
SH7253 SH7256R グループ

R01AN1639JJ0100

Rev.1.00

2013.04.02

EEPROM 書き込み/消去

要旨

本アプリケーションノートは、専用シーケンサ(FCU)による内蔵 EEPROM の書き込み/消去の動作例についてまとめたものです。

専用シーケンサ(FCU)に対してコマンドを発行することで内蔵 EEPROM のデータ書き込み/消去を行い、CPU の負担を軽減することができます。

本アプリケーションノートに掲載されているタスク例は動作確認済みですが、実際にご使用になる場合には、必ず動作環境を確認の上ご使用くださいますようお願いいたします。

【注】 本アプリケーションノートのサンプルコードは SH7254R グループ用に作成しています。SH7253 グループでご使用の場合は、本文 6. SH7253 グループでご使用の場合の変更点に従い、修正してください。SH7256 グループでご使用の場合は、製品レジスタ定義ファイル `iodefine.h` を SH7256R グループ用のものと差替えるのみで構いません。ソース上の各レジスタ名は、`iodefine.h` に合わせ、修正してください。

動作確認デバイス

SH72544R

適用条件

マイコン : SH72544R

動作周波数 : 内部クロック 200MHz

周辺クロック 40MHz

動作モード : シングルチップモード

C コンパイラ : ルネサス エレクトロニクス製

SuperH RISC engine Standard Toolchain (V.9.4.0.0)

目次

1. 仕様	3
2. 使用機能説明	4
3. 動作概要	7
3.1 EEPROM 書き込み全体の流れ	7
3.2 各動作の流れ	8
4. ソフトウェア説明	12
4.1 main 関数	13
4.2 init_eeeprom ルーチン	15
4.3 copy_firm ルーチン	18
4.4 fcu_md_pe ルーチン	20
4.5 fcu_eebck 関数	22
4.6 fcu_eerm 関数	24
4.7 fcu_eeprg 関数	26
4.8 fcu_res 関数	28
4.9 fcu_chk0 関数	29
4.10 fcu_chk01 関数	31

1. 仕様

SH7254R 内蔵 EEPROM への書き込み/消去を専用シーケンサ(FCU)へのコマンド発行で行います。

EEPROM へ書き込む値は、予めタスク内で用意されている値を用います。

FCU に対してブランクチェックコマンドを発行し、EEPROM 領域にデータが書き込まれているか確認を行います。対象の領域にデータが書き込まれていた場合、消去動作を行うブロックイレイズコマンドを発行し、対象のデータブロックを消去します。

FCU のエラーを検出する為に、FCU にコマンドを発行した後にエラーフラグを確認しています。

表 1.1 データ/アドレス一覧

	Address	Data
ROM (読み出し元)	0x00008000	H'0000~H'0003(H'003E) *2
EEPROM (書き込み先)	0x80102080	Unknown

*2:128 バイトの場合

2. 使用機能説明

図 2.1に EEPROM のブロック図を示します。また、以下に本タスクで使用した FCU 機能のレジスタについて説明します。

- フラッシュモードレジスタ (FMODR)

FMODR は FCU の動作モードを設定するレジスタです。本タスク例では、ブランクチェックコマンドをレジスタリードモードで行う様に設定しています。

- フラッシュアクセスエラー割り込み許可レジスタ (FAEINT)

FAEINT は、フラッシュインタフェースエラー割り込みの出力許可/禁止を設定するレジスタです。本タスク例では、すべての割り込みを禁止に設定します。

- EEPROM ブランクチェックレジスタ (EEPBCCNT)

EEPBCCNT は、ブランクチェック時の対象領域のアドレス及びサイズを決定するレジスタです。本タスク例では 8K バイト単位でブランクチェックを行う設定にしています。

EEPROM ブランクチェックステータスレジスタ (EEPBCSTAT)

EEPBCSTAT は、ブランクチェックの結果が格納されるレジスタです。

- フラッシュリセットレジスタ (FRESETR)

FRESETR は FCU と EEPROM を初期化のために使用するレジスタです。本タスク例では、消去、書き込み、ブランクチェック時にタイムアウトが発生した場合に、FRESETR は FCU を初期化します。

- フラッシュステータスレジスタ 0 (FSTATR0)

FSTATR0 は、FCU の状態を確認するためのレジスタです。本タスク例では、FCU の処理状態の確認、または FCU の不正 コマンド、EEPROM への不正アクセス、消去処理中のエラー、書き込み処理中のエラー判定に使用します。

- フラッシュステータスレジスタ 1 (FSTATR1)

FSTATR1 は、FCU の状態を示します。本タスク例では、FCU の処理結果の確認に使用します。

- EEPROM 書き込み/消去許可レジスタ 0 (EEPWE0)

EEPWE0 は、データマットの DB00-DB07 ブロックの書き込み/消去の許可/禁止をするためのレジスタです。本タスク例では全領域の書き込み/消去を許可しています。

- EEPROM 書き込み/消去許可レジスタ 1 (EEPWE1)

EEPWE1 は、データマットの DB08-DB15 ブロックの書き込み/消去の許可/禁止をするためのレジスタです。本タスク例では全領域の書き込み/消去を許可しています。

- EEPROM 読み出し許可レジスタ 0 (EEPWE0)

EEPWE0 は、データマットの DB00-DB07 ブロックの読み出しの許可/禁止をするためのレジスタです。本タスク例では全領域の読み出しを許可しています。

- EEPROM 読み出し許可レジスタ 1 (EEPRE1)

EEPWE0 は、データマットの DB08-DB15 ブロックの読み出しの許可/禁止をするためのレジスタです。本タスク例では全領域の読み出しを許可しています。

- フラッシュ P/E モードエントリレジスタ (FENTRYR)

FENTRYR は、EEPROM を P/E モードに設定するレジスタです。本タスク例では書き込み/消去の際に P/E モードに切り替えています。

- EEPROM マット選択レジスタ (EEPMAT)

EEPMAT は、EEPROM のマットを切り替えるために使用するレジスタです。本タスク例では、データマットを選択します。

- FCU RAM イネーブルレジスタ (FCURAME)

FCURAME は、FCU RAM 領域へのアクセス許可/禁止をするためのレジスタです。本タスク例では FCU ファームウェア 転送を行う際に FCU RAM 領域へのアクセスを許可しています。

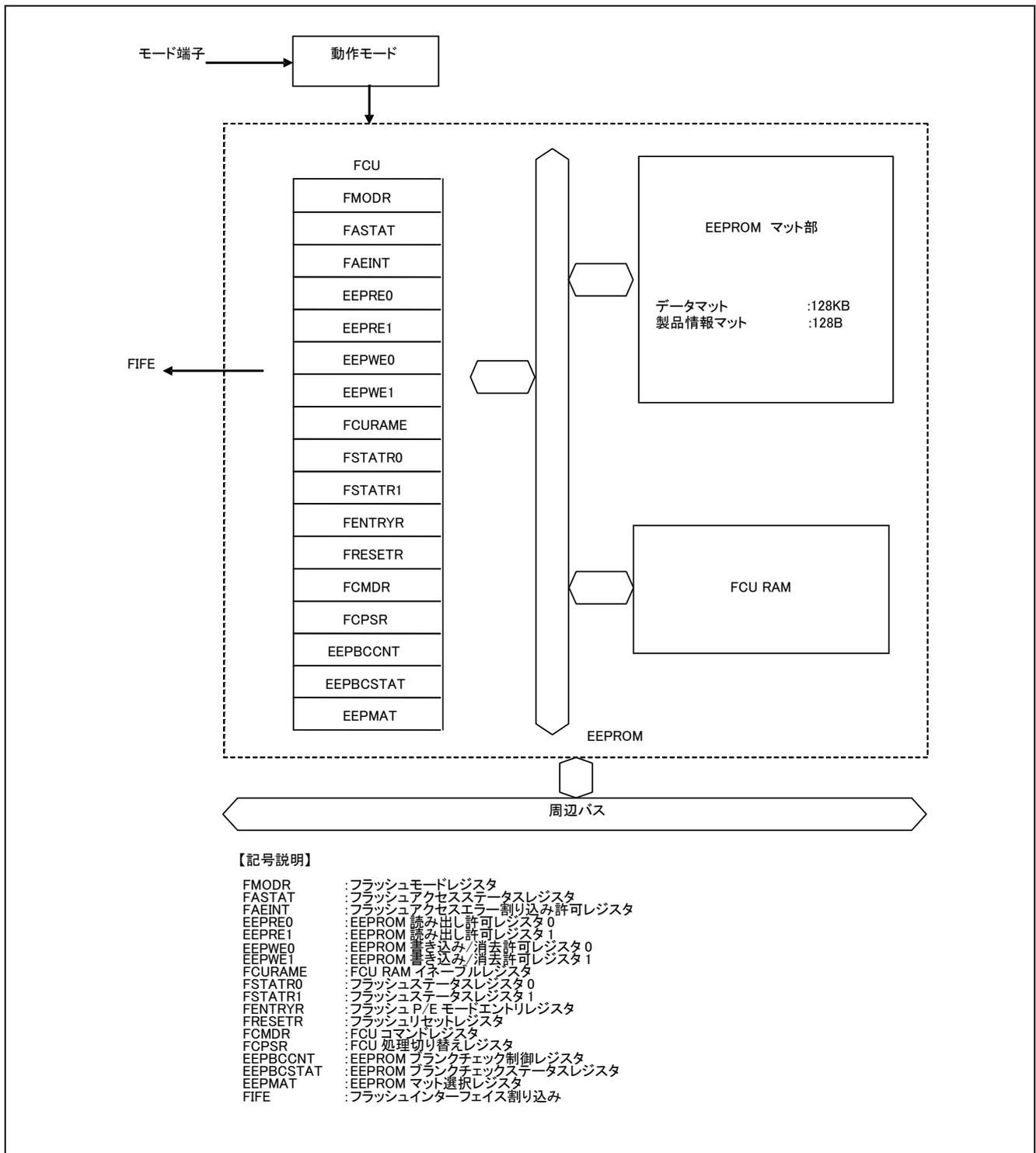


図 2.1 EEPROM ブロック図

3. 動作概要

3.1 EEPROM 書き込み全体の流れ

図3.1の流れに沿って、EEPROMへの書き込みを行います。CPUで消去状態のEEPROMを読み出すと不定値が読み出されますので、消去状態の確認にはブランクチェックコマンドを使用して下さい。

