み終了後には、ユーザマットからユーザブートマットへ切り替える必要があります。

ユーザが手続きプログラムを作成する際には、このマット切り替えの処理を組み込まなくてはなりません。

ユーザブートマットからユーザマットに切り替えるタイミングは書き込み／消去開始前(FKEYレジスタにH'SA設定前)、ユーザマットからユーザブートマットに切り替えるタイミングは消去／書き込み終了後(FKEYレジスタクリア後)となります。

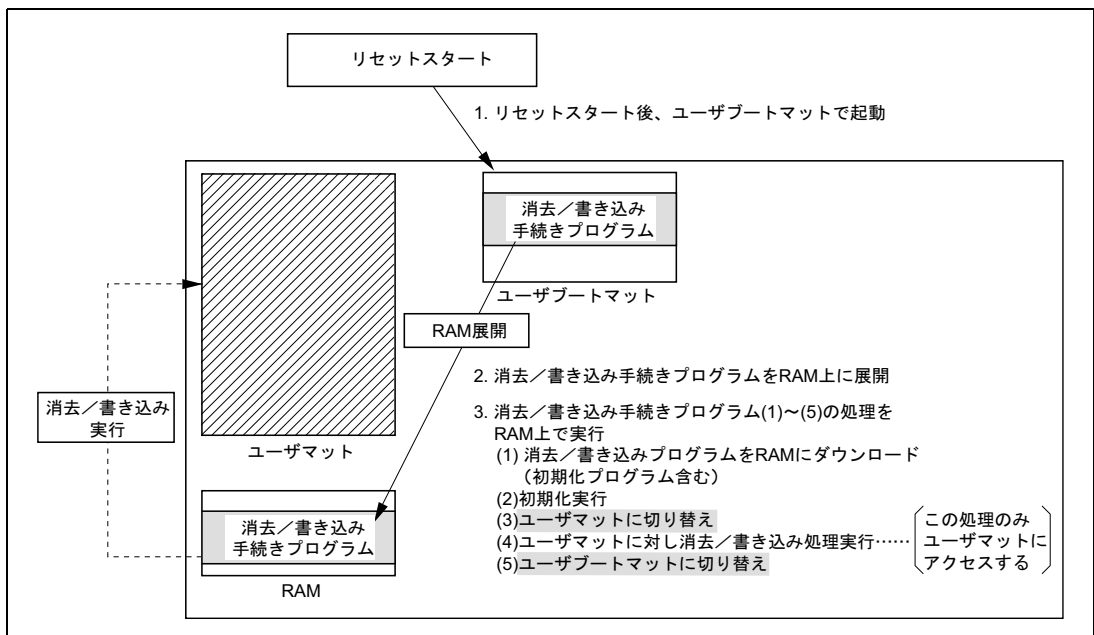


図 5.8 ユーザブートモード処理手順

ユーザブートモードにおける書き込み手続きプログラムのフローを図5.9に示します。ユーザプログラムモードとの違いは、網掛けになっている部分の処理が追加されます。

ユーザマットへの書き込み処理の際にマットの切り替えが必要となります。

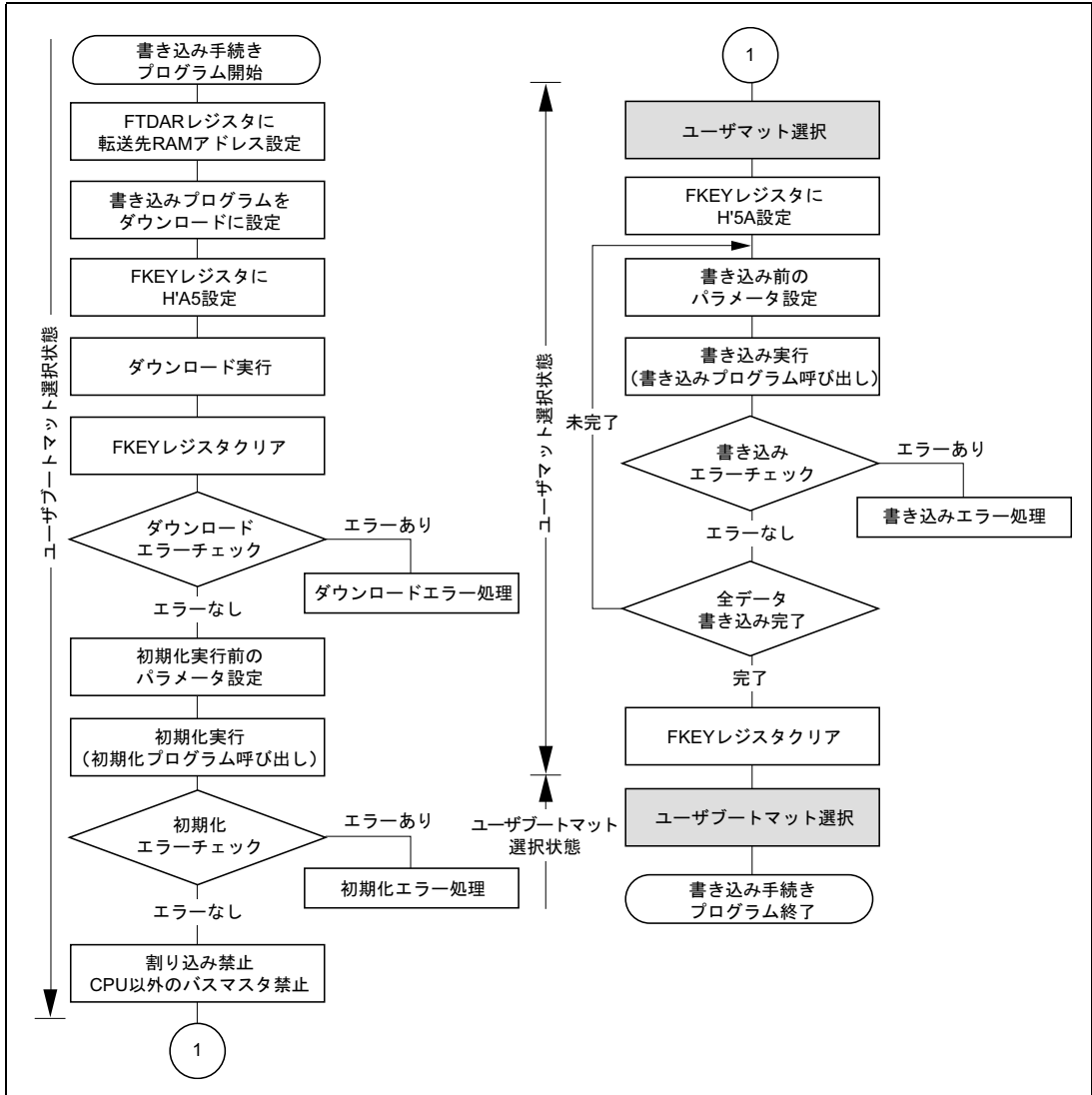


図 5.9 ユーザーブートモード書き込み手続きプログラムフロー例

ユーザーブートモードにおける消去手続きプログラムのフローを図5.10に示します。ユーザプログラムモードとの違いは、網掛けになっている部分の処理が追加されます。

ユーザーマットへの書き込み処理の際にマットの切り替えが必要となります。

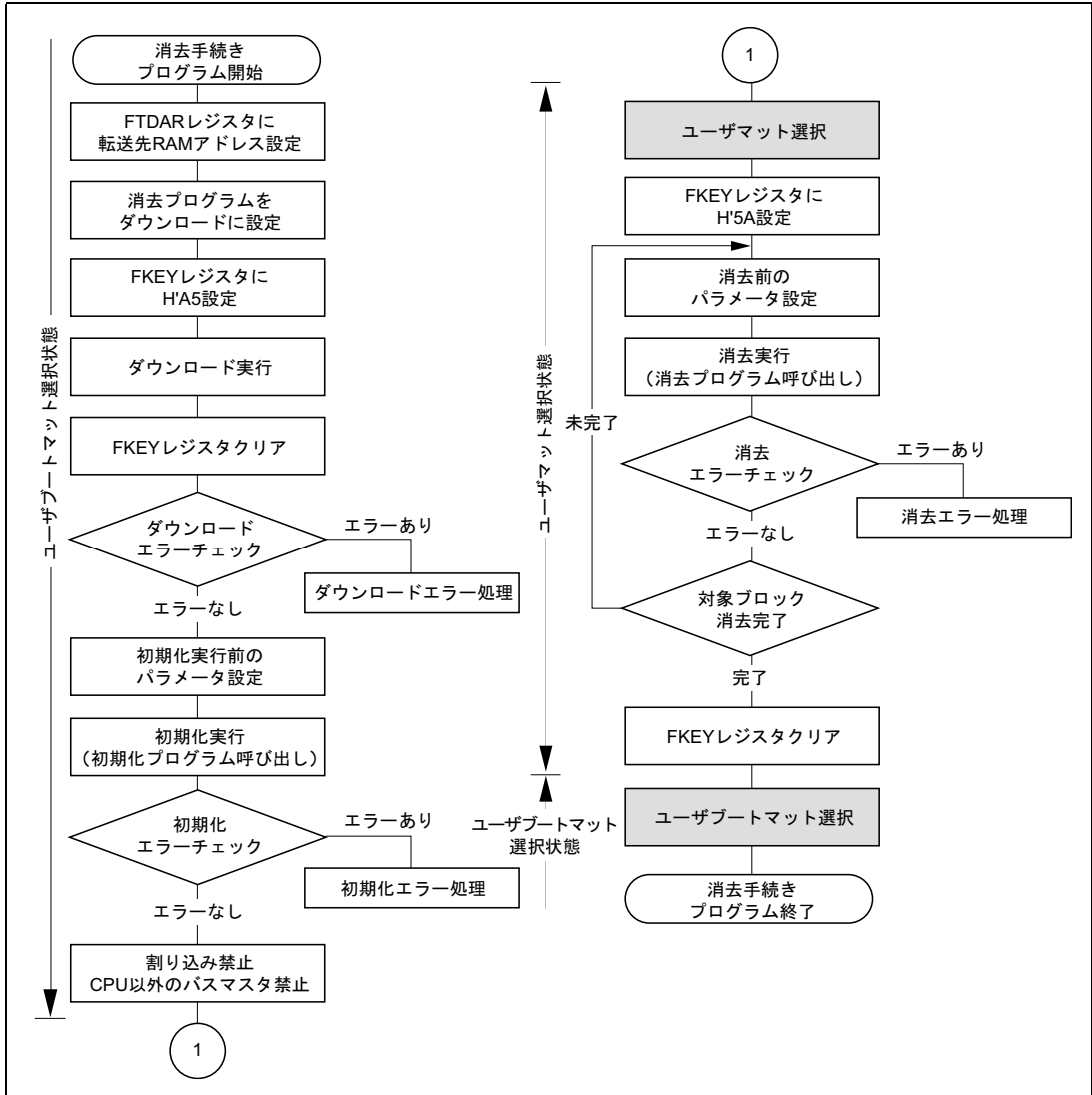


図 5.10 ユーザブートモード消去手続きプログラムフロー

6. ユーザ作成プログラム例

6.1 ユーザプログラムモードフラッシュ消去／書き込み手続きアプリケーションプログラム例

6.1.1 概要

本アプリケーションプログラム例では、ユーザプログラムモードでユーザアプリケーションが実行中に、外部からのアドバンストユーザデバッガ（以下AUDと称す）の制御でフラッシュ書き換えコマンドを受信し、フラッシュ消去／書き込み手続きアプリケーションプログラムを実行します。よって本手続きアプリケーションプログラムを使用するユーザアプリケーションは、周期的にコマンドエリアをリードし、コマンドがフラッシュ書き換えコマンド時にすべての割り込みを禁止し、本消去／書き込み手続きアプリケーションプログラムを内蔵RAMに展開後、実行する必要があります。

また、本消去／書き込み手続きアプリケーションプログラム例では、外部からのAUD制御とし、フラッシュ書き換えコマンドのほかに、消去ブロック情報と書き込みコマンド、書き込みデータ転送を行うものとします。

本消去／書き込み手続きアプリケーション例の処理の流れを図6.1に示します。

(1) 手続きプログラムのRAM展開

外部からのAUD制御により消去するブロック情報と書き換えコマンドを書き込む。

ユーザアプリケーションプログラムでコマンドエリアに書き換えコマンドが書き込まれていることを確認し、すべての割り込みを禁止後、ユーザマット上の手続きプログラムをRAM上に展開します。以降の制御は、すべてRAM上に展開した手続きプログラムにて行います。