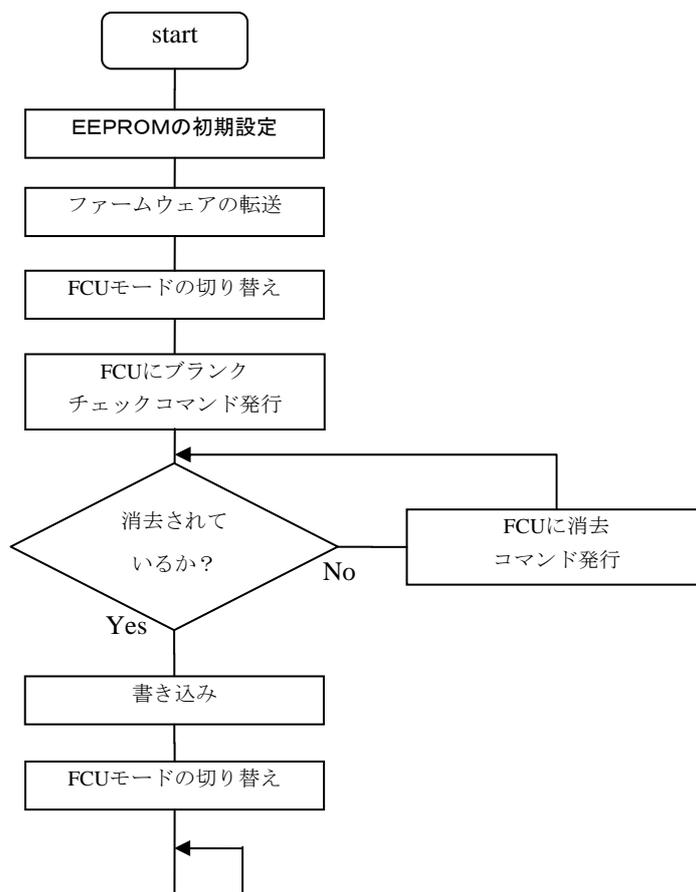


図 3.1 EEPROM ブランクチェックから書き込みへの流れ

3.2 各動作の流れ

本タスク例では、以下の手順で EEPROM の書き込みを行います。

① EEPROM の初期設定

EEPROM の書き込み/消去、読み出しプロテクト解除を設定します。

② FCU ファームウェアの転送

②-1:FCURAM へアクセスする為に FCURAME に 1 をセットし、FCURAM 領域へのアクセスを許可します。

②-2:ROM 上の FCU ファーム領域から FCURAM 領域へ FCU 用ファームウェアを転送します。

図 3.2 に FCU ファームウェア転送のイメージ図を示します。

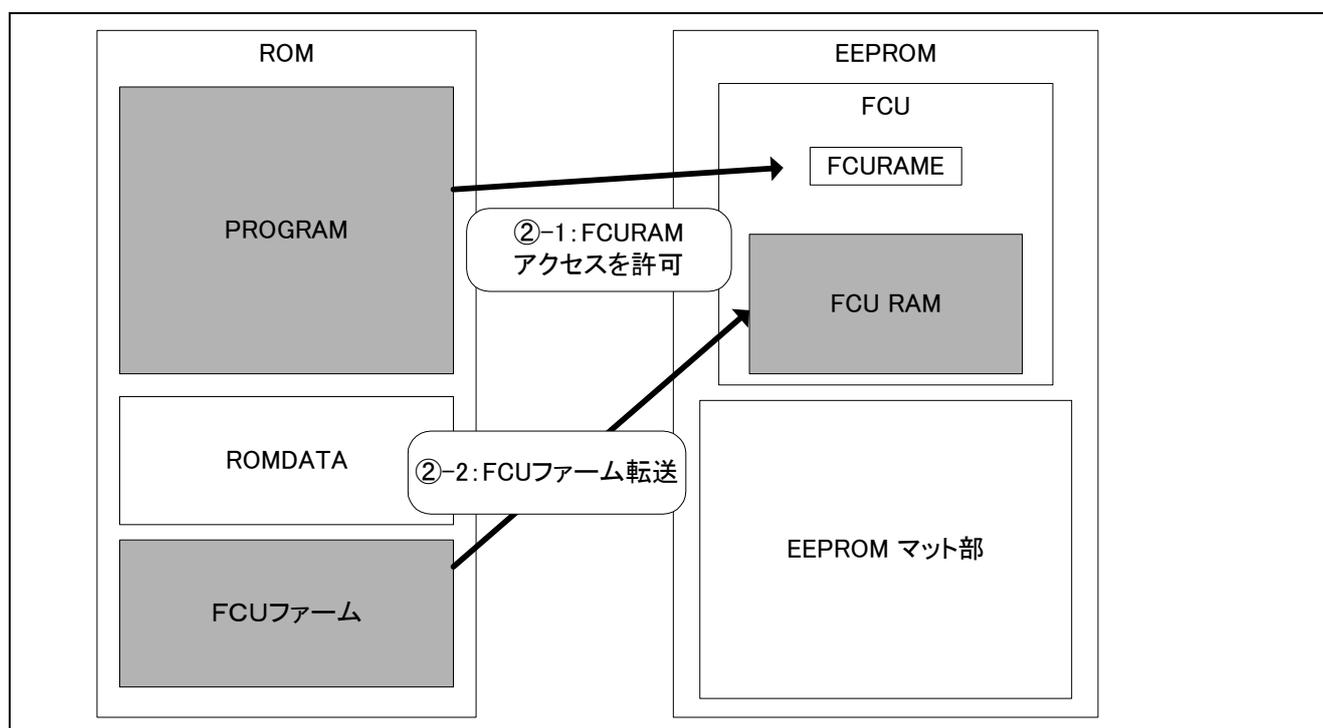


図 3.2 FCU ファームウェア転送のイメージ図

③ FCU のモード変更

FCU がコマンドの受付を可能にするために、FENTRYR レジスタを設定して FCU のモードを初期値の ROM/EEPROM リードモードから EEPROM P/E ノーマルモードへ移行します。これ以降、FCU コマンドを発行する事が可能になります。

④ データマットのブランクチェック

④-1:表 3.1 に示した順で FCU にブランクチェックコマンドを発行します。

- ④-2:FCU が正常にプログラムコマンドを受けると、コマンドが発行されたアドレス領域のブランクチェック処理を実行します。
- ④-3:対象領域のブランクチェックの結果がブランクチェックステータスレジスタ内のブランクチェックステータスビット(BCST)にセットされます。

図 3.3 にブランクチェックのイメージ図を示します。

表 3.1 ブランクチェックコマンド発行シーケンス

コマンドサイクル	アドレス	データ	アクセスサイズ	説明
1	H'80102000*1	H'71	バイト	ブランクチェックコマンド
2	H'80102000*1	H'D0	バイト	FCU 動作開始

*1:ブランクチェックする EEPROM Block 内の任意のアドレス

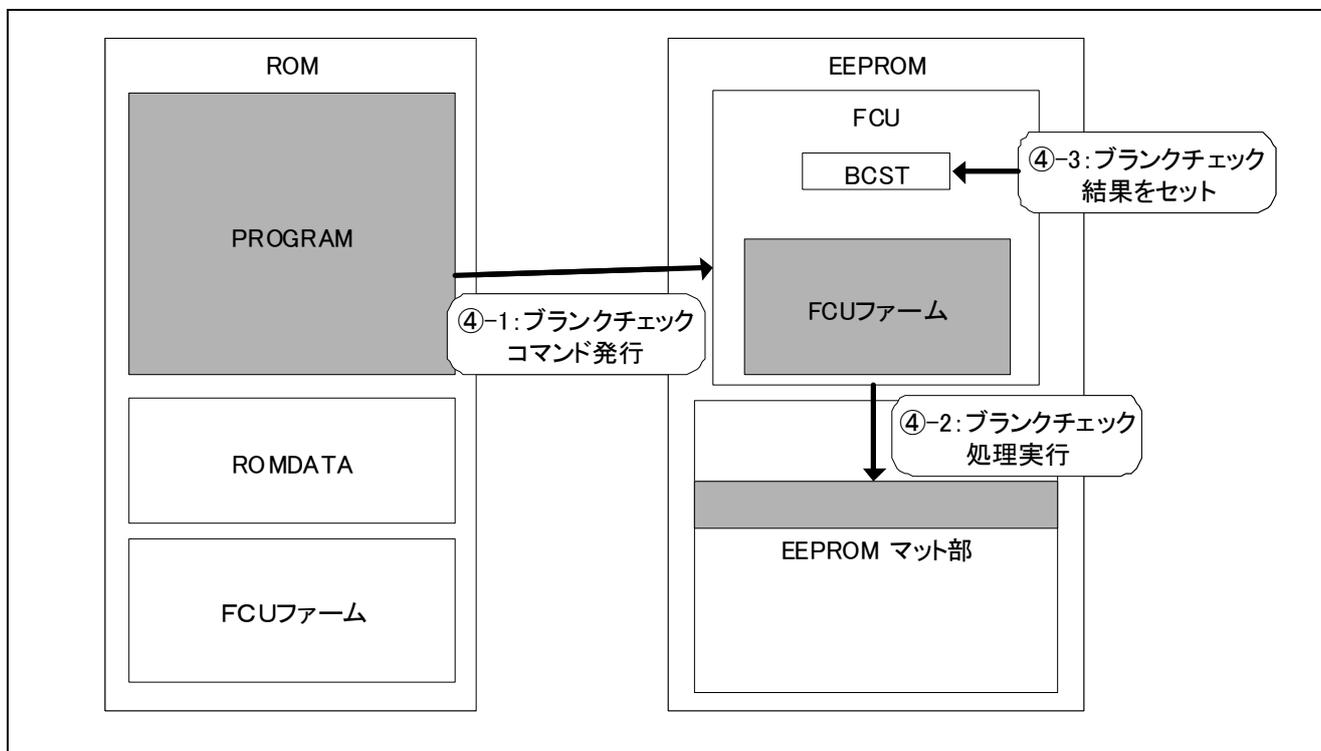


図 3.3 ブランクチェックのイメージ図

⑤ データマットのブロック消去

⑤-1:表 3.2 に示した順で FCU にブロックイレーズコマンドを発行します。

⑤-2:FCU が正常にプログラムコマンドを受けると、コマンドが発行されたアドレス領域の消去処理を行います。消去処理が行われている間は FRDY が自動的に 0 にクリアされます。

⑤-3:消去処理が終了すると FRDY に 1 がセットされます。

図 3.4 にブロック消去のイメージ図を示します。

表 3.2 ブロックイレーズコマンド発行シーケンス

コマンド サイクル	アドレス	データ	アクセスサイズ	説明
1	H'80102000*1	H'20	バイト	ブロックイレーズコマンド
2	H'80102000*1	H'D0	バイト	FCU 動作開始