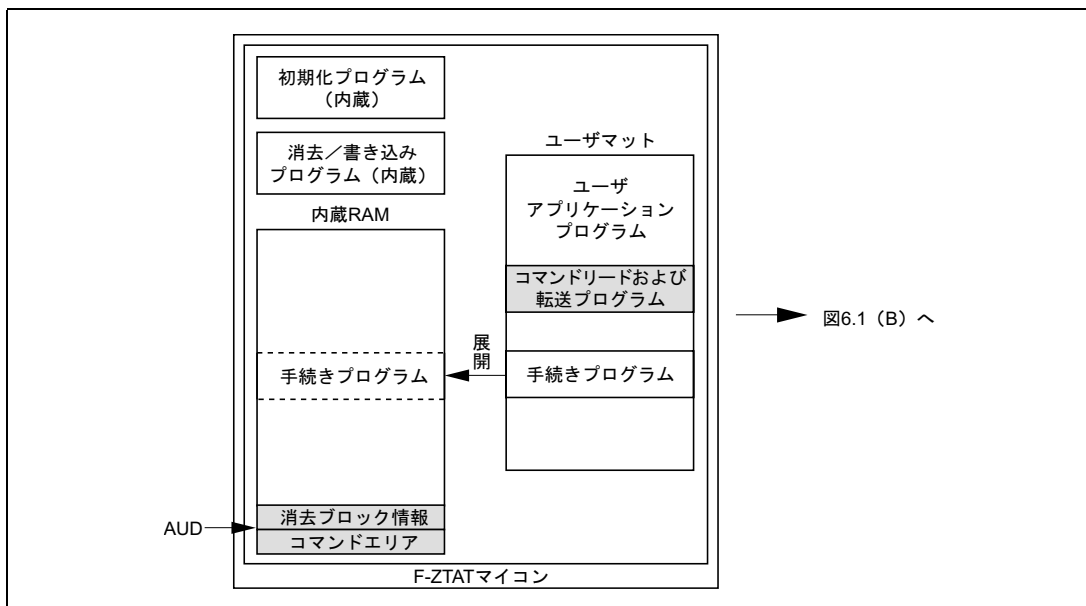


図 6.1 (A) 手続きプログラムのRAM展開

(2) 内蔵プログラムのダウンロード

消去時には消去プログラムを、書き込み時には書き込みプログラムをRAM上にダウンロードします。

同時に、初期化プログラムをダウンロードします。

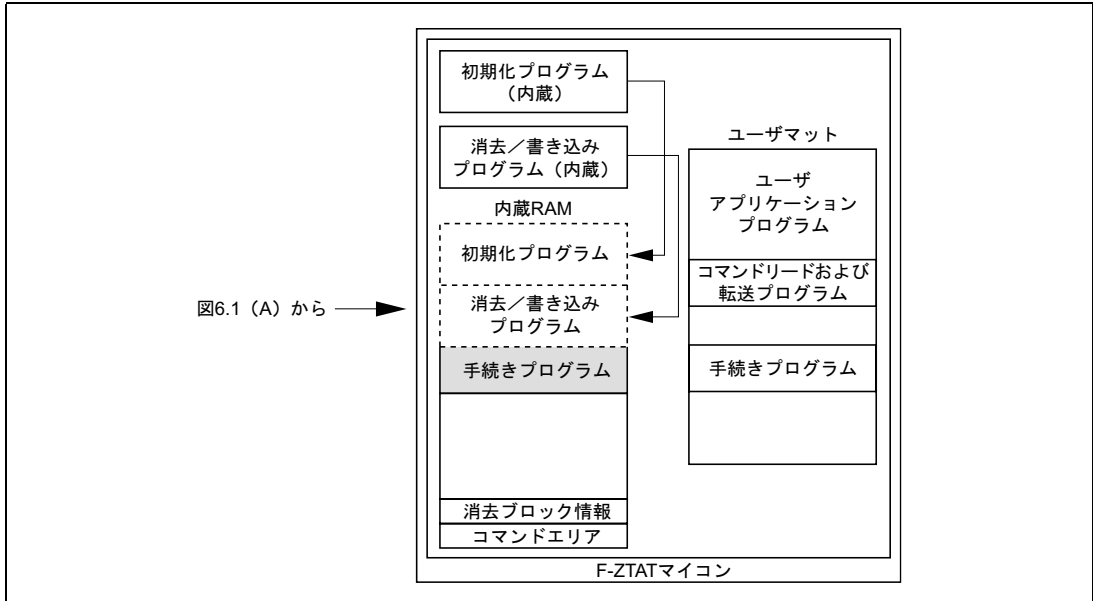


図6.1 (A) から →

図 6.1 (B) 内蔵プログラムのダウンロード

(3) ユーザマットの消去

RAM上にダウンロードした初期化プログラムをサブルーチン呼び出し、消去に必要な初期化処理を実行します。初期化終了後、消去ブロック情報からRAM上にダウンロードした消去プログラムをサブルーチン呼び出しすることで、ユーザマットの消去を行います。

このとき、ユーザマットの消去はブロック単位で行われるので、消去プログラムの呼び出しは1ブロックごとに行い、消去ブロック情報のすべての消去対象ブロックが消去されるまで続けます。

【注意】 手続きプログラム格納エリアとコマンドリードおよび転送プログラムのブロックは消去しない様に注意が必要です。

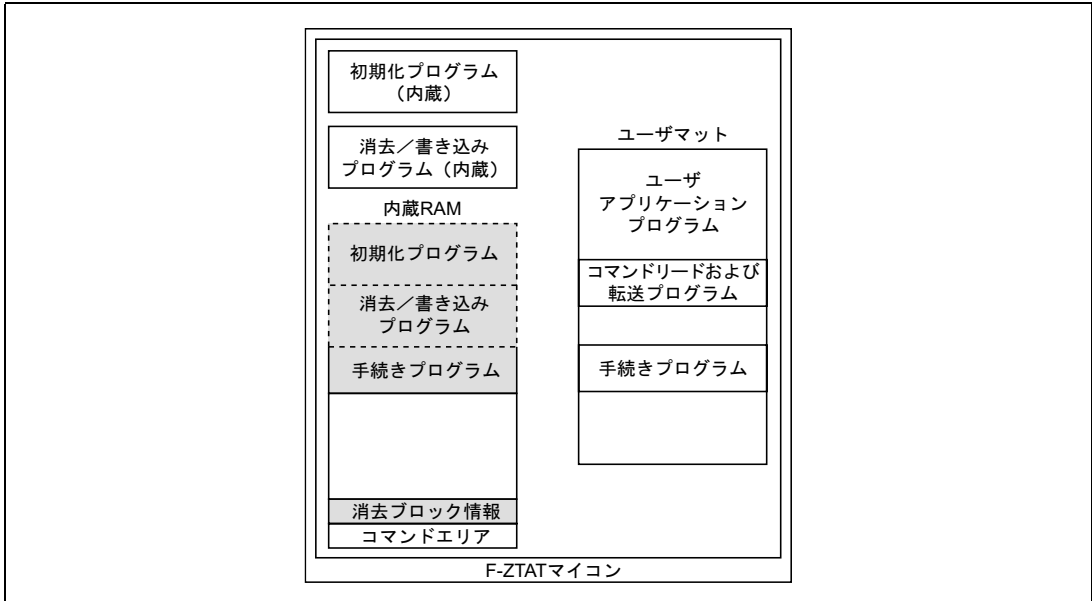


図 6.1 (C) ユーザマット消去

(4) ユーザマットの書き込み

RAM上にダウンロードした初期化プログラムをサブルーチン呼び出し、書き込みに必要な初期化処理を実行します。

初期化終了後、RAM上にダウンロードした書き込みプログラムをサブルーチン呼び出しすることで、消去されたブロックの先頭アドレスから順に外部からのAUD制御でRAM上に格納した書き込みデータをユーザマットの書き込みを行います。

このとき、ユーザマットへの書き込みは128バイト単位で行われるので、128バイト単位で外部からのAUD制御でRAM上に格納し、書き込みプログラムの呼び出しを行い、消去されたブロックのすべての書き込みデータ(アプリケーションプログラム)を書き込み完了するまで続けます。

【注意】 本手続きプログラムでは消去したブロックの先頭アドレスから順に書き込みます。外部からのAUD制御でRAM上に格納するデータはアドレス順に準備されるものとします。

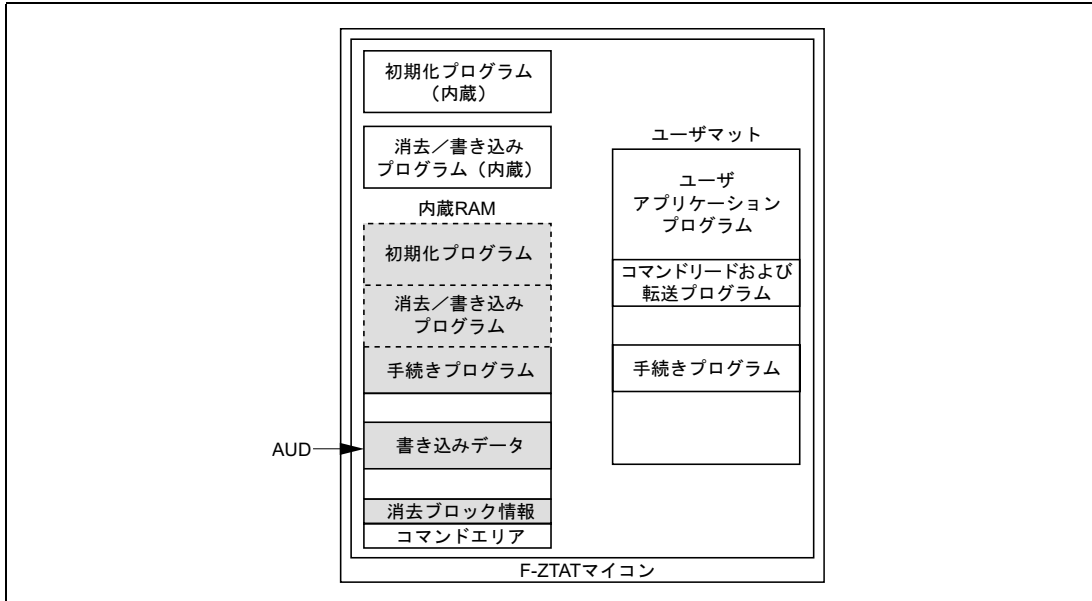


図 6.1 (D) ユーザマト書き込み

6.1.2 詳細仕様

- ユーザアプリケーション仕様
 - 2ms ほどのコンペアマッチ処理内でコマンドエリアを参照し、フラッシュ書き換えコマンドのときに手続きアプリケーションプログラムを内蔵RAMに展開後実行します。
- ユーザプログラムモードフラッシュ消去/書き込み手続きアプリケーションプログラム仕様
 - ユーザアプリケーションにより内蔵RAMに展開され実行します。
 - 実行手順は、以下のとおりです。
 - フラッシュ消去手続きと消去ブロック情報からブロック消去を行います。
 - 消去終了後、コマンドエリアに消去終了コマンドを書き込み、書き込みデータを要求します。
 - フラッシュ書き込み手続きと内蔵RAMに準備された書き込みデータを消去ブロックの先頭アドレスから順に書き込みます。
 - 128バイト書き込み終了ごとにコマンドエリアに書き込み終了コマンドを書き込み次の書き込みデータを要求します。

- コマンド仕様
 コマンドエリアをH'FFFFB000～H'FFFFB003とします。
 コマンドは外部からのAUD制御によって書き換える場合と消去／書き込み手続きプログラムによるCPU書き換えの場合があります。
 表6.1にコマンド仕様を示します。

表 6.1 コマンド仕様

コマンド (H'FFFFB000～H'FFFFB003)		
コマンド	コマンド名	機能
H'464C5752	FLWR	フラッシュメモリ書き換え (AUD 書き換え)
H'45444552	EDER	フラッシュメモリ消去プログラムダウンロードエラー (CPU 書き換え)
H'45494552	EIER	フラッシュメモリ消去初期設定エラー (CPU 書き換え)
H'45524552	ERER	フラッシュメモリブロック消去エラー (CPU 書き換え)
H'45454E44	EEND	フラッシュメモリ消去正常終了 (CPU 書き換え)
H'57444552	WDER	フラッシュメモリ書き込みプログラムダウンロードエラー (CPU 書き換え)
H'57494552	WIER	フラッシュメモリ書き込み初期設定エラー (CPU 書き換え)
H'57454E44	WEND	フラッシュメモリページ (128Byte) 書き込み正常終了 (CPU 書き換え)
H'57524552	WRER	フラッシュメモリページ (128Byte) 書き込みエラー (CPU 書き換え)
H'42554631	BUF1	BUFF1 エリアに書き込みデータ準備終了 (AUD 書き換え)
H'42554632	BUF2	BUFF2 エリアに書き込みデータ準備終了 (AUD 書き換え)

- 消去ブロック情報
 消去ブロック情報をH'FFFFB004～H'FFFFB007とします。
 消去ブロック情報は外部からのAUD制御によって、フラッシュ書き換えコマンド (FLWR) 書き換え直前に書き換えます。情報はビット0～15に設定され、ビット0はEB0ブロック、ビット15はEB15ブロックに対応します。

ユーザプログラムモードフラッシュ消去／書き込み手続きアプリケーションプログラム例の命令制御シーケンスを図6.2に示します。

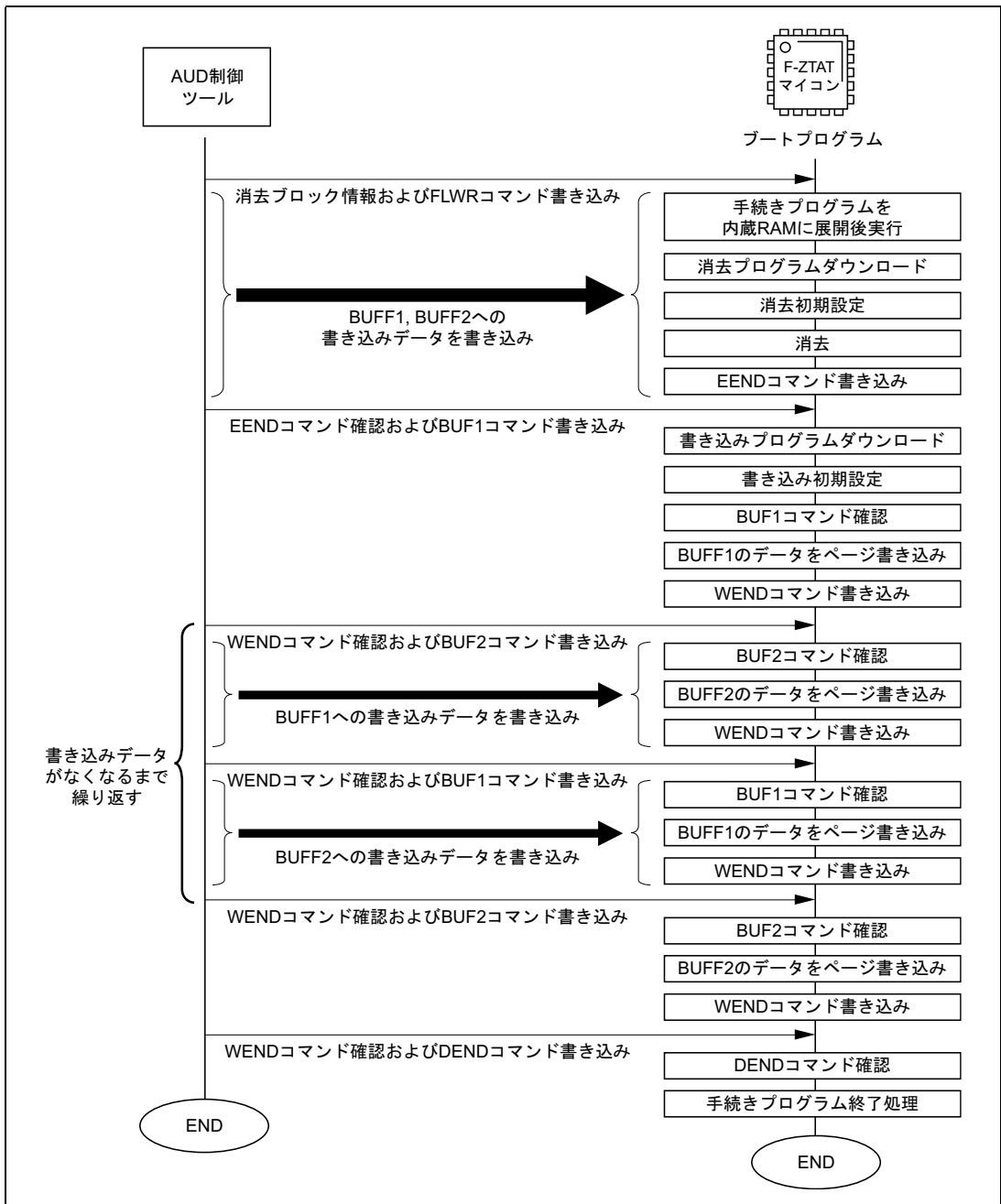


図 6.2 コマンド制御シーケンス

6.1.3 メモリマップ

本アプリケーション例のユーザマット上のユーザアプリケーションと消去／書き込み手続きアプリケーションの格納エリアおよび内蔵RAM上の消去／書き込み手続きアプリケーションと消去／書き込みプログラムの実行エリアメモリマップを表6.2に示します。

表 6.2 ユーザマットおよび内蔵 RAM メモリマップ

ユーザマット	
H'0000 0000~039F	ベクタテーブル
H'0000 0600~0FFF	コンペアマッチ処理 (コマンド解析)
H'0000 1000~11CF	手続きアプリケーションプログラム格納エリア
H'0000 2000~FFFF	フラッシュ書き換え対象エリア
内蔵 RAM	
H'FFFF 0000~07FF	消去プログラムダウンロード領域
0000	DPFR (戻り値: 1 バイト)
0001~000F	システム領域 (15 バイト)
0010~001F	消去処理エントリー
0020~002F	消去初期化エントリー
0030~07FF	消去初期化+消去プログラム
H'FFFF 0800~0FFF	書き込みプログラムダウンロード領域
0800	DPFR (戻り値: 1 バイト)
0801~080F	システム領域 (15 バイト)
0810~081F	書き込み処理エントリー
0820~082F	書き込み初期化エントリー
0830~0FFF	書き込み初期化+書き込みプログラム
H'FFFF 1000~107F	BUFF1 (書き込みデータ格納エリア 1)
H'FFFF 1080~10FF	BUFF2 (書き込みデータ格納エリア 2)
H'FFFF 1100~12CF	手続きアプリケーションプログラム実行エリア
H'FFFF B000~B003	コマンドエリア (4 バイト) :command
H'FFFF B004~B005	消去ブロック情報 (2 バイト) :ERASE_block

以上のコンペアマッチ処理 (コマンド解析) プログラムと消去／書き込み手続きアプリケーションプログラムリストを添付のリスト1、リスト2に示します。

6.2 ユーザブートモードフラッシュ消去／書き込み手続きアプリケーションプログラム例

6.2.1 概要

本アプリケーションプログラム例は、ユーザブートモードで起動後、ユーザブートプログラムによって、ユーザブートマットに格納されているフラッシュ消去／書き込み手続きアプリケーションプログラムを内蔵RAMに展開後フラッシュ全面消去および書き込みを行います。