*1:消去する EEPROM Block 内の任意のアドレス

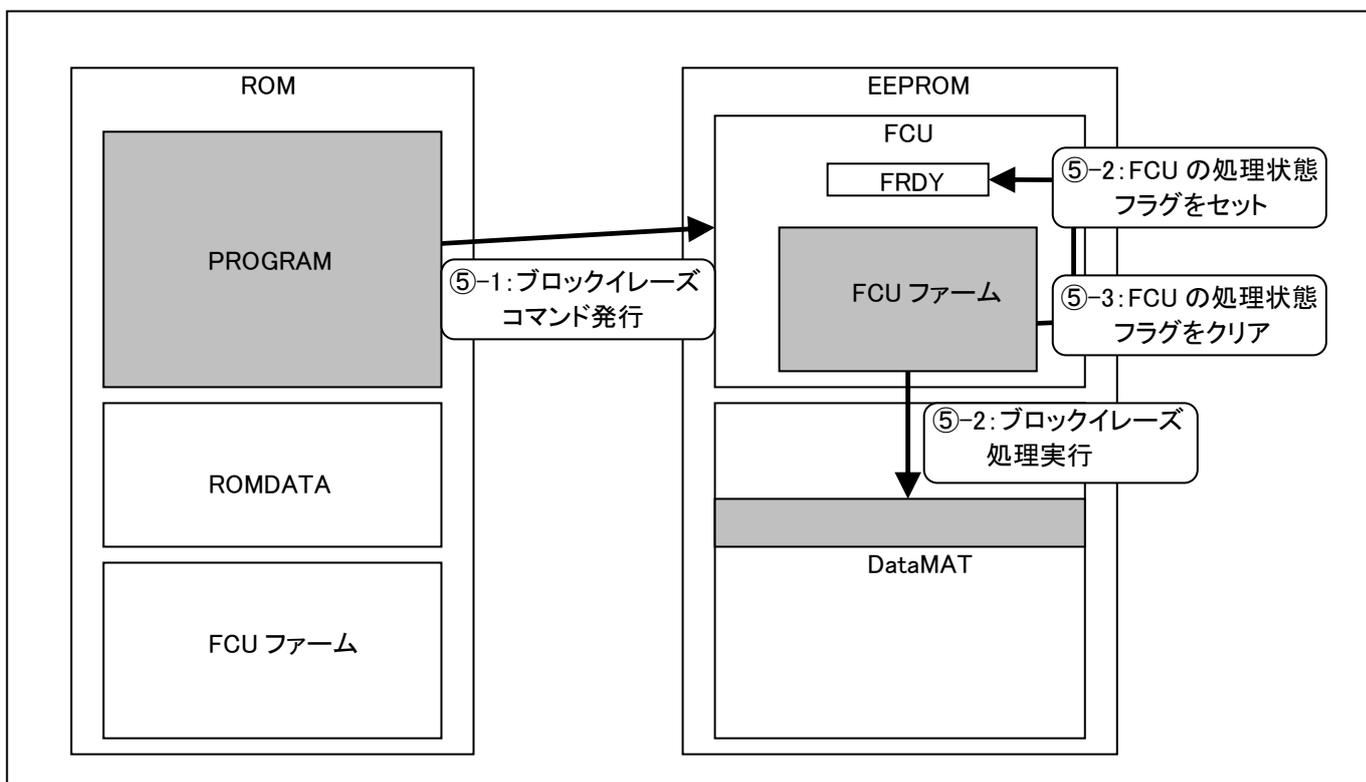


図 3.4 ブロック消去のイメージ図

- ⑥ プログラムコマンドの発行による、書き込み対象領域へのデータ書き込み
 - ⑥-1:表 3.3 に示した順で、FCUにプログラムコマンドを発行します。
 - ⑥-2:FCU が正常にプログラムコマンドを受けると、コマンドが発行されたアドレス領域にデータの書き込み処理を行います。書き込み処理が行われている間はFRDYが自動的に0にクリアされます。
 - ⑥-3:書き込み処理が終了すると FRDYに1がセットされます。
- 本タスク例で使用するプログラムコマンドシーケンスを表 3.3 に示します。
- 図 3.5 にデータ書き込みのイメージ図を示します。

表 3.3 プログラムコマンド発行シーケンス

コマンドサイクル	アドレス	データ	アクセスサイズ	説明
1	H'80102080 ^{*1}	H'E8	バイト	プログラムコマンド
2	H'80102080 ^{*1}	H'04 (H'40) ^{*2}	バイト	書き込みデータサイズ(8バイト/128バイト)
3~6 (3~66) ^{*2}	H'80102080	H'0000~ H'0003 (H'003E) ^{*2}	ワード	書き込み先の先頭アドレス指定と書き込みデータの送信
7 (67) ^{*2}	H'80102080 ^{*1}	H'D0	バイト	FCU 動作開始

*1:任意の EEPROM 領域内のアドレス *2:128 バイトの場合

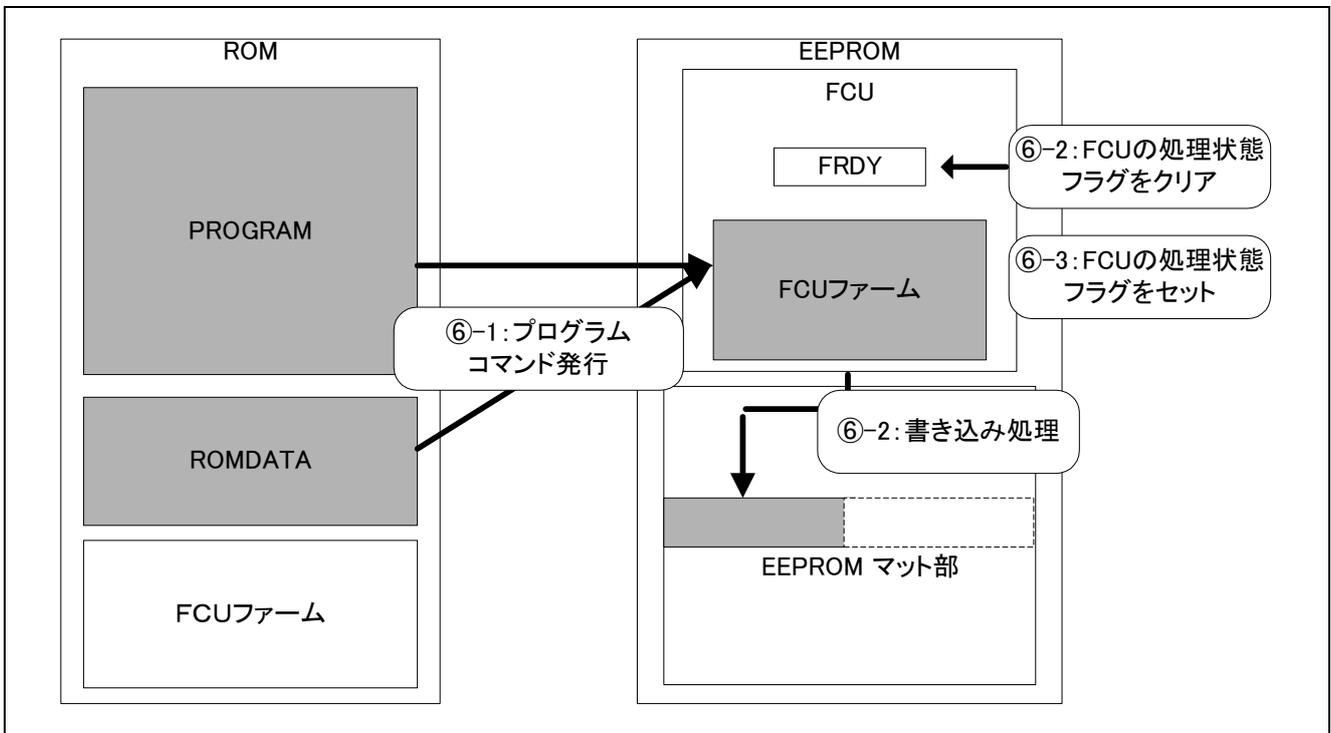


図 3.5 データ書き込みのイメージ図

4. ソフトウェア説明

関数一覧

表 4.1 関数一覧

関数名	内容
main	メインルーチン 各プログラムの呼び出し
init_eeprom	EEPROM の初期化
copy_firm	FCU ファームウェアの転送
md_fcu_pe	FCU を EEPROM P/E ノーマルモードに移行
fcu_eebck	ブランクチェックコマンド発行
fcu_eerm	ブロックイレーズコマンド発行
fcu_eeprg	書き込みコマンド発行
fcu_res	FCU,ROM,EEPROM のリセット
fcu_chk0	FSTART0 のエラーフラグチェック
fcu_chk01	FSTART0 と FSTATR1 のエラーフラグチェック.

4.1 main 関数

(1) 機能概要

各関数のコールを行います。

(2) 引数

無し

(3) 戻り値

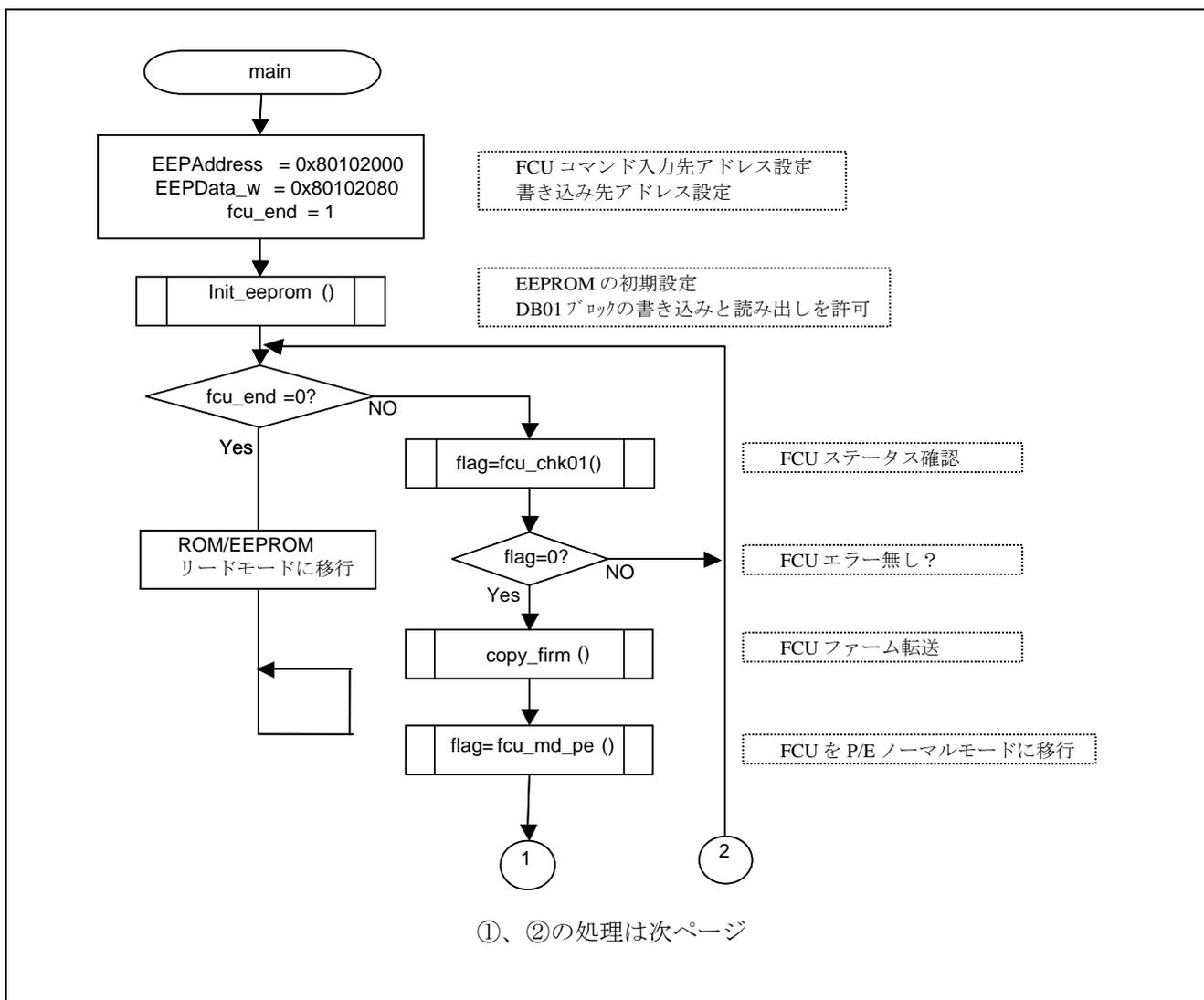
無し

(4) 使用内部レジスタ

表 4.2 EEPROM ブランクチェックステータスレジスタ (EEPBCSTAT)

ビット	ビット名	設定値	備考
0	BCST	-	ブランクチェックステータスビット 0:ブランクチェック領域は消去されています (blank)。 1:ブランクチェックエリアに値が存在しています。

(5) フローチャート



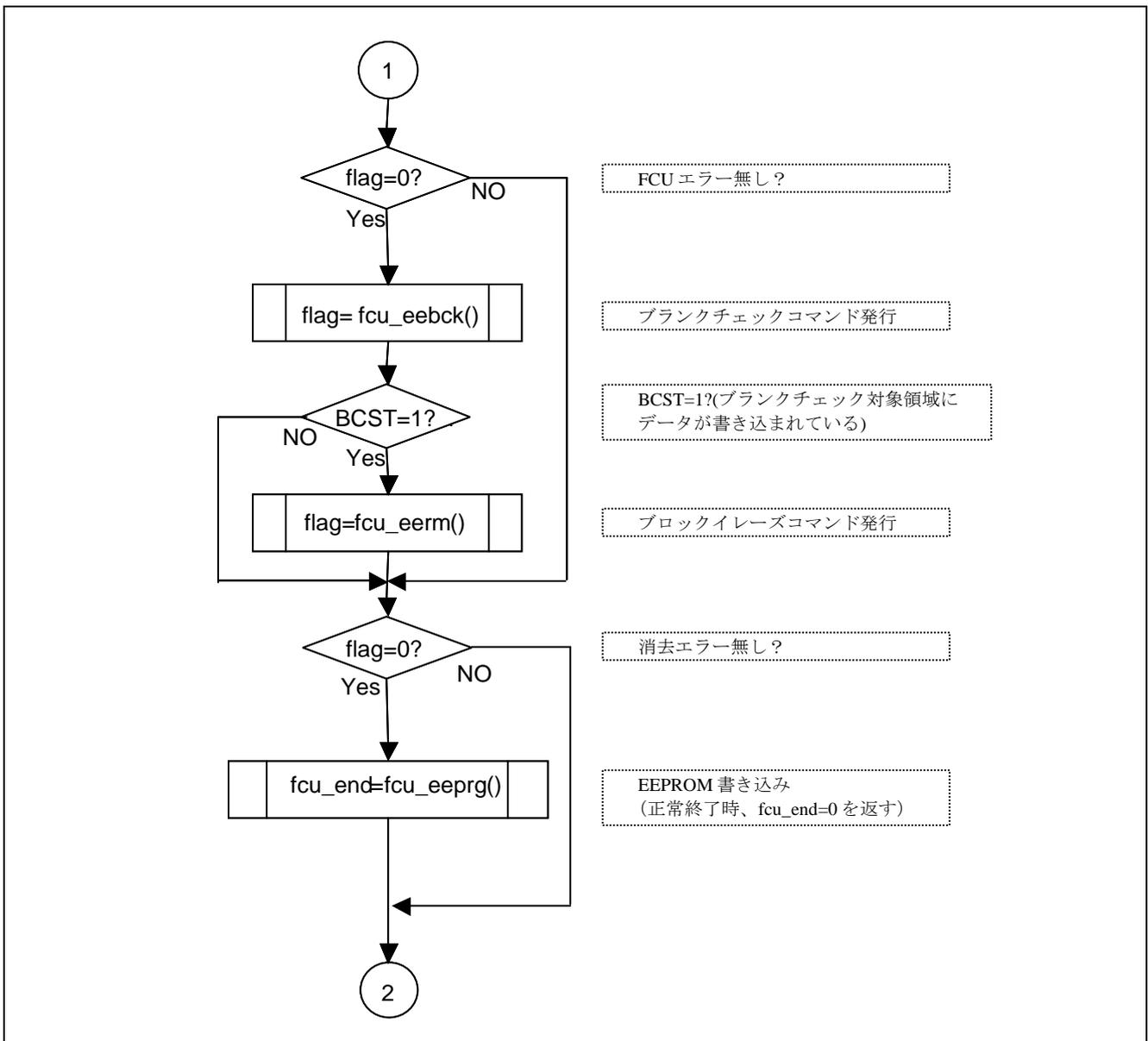


図 4.1 main 関数フローチャート

4.2 init_eeprom ルーチン

(1) 機能概要

EEPROM の書き込み/読み出しを許可します。

(2) 引数

無し

(3) 戻り値

無し

(4) 使用内部レジスタ

表 4.3 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	ビット名	設定値	備考
7	ROMAEIE	B'0	ROM アクセス違反割り込みイネーブル
4	CMDLKIE	B'0	FCU コマンドロック割り込みイネーブル
3	EEPAEIE	B'0	EEPROM アクセス違反割り込みイネーブル
2	EEPIFEIE	B'0	EEPROM 命令フェッチ違反割り込みイネーブル
1	EEPRPEIE	B'0	EEPROM リードプロテクト違反割り込みイネーブル
0	EEPWPEIE	B'0	EEPROM 書き込み/消去プロテクト違反割り込みイネーブル

表 4.4 EEPROM 読み出し許可レジスタ 0 (EEPRE0)

ビット	ビット名	設定値	備考
15-8	KEY	H'2D	キーコード 書き込みデータは保持されません。
7	DBRE07	B'0	ブロック読み出し許可ビット 0:読み出し禁止 1:読み出し許可
6	DBRE06	B'0	
5	DBRE05	B'0	
4	DBRE04	B'0	
3	DBRE03	B'0	
2	DBRE02	B'0	
1	DBRE01	B'1	
0	DBRE00	B'0	

表 4.5 EEPROM 読み出し許可レジスタ 1 (EEPRE1)

ビット	ビット名	設定値	備考
15-8	KEY	H'D2	キーコード 書き込みデータは保持されません。
7	DBRE15	B'0	ブロック読み出し許可ビット 0:読み出し禁止 1:読み出し許可
6	DBRE14	B'0	
5	DBRE13	B'0	
4	DBRE12	B'0	
3	DBRE11	B'0	
2	DBRE10	B'0	
1	DBRE09	B'0	
0	DBRE08	B'0	

表 4.6 EEPROM 書き込み/消去許可レジスタ 0 (EEPWE0)

ビット	ビット名	設定値	備考
15-8	KEY	H'1E	キーコード 書き込みデータは保持されません。
7	DBWE07	B'0	ブロック読み出し許可ビット 0:読み出し禁止 1:読み出し許可
6	DBWE06	B'0	
5	DBWE05	B'0	
4	DBWE04	B'0	
3	DBWE03	B'0	
2	DBWE02	B'0	
1	DBWE01	B'1	
0	DBWE00	B'0	

表 4.7 EEPROM 書き込み/消去許可レジスタ 1 (EEPWE1)

ビット	ビット名	設定値	備考
15-8	KEY	H'E1	キーコード 書き込みデータは保持されません。
7	DBWE15	B'0	ブロック読み出し許可ビット 0:読み出し禁止 1:読み出し許可
6	DBWE14	B'0	
5	DBWE13	B'0	
4	DBWE12	B'0	
3	DBWE11	B'0	
2	DBWE10	B'0	
1	DBWE09	B'0	
0	DBWE08	B'0	

(5) フローチャート

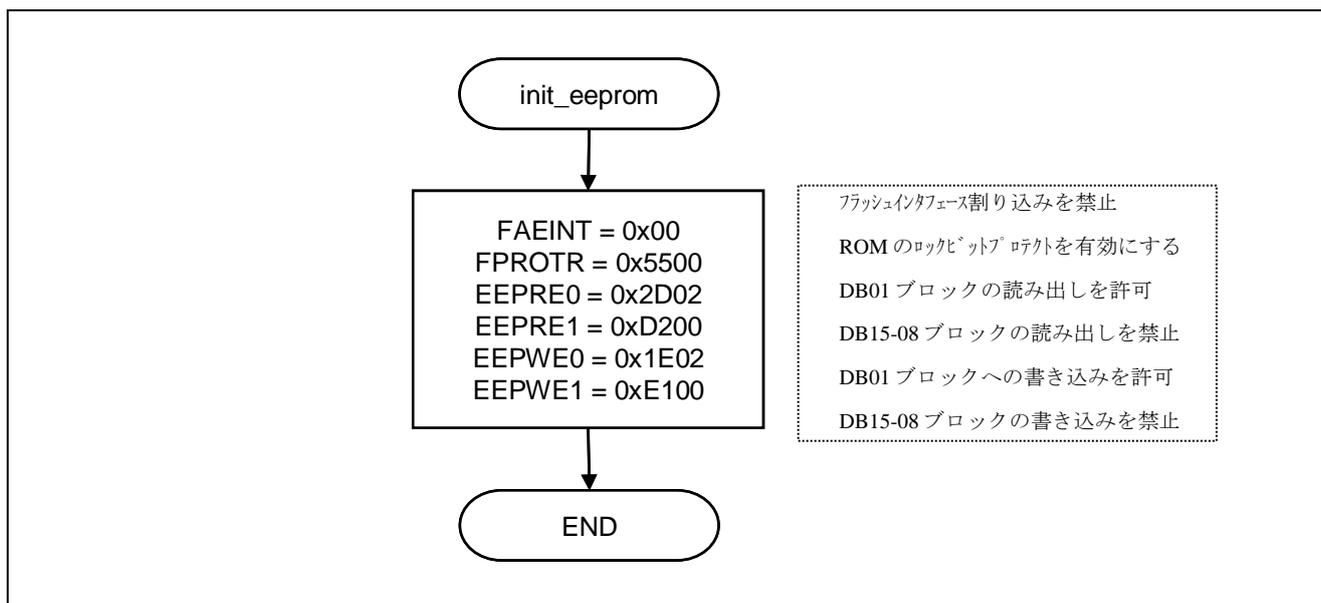


図 4.2 init_eeprom 関数フローチャート

4.3 copy_firm ルーチン

(1) 機能概要

ROM上のFCUファームウェアを、RAM上のファームウェア領域に転送します。

(2) 引数

無し

(3) 戻り値

無し

(4) 使用内部レジスタ

表 4.8 EEPROM マット選択レジスタ (EEPSTAT)