また、本消去／書き込み手続きアプリケーションプログラム例では、外部からのAUD制御で書き込みデータ転送を行うものとします。

ここで、ユーザブートモードがユーザプログラムモードと違うことは、フラッシュ消去／書き込み手続きアプリケーションプログラムがユーザブートマットに格納されていることとフラッシュ消去／書き込み時にマット切り替えが必要なことです。

本消去／書き込み手続きアプリケーション例の処理の流れを図6.3に示します。

(1) 手続きプログラムの RAM 展開

ユーザブートモードで起動後、ユーザブートプログラムによって、ユーザブートマット上の手続きプログラムをRAM上に展開します。以降の制御は、すべてRAM上に展開した手続きプログラムにて行います。

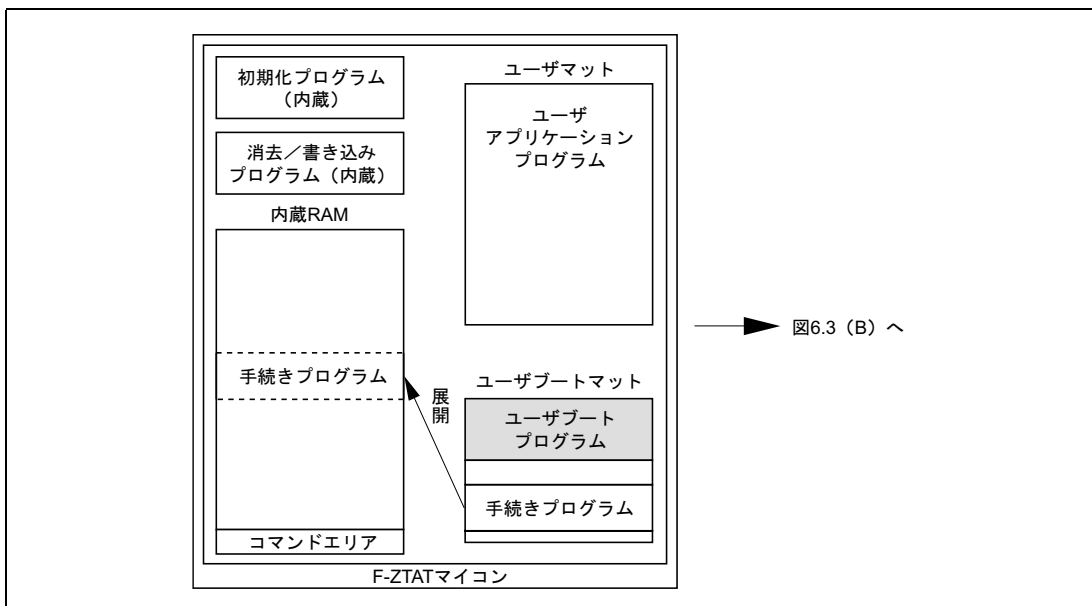


図 6.3 (A) 手続きプログラムの RAM 展開

(2) 内蔵プログラムのダウンロード

消去時には消去プログラムを、書き込み時には書き込みプログラムをRAM上にダウンロードします。

同時に、初期化プログラムをダウンロードします。

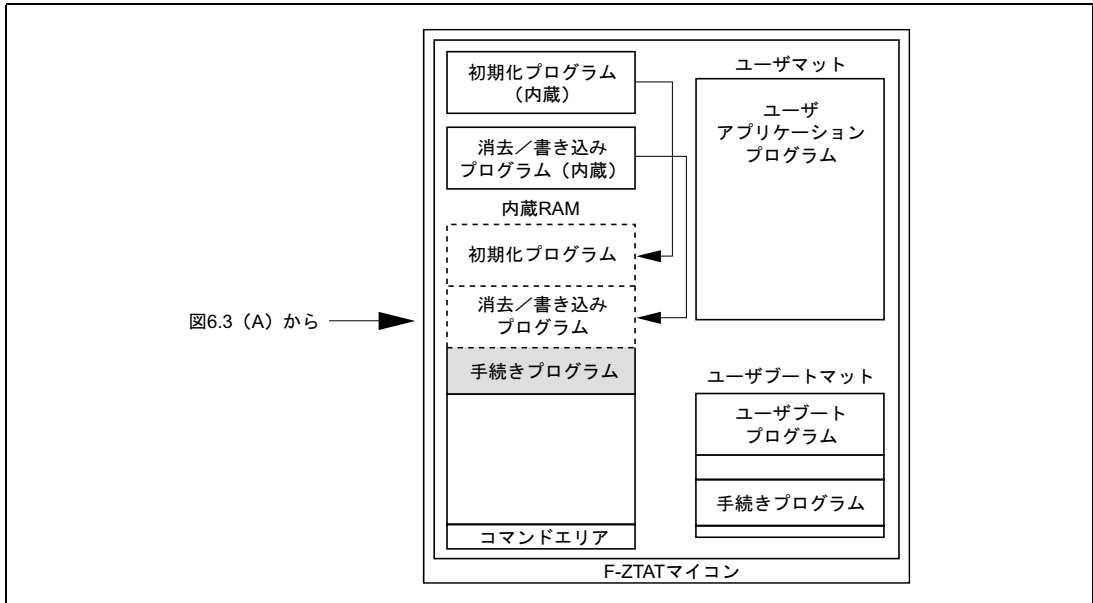


図6.3 (A) から

図 6.3 (B) 内蔵プログラムのダウンロード

(3) ユーザマットの消去

ユーザブートマット**選択状態**でRAM上にダウンロードした初期化プログラムをサブルーチン呼び出し、消去に必要な初期化処理を実行します。

初期化終了後、**ユーザマット選択にマット切り替え**後消去ブロックごとにRAM上にダウンロードした消去プログラムをサブルーチン呼び出しすることで、ユーザマットの消去を行います。

本手続きアプリケーションプログラム例では、ユーザマットすべてを消去します。

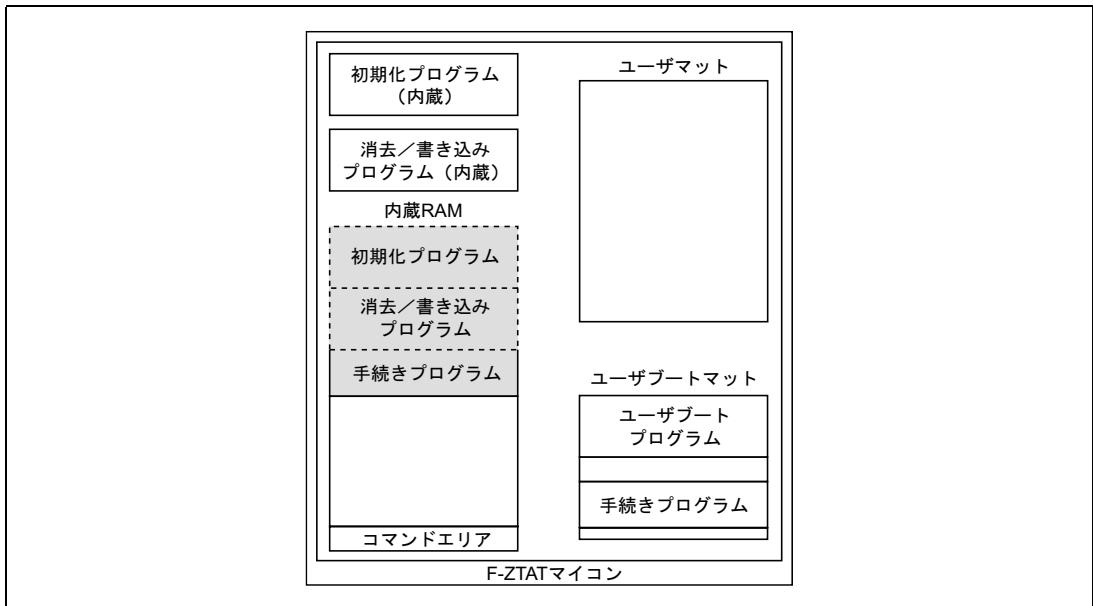


図 6.3 (C) ユーザマット消去

(4) ユーザマットの書き込み

ユーザブートマット**選択状態**でRAM上にダウンロードした初期化プログラムをサブルーチン呼び出し、書き込みに必要な初期化処理を実行します。

初期化終了後、**ユーザマット選択にマット切り替え**後RAM上にダウンロードした書き込みプログラムをサブルーチン呼び出しすることで、先頭アドレスから順に外部からのAUD制御でRAM上に格納した書き込みデータをユーザマットの書き込みを行います。