ビット	ビット名	設定値	備考
15-8	KEY	H'B3	キーコード 書き込みデータは保持されません。
0	EEPSEL	B'0	EEPROM マット選択ビット 0: データマット選択 1: 製品情報マット選択

表 4.9 FCURAM イネーブルレジスタ (FCURAME)

ビット	ビット名	設定値	備考
15-8	KEY	H'C4	キーコード 書き込みデータは保持されません。
0	FCRME	B'1	FCURAM enable 0: FCURAM 領域への書き込み禁止 1: FCURAM 領域への書き込み許可

表 4.10 フラッシュ P/E モードエン트리レジスタ (FENTRYR)

ビット	ビット名	設定値	備考
15-8	KEY	H'AA	キーコード 書き込みデータは保持されません。
7	FENTRTD	B'0	EEPROM P/E モードエン트리ビット 0:EEPROM はリードモード 1:EEPROM は P/E モード
5	FENTRY5	B'0	ROM P/E モードエン트리ビット 5
4	FENTRY4	B'0	ROM P/E モードエン트리ビット 4
3	FENTRY3	B'0	ROM P/E モードエン트리ビット 3
2	FENTRY2	B'0	ROM P/E モードエン트리ビット 2
1	FENTRY1	B'0	ROM P/E モードエン트리ビット 1
0	FENTRY0	B'0	ROM P/E モードエン트리ビット 0

(5) フローチャート

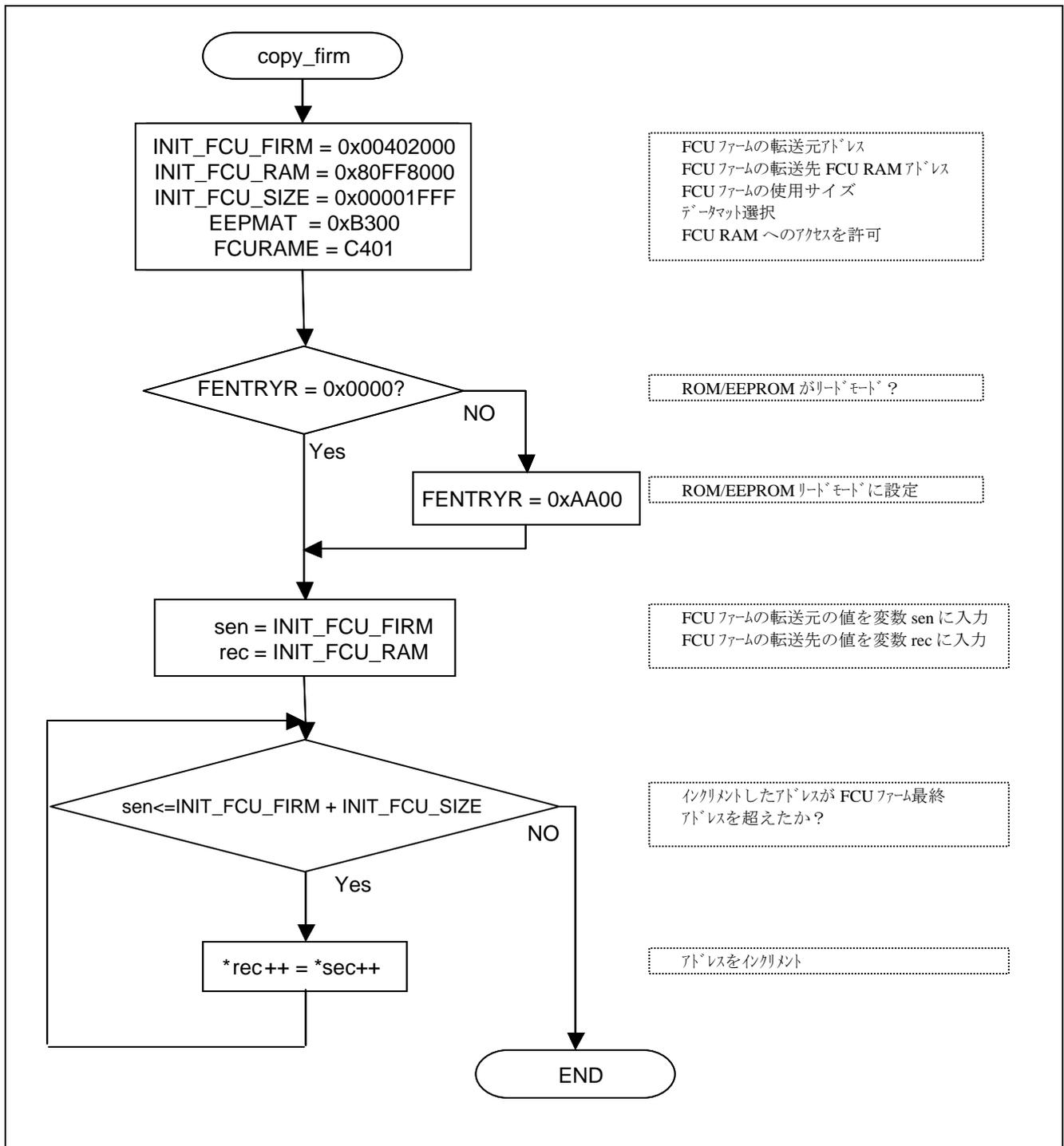


図 4.3 copy_firm 関数フローチャート

4.4 fcu_md_pe ルーチン

(1) 機能概要

FCU を P/E ノーマルモードに移行させます。

(2) 引数

無し

(3) 戻り値

unsigned char fcu_pe_flag

表 4.11 fcu_md_pe 戻り値

戻り値	内容
H'00	モード移行時にエラーは無し
H'01	FCU はモード変更命令を正常に実行していません

(4) 使用内部レジスタ

表 4.12 フラッシュアクセスステータスレジスタ (FASTAT)

ビット	ビット名	設定値	備考
7	ROMAE	B'0	ROM アクセス違反ビット
4	CMDLK	B'0	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 1:FCU はコマンドロック状態
3	EEPAE	B'0	EEPROM アクセス違反ビット 0:EEPROM アクセス違反無し 1:EEPROM アクセス違反発生
2	EEPIFE	B'0	EEPROM 命令フェッチビット 0: EEPROM 命令フェッチ違反無し 1: EEPROM 命令フェッチ違反あり
1	EEPRPE	B'0	EEPROM リードプロテクト違反ビット 0: EEPRE0,1 設定に違反した EEPROM 読み出し無し 1: EEPRE0,1 設定に違反した EEPROM 読み出し発生
0	EEPWPE	B'0	EEPROM 書き込み/消去プロテクト違反ビット 0: EEPWE 0,1 設定に違反した EEPROM 書き込み無し 1: EEPWE 0,1 設定に違反した EEPROM 書き込み発生

表 4.13 フラッシュ P/E モードエントリレジスタ (FENTRYR)

ビット	ビット名	設定値	備考
15-8	KEY	H'AA	キーコード 書き込みデータは保持されません。
7	FENTRTD	B'1	EEPROM P/E モードエントリビット 0:EEPROM はリードモード 1:EEPROM は P/E モード
5	FENTRY5	B'0	ROM P/E モードエントリビット 5
4	FENTRY4	B'0	ROM P/E モードエントリビット 4
3	FENTRY3	B'0	ROM P/E モードエントリビット 3
2	FENTRY2	B'0	ROM P/E モードエントリビット 2
1	FENTRY1	B'0	ROM P/E モードエントリビット 1
0	FENTRY0	B'0	ROM P/E モードエントリビット 0

(5) フローチャート

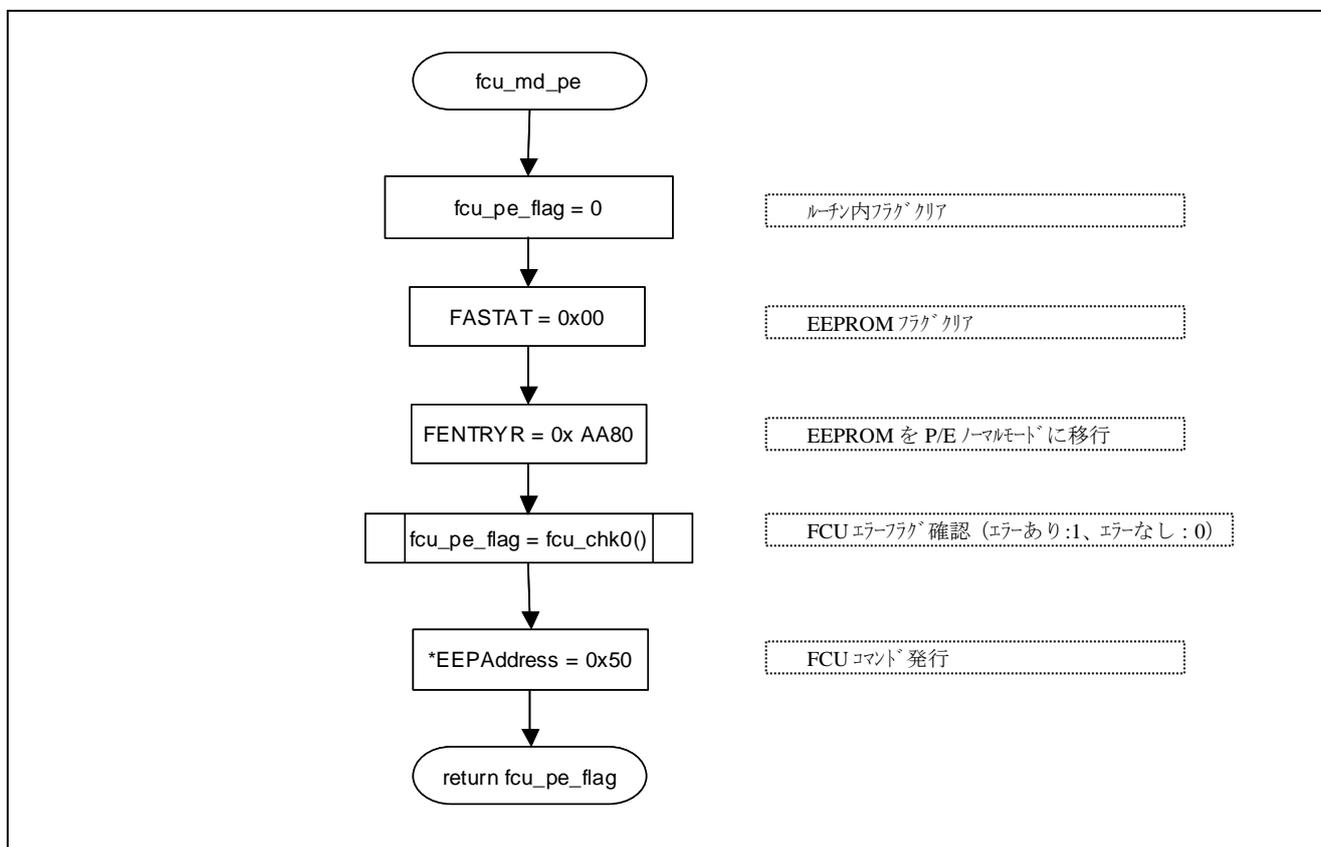


図 4.4 fcu_md_pe 関数フローチャート

4.5 fcu_eebck 関数

(1) 機能概要

ブランクチェックコマンドを発行し、対象領域の確認を行います。

(2) 引数

無し

(3) 戻り値

unsigned char fcu_eebck_flag

表 4.14 fcu_eebck 戻り値

戻り値	内容
H'00	ブランクチェックコマンド発行時にエラー無し
H'01	タイムアウトか不正なコマンド発行、EEPROM/ROM アクセスが発生しています。

(4) 使用内部レジスタ

表 4.15 フラッシュモードレジスタ (FMODR)

ビット	ビット名	設定値	備考
4	FRDMD	B'1	FCU リードモードセレクトビット 0: メモリ領域リードモード 1: レジスタリードモード

表 4.16 EEPROM ブランクチェックレジスタ (EEPBCNT)

ビット	ビット名	設定値	備考
12-3	BCADR	B'0000000000	ブランクチェックアドレス設定ビット
0	BCSIZE	B'1	ブランクチェックサイズ設定ビット 0: ブランクチェック対象エリアは 8byte 1: ブランクチェック対象エリアは 8Kbyte