本手続きアプリケーションプログラム例では、ユーザマットの先頭アドレスから順に1 Mbyteすべて書き込みます。また、外部からのAUD制御でRAM上に格納するデータはアドレス順に準備されるものとします。

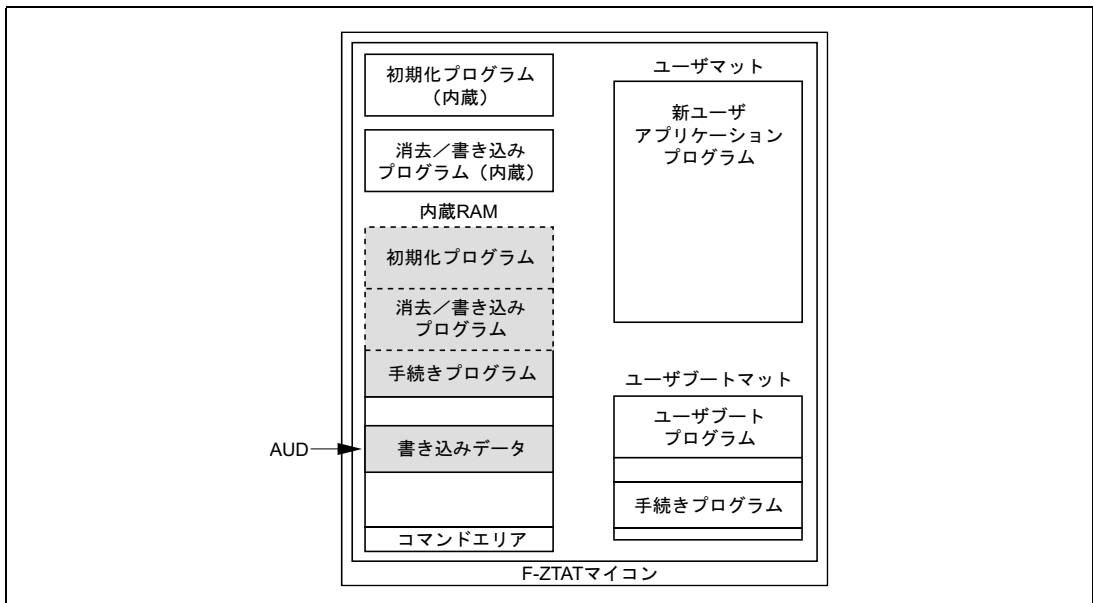


図 6.3 (D) ユーザマット書き込み

6.2.2 詳細仕様

- ユーザブートプログラム仕様
起動後、消去／書き込み手続きアプリケーションプログラムを内蔵RAMに展開後実行します。
- ユーザブートモードフラッシュ消去／書き込み手続きアプリケーションプログラム仕様
ユーザブートプログラムにより内蔵RAMに展開され実行します。
実行手順は、以下のとおりです。
 1. フラッシュ消去手続き後**ユーザマット選択にマット切り替え**し、からブロック消去を行います。
 2. 消去終了後、**ユーザブートマット選択にマット切り替え**し、コマンドエリアに消去終了コマンドを書き込み、書き込みデータを要求します。
 3. フラッシュ書き込み手続き後**ユーザマット選択にマット切り替え**し、内蔵RAMに準備された書き込みデータを消去ブロックの先頭アドレスから順に書き込みます。
 4. 128バイト書き込み終了ごとにコマンドエリアに書き込み終了コマンドを書き込み次の書き込みデータを要求します。
- コマンド仕様
コマンドエリアをH'FFFFB000～H'FFFFB003とします。
コマンドは外部からのAUD制御によって書き換える場合と消去／書き込み手続きプログラムによるCPU書き換え場合があります。
表6.3にコマンド仕様を示します。

表 6.3 コマンド仕様

コマンド (H'FFFFB000～H'FFFFB003)		
コマンド	コマンド名	機能
H'45444552	EDER	フラッシュメモリ消去プログラムダウンロードエラー (CPU 書き込み)
H'45494552	EIER	フラッシュメモリ消去初期設定エラー (CPU 書き込み)
H'45524552	ERER	フラッシュメモリブロック消去エラー (CPU 書き込み)
H'45454E44	EEND	フラッシュメモリ消去正常終了 (CPU 書き込み)
H'57444552	WDER	フラッシュメモリ書き込みプログラムダウンロードエラー (CPU 書き込み)
H'57494552	WIER	フラッシュメモリ書き込み初期設定エラー (CPU 書き込み)
H'57454E44	WEND	フラッシュメモリページ (128Byte) 書き込み正常終了 (CPU 書き込み)
H'57524552	WRER	フラッシュメモリページ (128Byte) 書き込みエラー (CPU 書き込み)
H'42554631	BUF1	BUF1 エリアに書き込みデータ準備終了(AUD 書き込み) (CPU 書き込み)
H'42554632	BUF2	BUF2 エリアに書き込みデータ準備終了(AUD 書き込み) (CPU 書き込み)