(5) フローチャート

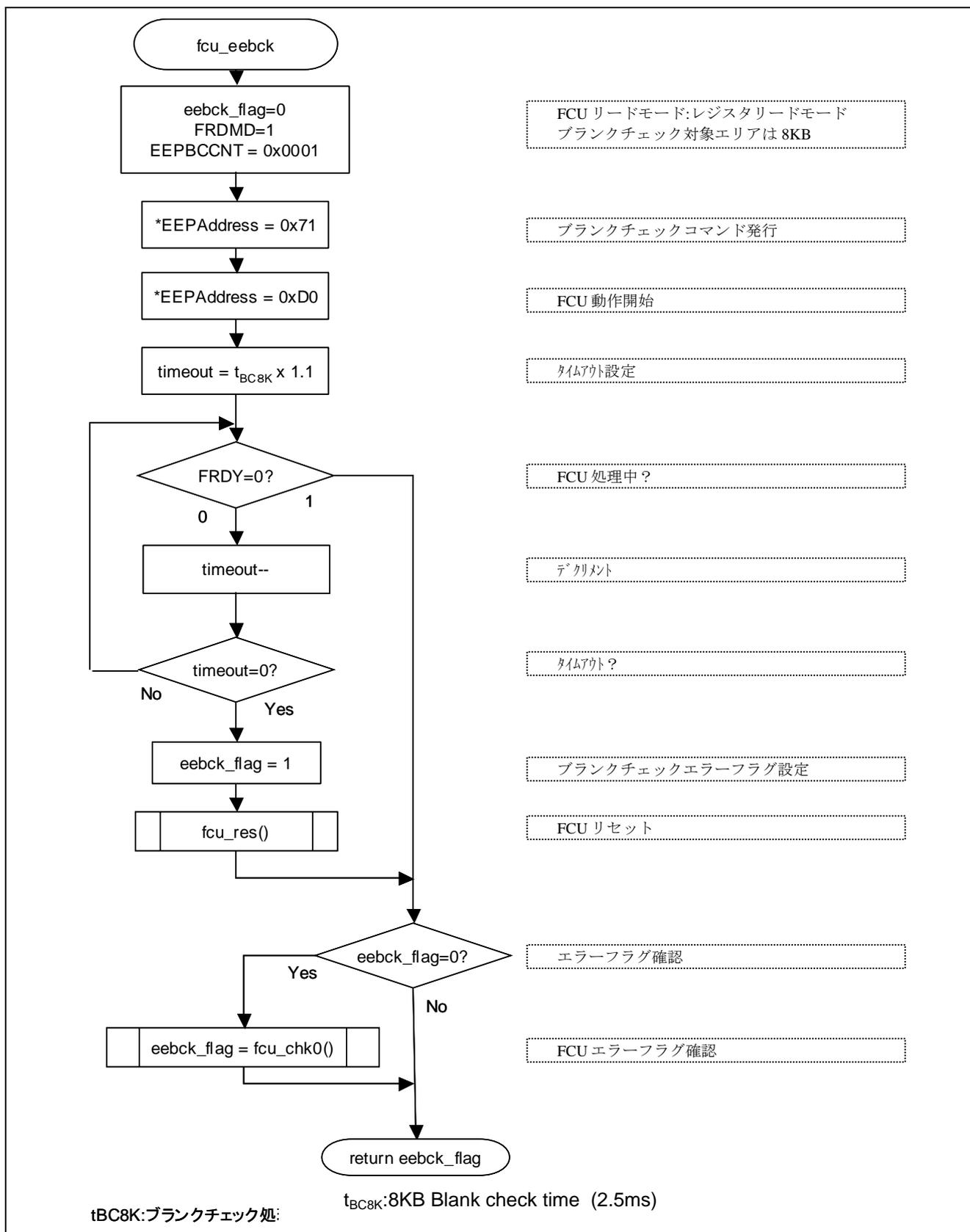


図 4.5 fcu_eebck 関数フローチャート

4.6 fcu_eerm 関数

(1) 機能概要

ブロックイレーズコマンドを発行し、結果の確認を行います。

(2) 引数

無し

(3) 戻り値

unsigned char fcu_eerm_flag

表 4.17 fcu_eem 戻り値

戻り値	内容
H'00	ブロックイレーズコマンド発行時にエラー無し
H'01	タイムアウトか不正なコマンド発行、EEPROM/ROM アクセスが発生しています。

(4) 使用内部レジスタ

無し

(5) フローチャート

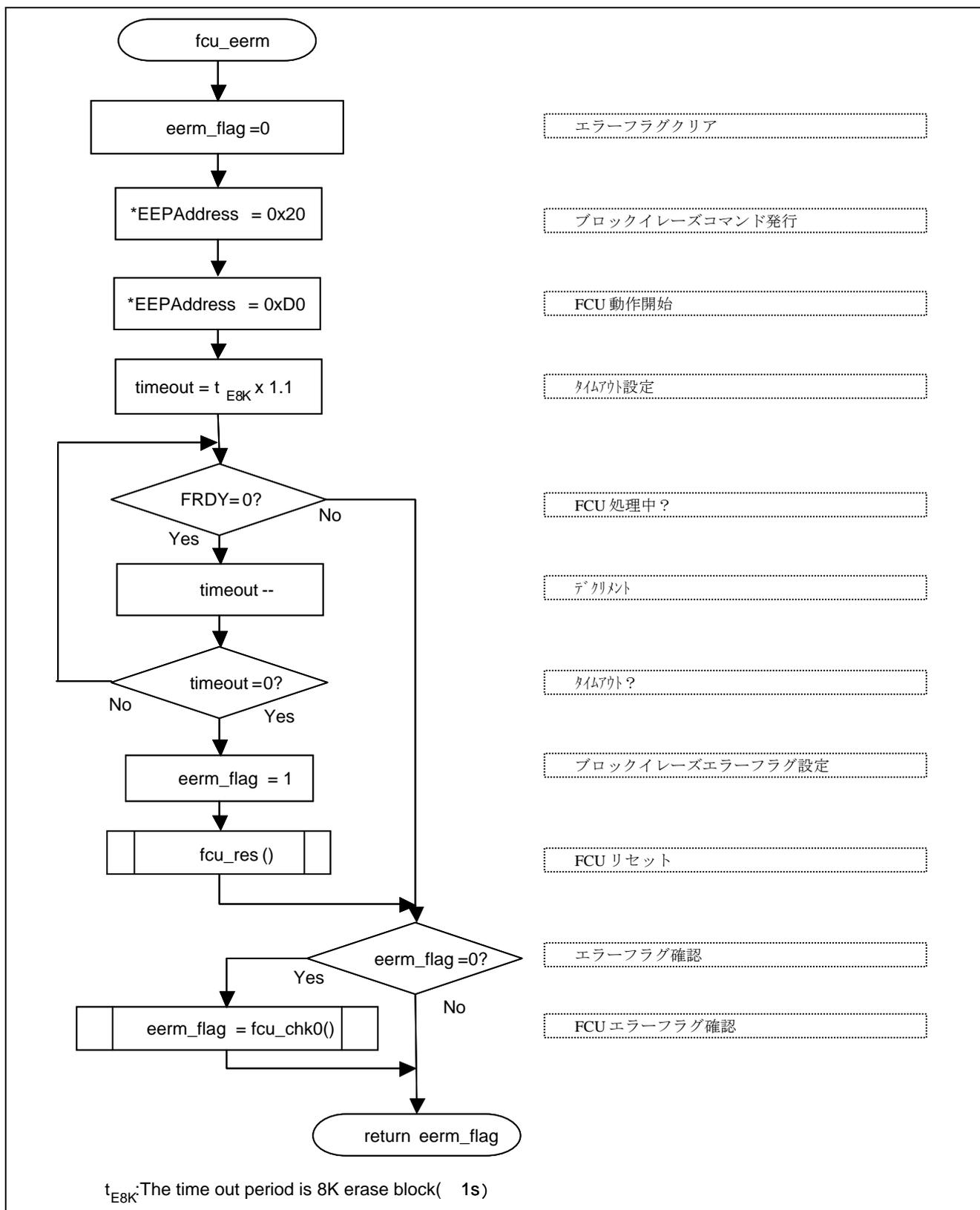


図 4.6 fcu_eem 関数フローチャート

4.7 fcu_eeprg 関数

(1) 動作概要

プログラムコマンドを発行して、書き込みを行い、結果を確認します。

(2) 引数

無し

(3) 戻り値

unsigned char fcu_eeprg_flag

表 4.18 fcu_eeprg 戻り値

戻り値	内容
H'00	プログラムコマンド発行時にエラー無し
H'01	タイムアウトか不正なコマンド発行、EEPROM/ROM アクセスが発生しています。

(4) 使用内部レジスタ

無し

(5) フローチャート

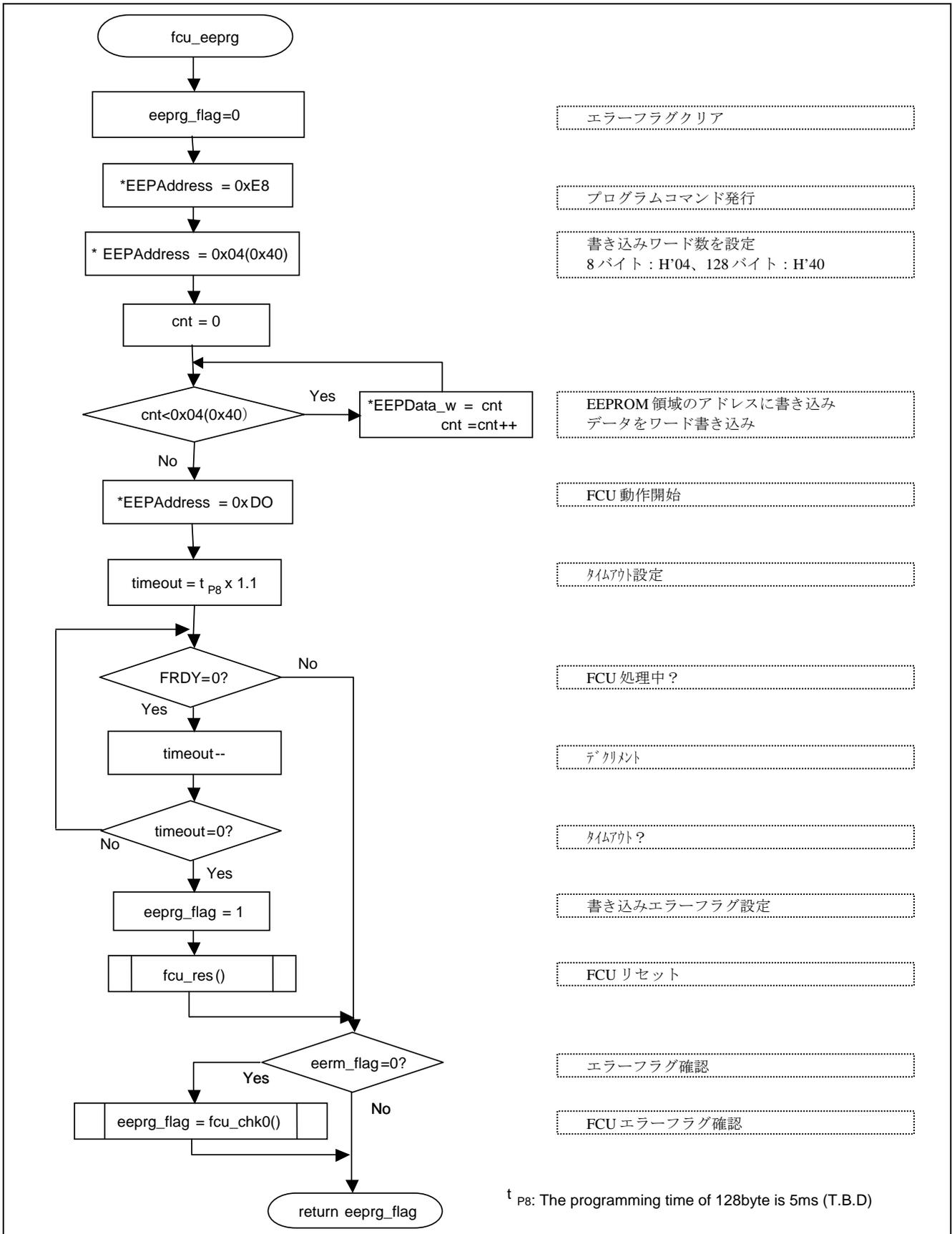


図 4.7 fcu_eeprg 関数フローチャート

4.8 fcu_res 関数

(1) 動作概要

FCU, EEPROM を初期化します。

(2) 引数

無し

(3) 戻り値

無し

(4) 使用内部レジスタ

表 4.19 フラッシュリセットレジスタ (FRESETR)

ビット	ビット名	設定値	備考
15-8	KEY	H'CC	キーコード 書き込みデータは保持されません。
0	FCRME	B'1	Flash リセットビット 0:FCU はリセットされない 1:FCU はリセットされます

(5) フローチャート

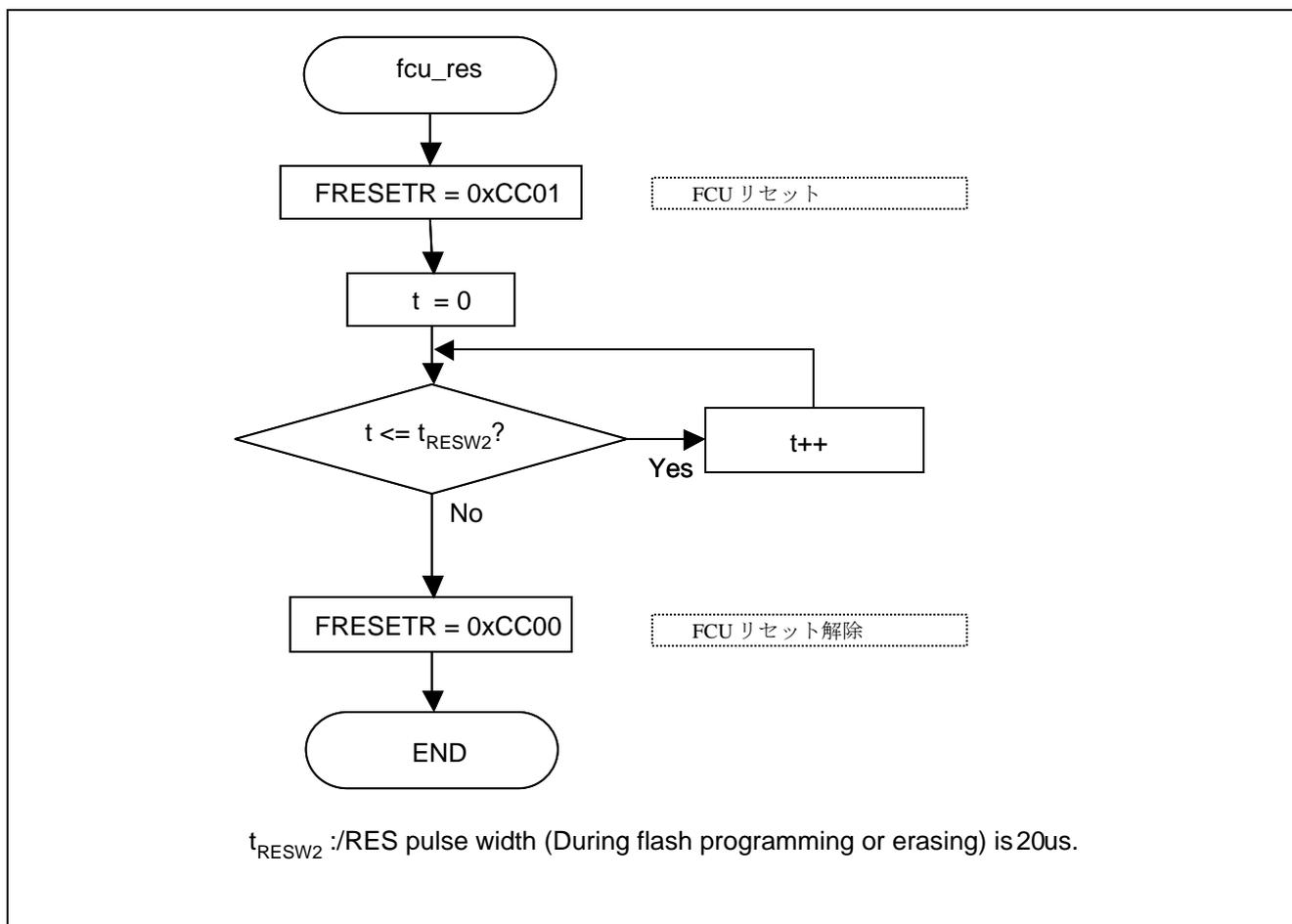


図 4.8 fcu_res 関数フローチャート

4.9 fcu_chk0 関数

(1) 動作概要

FSTART0 レジスタを読み出し、エラーを確認します。

(2) 引数

無し

(3) 戻り値

unsigned char fcu_chk0

表 4.20 fcu_chk0 戻り値

戻り値	内容
H'00	FSTART0 にエラーフラグはセットされていません。
H'01	FCU が不正なコマンドや EEPROM/ROM アクセスを検出したか、書き込み/消去中にエラーが発生しています。

(4) 使用内部レジスタ

表 4.21 フラッシュステータスレジスタ 0 (FSTATR0)

ビット	ビット名	設定値	備考
7	FRDY	-	フラッシュレディビット
6	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/EEPROM アクセスを検出していない 1:FCU は不正なコマンドや ROM/EEPROM アクセスを検出しています
5	ERSERR	-	消去エラービット 0: 消去処理は正常終了 1: 消去処理中にエラー発生
4	PRGERR	-	書き込みエラービット 0: 書き込み処理は正常終了 1: 書き込み処理中にエラー発生
3	SUSRDY	-	サスペンドレディビット
1	ERSSPD	-	消去サスペンドステータスビット
0	PRGSPD	-	書き込みサスペンドステータスビット