ユーザブートモードフラッシュ消去／書き込み手続きアプリケーションプログラム例のコマンド制御シーケンスを図6.4に示します。

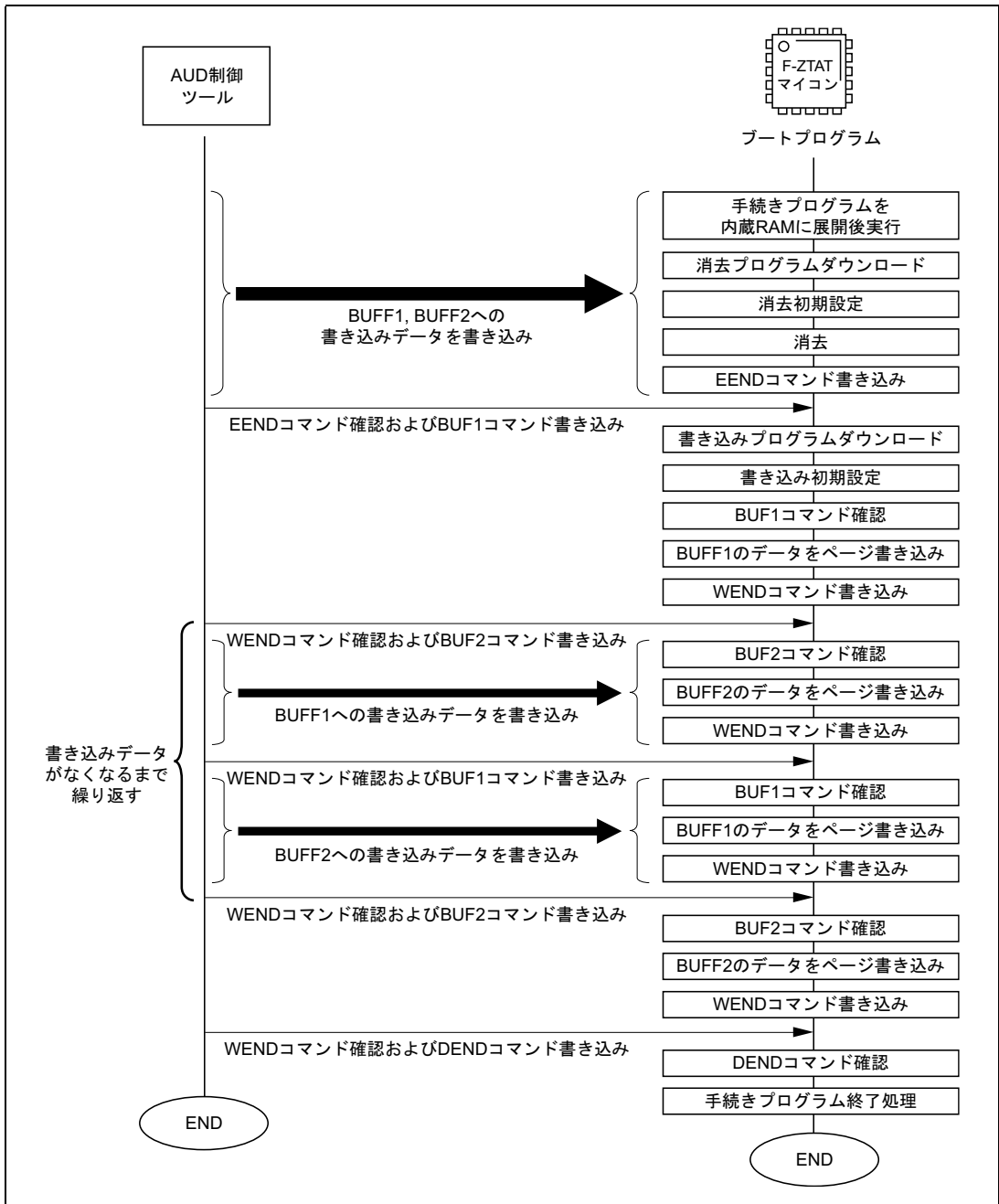


図 6.4 コマンド制御シーケンス

本アプリケーション例のユーザブートマット上のユーザアプリケーションと消去／書き込み手続きアプリケーションの格納エリアおよび内蔵RAM上の消去／書き込み手続きアプリケーションと消去／書き込みプログラムの実行エリアメモリマップを表6.4に示します。

表 6.4 ユーザブートマットおよび内蔵 RAM メモリ

ユーザブートマット	
H'0000 0000~039F	ベクタテーブル
H'0000 0600~0FFF	手続きアプリケーションプログラム転送プログラム
H'0000 1000~11F4	手続きアプリケーションプログラム格納エリア
内蔵 RAM	
H'FFFF 0000~07FF	消去プログラムダウンロード領域
0000	DPFR (戻り値: 1バイト)
0001~000F	システム領域 (15バイト)
0010~001F	消去処理エントリー
0020~002F	消去初期化エントリー
0030~07FF	消去初期化+消去プログラム
H'FFFF 0800~0FFF	書き込みプログラムダウンロード領域
0800	DPFR (戻り値: 1バイト)
0801~080F	システム領域 (15バイト)
0810~081F	書き込み処理エントリー
0820~082F	書き込み初期化エントリー
0830~0FFF	書き込み初期化+書き込みプログラム
H'FFFF 1000~107F	BUFF1 (書き込みデータ格納エリア1)
H'FFFF 1080~10FF	BUFF2 (書き込みデータ格納エリア2)
H'FFFF 1100~12F4	手続きアプリケーションプログラム実行エリア
H'FFFF B000~B003	コマンドエリア (4バイト) :command

以上の消去／書き込み手続きアプリケーションプログラム転送プログラムと消去／書き込み手続きアプリケーションプログラムリストを添付のリスト3、リスト4に示します。

7. 付録

リスト 1. コンペアマッチ処理 (コマンド解析) プログラム

```

#include      <stdio.h>
#include      <machine.h>
#include      "SH7058.h"
#include      "reg.h"
#pragma inline_asm(jmp_RAM1100)
void         jmp_RAM1100(void) {          /*転送後のジャンプ処理 */
    MOV.L    #H'FFFF1100,R0
    JMP      @R0
    NOP
}
extern unsigned char *_FWRT_BGN;        /* 手続きアプリケーションプログラム格納先頭アドレス */
extern unsigned char *_FWRT_END;        /* 手続きアプリケーションプログラム格納最終アドレス */
extern unsigned char *_FWRT_RAM;        /* 手続きアプリケーションプログラム転送先アドレス */

#pragma interrupt(CMT00)
void CMT00(void){
    unsigned char *src, *dst;
    CMT0.CMCSR &= 0x0041;                /*      コンペアマッチ割り込みフラグクリア      */
    if(command == FWRT){                /*      コマンド解析      */
        set_imask(0xF);                /*      割り込みマスク      */
        /*      手続きアプリケーションプログラムを内蔵 RAM へ転送      */
        for(src=_FWRT_BGN, dst=_FWRT_RAM; src<=_FWRT_END; src++, dst++){
            *dst = *src;
        }
        jmp_RAM1100();                /*      内蔵 RAM へジャンプ      */
    }
}
    
```

リスト 2. ユーザプログラムモード消去／書き込み手続きアプリケーションプログラムリスト

```

#include      <machine.h>
#include      <stdio.h>
#include      "SH7058.h"
#include      "reg.h"

#pragma inline_asm(NOP4)
static void NOP4(void){
    NOP
    NOP
    NOP
    NOP
}

void FWRT_main(void){ /*      0x00001000 → 0xFFFF1100      */
    unsigned long ERR_CHECK;
    unsigned long FPEFEQ;
    unsigned long FUBRA;
    unsigned long FPFR;
    unsigned long FEBS;
    unsigned long FMPAR;
    unsigned long FMPDR;

/* 消去プログラムダウンロード */
    FLASH.FTDAR = RAM00; /*      ダウンロード先アドレス設定(H'FFFF0000～)      */
    RAM00_TOP = 0xFF; /*      ダウンロード処理エラー確認用レジスタクリア      */
    FLASH.FECS = 0x01; /*      消去プログラム選択 (EPVB set)      */
    FLASH.FKEY = 0xA5; /*      消去プログラムダウンロード許可      */
    FLASH.FCCS |= 0x01; /*      ダウンロード要求 (SCO set)      */
    NOP4(); /*      NOP 命令×4      */
    FLASH.FKEY = 0x00; /*      FKEY clear      */
    if(RAM00_TOP != 0x00){ /*      ダウンロードエラー?      */
        command = EDER; /*      ダウンロードエラー処理 0x4544452      */
        while(1);
    }
}
    
```



```

/* 消去初期設定 */
FPEFEQ = 0x00000FA0; /* 40MHz H'0FA0 => D'4000 */
FUBRA = 0x00000000;
ERR_CHECK = FLASH_INIT(FPEFEQ,FUBRA);
/* FPEFEQ-->R4,FUBRA-->R5,ERR_CHECK-->R0 */
if(ERR_CHECK != 0x00000000){ /* 消去初期設定エラー? */
    command = EIER; /* 消去初期設定エラー処理 0x45494552 */
    while(1);
}
/* 消去処理 */
FLASH.FKEY = 0x5A; /* FKEY set */
FEBS = 0x00000000;
while(ERASE_block != 0x0000){ /* ブロック消去 */
    if((ERASE_block & 0x0001)==0x0001){
        ERR_CHECK = FLASH_ERASE(FEBS);
        /* FEBS-->R4,ERR_CHECK-->R0 */
        if(ERR_CHECK != 0x00000000){ /*消去エラー? */
            command = ERER; /*消去エラー処 0x45524552 */
            while(1);
        }
    }
    ERASE_block = (ERASE_block>>1);
    FEBS++;
}
FLASH.FKEY = 0x00; /* FKEY clear */
command = EEND; /* 消去正常完了 0x45454E44 */
/* 書き込みプログラムダウンロード */
FLASH.FTDAR = RAM08; /*ダウンロード先アドレス設定(H'FFFF0800) */
RAM08_TOP = 0xFF; /*ダウンロード処理エラー確認用レジスタクリア */
FLASH.FPCS = 0x01; /*書き込みプログラム選択(PVVS set) */
FLASH.FKEY = 0xA5; /*書き込みプログラムダウンロード許可 */
FLASH.FCCS |= 0x01; /*ダウンロード要求(SCO set) */
NOP4(); /*NOP 命令×4 */
FLASH.FKEY = 0x00; /*FKEY clear */
if(RAM08_TOP != 0x00){ /*ダウンロードエラー? */
    command = WDER; /*ダウンロードエラー処理 0x57444552 */
    while(1);
}
/* 書き込み初期設定 */
FPEFEQ = 0x00000FA0; /* 40MHz H'0FA0 => D'4000 */
FUBRA = 0x00000000;
ERR_CHECK = FLASHwr_INIT(FPEFEQ,FUBRA);
/* FPEFEQ-->R4,FUBRA-->R5,ERR_CHECK-->R0 */
if(ERR_CHECK != 0x00000000){ /* 初期設定エラー? */
    command = WIER; /* 初期設定エラー処理 0x57494552 */
    while(1);
}
WRITE_block = 0x00010000;
    
```

```

/* 書き込み処理          */
FLASH.FKEY = 0x5A;          /* FKEY set          */
FMPAR = 0x00002000;        /* 0x00002000~0x000FFFFF 書き込み */
while(command != DEND){    /* 書き込み終了?    */
    while(command != BUF1 && command != BUF2);
                                /* コマンド入力待ち 0x42554631 */
    if(command == BUF1){    /* BUFF1 のデータ書き込み? */
        FMPDR = 0xFFFF1000;
    }
    else{
        FMPDR = 0xFFFF1080;
    }
    ERR_CHECK = FLASH_WRITE(FMPDR,FMPAR);
                                /* FMPDR-->R4,FMPAR-->R5,ERR_CHECK-->R0 */
    if(ERR_CHECK != 0x00000000){ /* 書き込みエラー? */
        command = WRER;    /* 書き込みエラー処理 0x57524552 */
        while(1);
    }
    FMPAR += 0x80;
    command = WEND;        /* ページ書き込み終了 0x57454E44 */
}
FLASH.FKEY = 0x00;        /* FKEY clear      */
while(1);
}
    
```

リスト3. 消去／書き込み手続きアプリケーションプログラム転送プログラムリスト

```

/* ユーザブートマット上メインプログラム          */
#include      <machine.h>
#include      <stdio.h>
#include      "SH7058FCC.h"
#include      "reg.h"
#pragma inline_asm(jmp_RAM)
static void jmp_RAM(void) {                          /* 転送後のジャンプ処理          */
    MOV.L    #H'FFFF1100,R0
    JMP      @R0
    NOP
}
extern unsigned char *_FWRT_BGN; /* 手続きアプリケーションプログラム格納先頭アドレス */
extern unsigned char *_FWRT_END; /* 手続きアプリケーションプログラム格納最終アドレス */
extern unsigned char *_FWRT_RAM; /* 手続きアプリケーションプログラム転送先アドレス */

void uboot_main(void) {                              /* 0x0001000          */
    unsigned char *src, *dst;
    /* ユーザブートマットから内蔵RAMへコピー */
    for(src=_FWRT_BGN, dst=_FWRT_RAM; src<=_FWRT_END; src++, dst++){
        *dst = *src;
    }
    jmp_RAM();
    while(1);
}
    
```

リスト4. ユーザブートモード消去／書き込み手続きアプリケーションプログラムリスト

```

#include      <machine.h>
#include      <stdio.h>
#include      "SH7058.h"
#include      "reg.h"
#pragma inline_asm(NOP4)
static void NOP4(void){
    NOP
    NOP
    NOP
    NOP
}

void e_main(void){
    /*      0xFFFF2000      */
    unsigned long ERR_CHECK,FPEFEQ,FUBRA,FPPR,FEBS,FMPAR,FMPDR;
/* 消去プログラムダウンロード処理      */
    FLASH.FTDAR = RAM00;      /*      ダウンロード先アドレス設定(H'FFFF0000)      */
    RAM00_TOP = 0xFF;      /*      ダウンロード処理エラー確認用レジスタクリア      */
    FLASH.FECS = 0x01;      /*      消去プログラム選択(EPVB set)      */
    FLASH.FKEY = 0xA5;      /*      消去プログラムダウンロード許可      */
    FLASH.FCCS |= 0x01;      /*      ダウンロード要求(SCO set)      */
    NOP4();      /*      NOP 命令×4      */
    FLASH.FKEY = 0x00;      /*      FKEY clear      */
    if(RAM00_TOP != 0x00){      /*      ダウンロードエラー?      */
        command = EDER;      /*      ダウンロードエラー処理 0x45444552      */
        while(1);
    }
/* 消去初期設定      */
    FPEFEQ = 0x00000FA0;      /*      40MHz      H'0FA0 => D'4000      */
    FUBRA = 0x00000000;
    ERR_CHECK = FLASH_INIT(FPEFEQ,FUBRA);
/*      FPEFEQ-->R4,FUBRA-->R5,ERR_CHECK-->R0      */
    if(ERR_CHECK != 0x00000000){      /*      初期設定エラー?      */
        command = EIER;      /*      初期設定エラー処理 0x45494552      */
        while(1);
    }
    FLASH.FMATS = 0x55;      /*      ユーザマットにマット切り替え      */
    NOP4();      /*      NOP 命令×4      */
}
    
```

```

/* 消去処理 */
FLASH.FKEY = 0x5A; /* FKEY set */
FEBS = 0x00000000;
ERASE_block = 0xFFFF; /* all erase */
while(ERASE_block != 0x0000){ /* ブロック消去処理 */
    if((ERASE_block & 0x0001)==0x0001){
        ERR_CHECK = FLASH_ERASE(FEBS); /* FEBS-->R4, ERR_CHECK-->R0 */
        if(ERR_CHECK != 0x00000000){ /* 消去エラー? */
            command = EDER; /* 消去エラー処理 0x45524552 */
            while(1);
        }
    }
    ERASE_block = (ERASE_block>>1);
    FEBS++;
}
FLASH.FKEY = 0x00; /* FKEY clear */
FLASH.FMATS = 0xAA; /* ユーザーブートマットにマット切り替え */
NOP4(); /* NOP 命令×4 */
command = EEND; /* 消去正常完了 0x45454E44 */
while(command == EEND);
/* 書き込みプログラムダウンロード処理 */
FLASH.FTDAR = RAM08; /* ダウンロード先アドレス設定 (H'FFFF0800) */
RAM08_TOP = 0xFF; /* ダウンロード処理エラー確認用レジスタクリア */
FLASH.FPCS = 0x01; /* 書き込みプログラム選択 (PPVS set) */
FLASH.FKEY = 0xA5; /* 書き込みプログラムダウンロード許可 */
FLASH.FCCS |= 0x01; /* ダウンロード要求 (SCO set) */
NOP4(); /* NOP×4 */
FLASH.FKEY = 0x00; /* FKEY clear */
if(RAM08_TOP != 0x00){ /* ダウンロードエラー? */
    command = WDER; /* ダウンロードエラー処理 0x57444552 */
    while(1);
}
/* 書き込み初期設定 */
FPEFEQ = 0x0000FA0; /* 40MHz H'0FA0 => D'4000 */
FUBRA = 0x00000000;
ERR_CHECK = FLASHwr_INIT(FPEFEQ, FUBRA);
/* FPEFEQ-->R4, FUBRA-->R5, ERR_CHECK-->R0 */
if(ERR_CHECK != 0x00000000){ /* 初期設定エラー? */
    command = WIER; /* 初期設定エラー 0x57494552 */
    while(1);
}
FLASH.FMATS = 0x55; /* ユーザマットにマット切り替え */
NOP4(); /* NOP 命令×4 */
    
```

```

/* 書き込み処理 */
FLASH.FKEY = 0x5A; /* FKEY set */
FMPAR=0x00000000;
while(FMPAR < 0x00100000){ /* 書き込み終了 */
    while(command != BUF1 && command != BUF2); /* コマンド入力待ち 0x42554631 */
    if(command == BUF1){ /* コマンド 0x42554631 */
        FMPDR = 0xFFFF1000;
    }
    else{
        FMPDR = 0xFFFF1080;
    }
    ERR_CHECK = FLASH_WRITE(FMPDR,FMPAR);
    /* FMPDR-->R4,FMPAR-->R5,ERR_CHECK-->R0 */
    if(ERR_CHECK != 0x00000000){ /* 書き込みエラー? */
        command = WRER; /* 書き込みエラー処理 0x57524552 */
        while(1);
    }
    FMPAR += 0x80;
    command = WROK; /* */
}
FLASH.FKEY = 0x00; /* FKEY clear */
FLASH.FMATS = 0xAA; /* ユーザーブートマットにマット切り替え */
NOP4(); /* NOP 命令×4 */
command = WEND;; /* 書き込み完了メッセージ 0x57454E44 */
while(1);
}
    
```

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したのですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。