(5) フローチャート

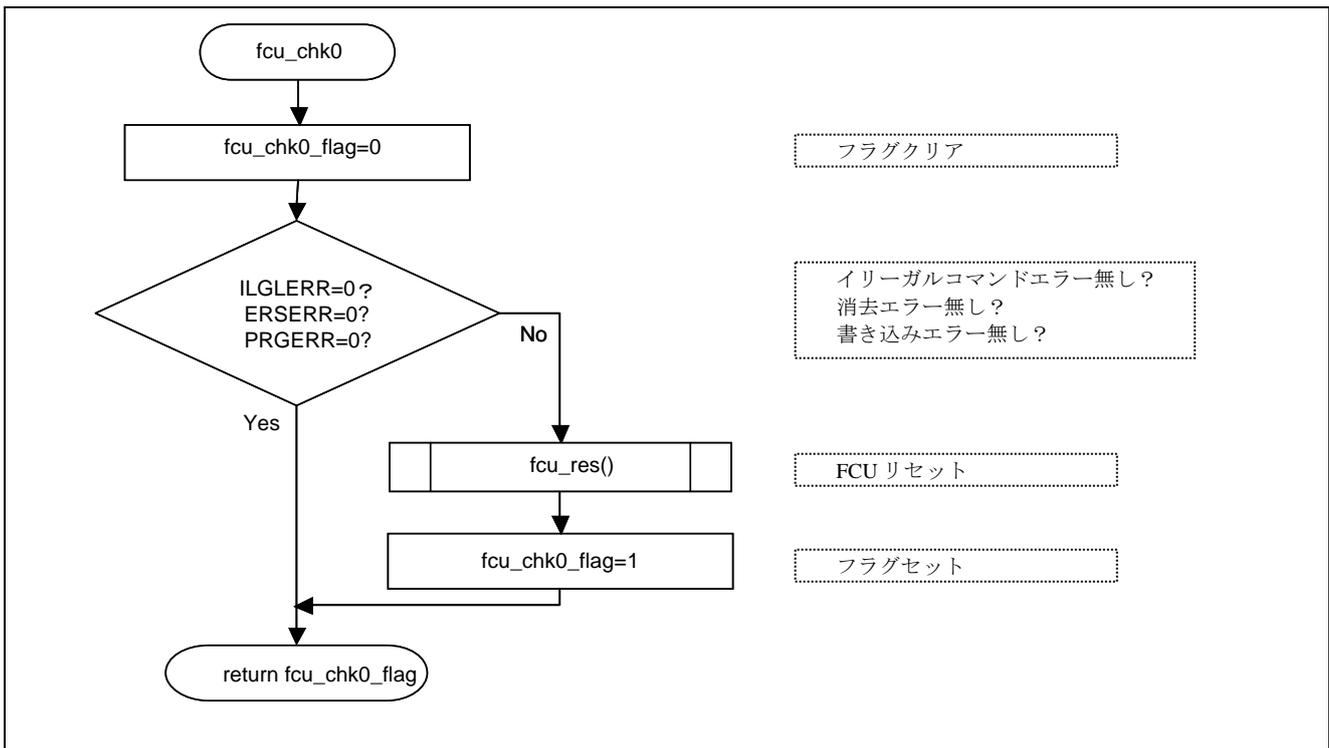


図 4.9 fcu_chk0 関数フローチャート

4.10 fcu_chk01 関数

(1) 動作概要

FSTART0,FSTART1 レジスタを読み出し、エラーを確認します。

(2) 引数

無し

(3) 戻り値

unsigned char fcu_chk01

表 4.22 fcu_chk01 戻り値

戻り値	内容
H'00	No error
H'01	FCU が不正なコマンドや EEPROM/ROM アクセスを検出したか、書き込み/消去中にエラーが発生したか、FCU 処理中に ECC エラーが発生しています。

(4) 使用内部レジスタ

表 4.23 フラッシュステータスレジスタ 0(FSTATR0)

ビット	ビット名	設定値	備考
7	FRDY	-	フラッシュレディビット
6	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/EEPROM アクセスを検出していない 1:FCU は不正なコマンドや ROM/EEPROM アクセスを検出しています
5	ERSERR	-	消去エラービット 0: 消去処理は正常終了 1: 消去処理中にエラー発生
4	PRGERR	-	書き込みエラービット 0: 書き込み処理は正常終了 1: 書き込み処理中にエラー発生
3	SUSRDY	-	サスペンドレディビット
1	ERSSPD	-	消去サスペンドステータスビット
0	PRGSPD	-	書き込みサスペンドステータスビット

表 4.24 フラッシュステータスレジスタ 1(FSTATR1)

ビット	ビット名	設定値	備考
7	FCUERR	-	FCU エラービット 0: FCU 処理中にエラー未発生 1: FCU 処理中にエラーが発生
4	FLOCKST	-	ロックビットステータスビット
1	FRDTCT	-	FCU RAM2 ビット誤り検出モニタビット 0: 2 ビット誤りを検出していない 1: 2 ビット誤りを検出した
0	FRCRCT	-	FCU RAM1 ビット誤り訂正モニタビット 0: 1 ビット誤りを訂正していない 1: 1 ビット誤りを訂正した

(5) フローチャート

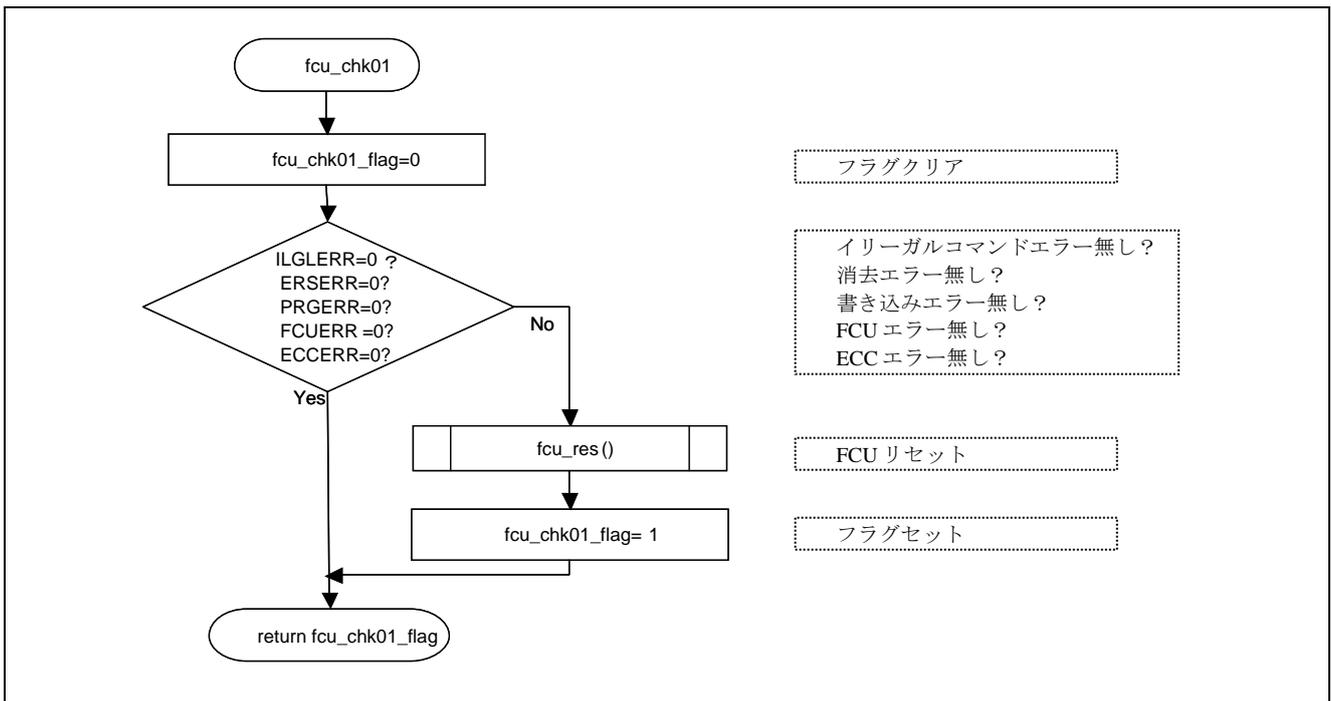


図 4.10 fcu_chk01 関数フローチャート

5. サンプルプログラム

```

1  /*****
2  * DISCLAIMER
3  * This software is supplied by Renesas Electronics Corporation and is only
4  * intended for use with Renesas products. No other uses are authorized. This
5  * software is owned by Renesas Electronics Corporation and is protected under
6  * all applicable laws, including copyright laws.
7  * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
8  * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
9  * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
10 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
11 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
12 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
13 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
14 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
15 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
16 * Renesas reserves the right, without notice, to make changes to this software
17 * and to discontinue the availability of this software. By using this software,
18 * you agree to the additional terms and conditions found by accessing the
19 * following link:
20 * http://www.renesas.com/disclaimer *
21 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
22 *****/
23 /*****
24 * File Name      : main.c
25 * Version        : 1.00
26 * Device(s)     : SH72544R
27 * Tool-Chain    : High-performance Embedded Workshop (Ver.4.09.00).
28 * H/W Platform  : SH725xEVB04
29 * Description    : This is the main tutorial code.
30 * Operation     : 1. Compile and download the sample code. Click 'Reset Go'
31 *                to start the software.
32 *****/
33 * History : DD.MM.YYYY Version Description
34 *         : 26.12.2011 1.00 First Release
35 *****/
36 /*****
37 Includes <System Includes> , "Project Includes"
38 *****/
39 #include <machine.h> /* ライブラリ関数用ヘッダファイル */
40 #include "iodefine.h" /* 周辺レジスタ定義ヘッダファイル */
41
42 /*****
43 Macro definitions
44 *****/
45 #define FSTATR0_ILGLERR (0x40) /* イレガールコマンドエラー */
46 #define FSTATR0_ERSERR (0x20) /* 消去エラー */
47 #define FSTATR0_PRGERR (0x10) /* 書き込みエラー */
48 #define FSTATR1_FCUERR (0x80) /* FCUエラー */
49 #define FSTATR1_ECCERR (0x01) /* ECCエラー */
50

```

```
51  /*****
52  Private global variables and functions
53  *****/
54  unsigned int    *sen, *rec;
55  unsigned long   INT_FCU_FIRM,INT_FCU_SIZE,INT_FCU_RAM,Timeout;
56  unsigned char   flag,fcu_end,cnt;
57  unsigned char  *EEPAddress; /* EEPROM 領域アドレス */
58  unsigned short *EEPData_w; /* 書き込み先頭アドレス */
59
60  void main(void); /* メインルーチン */
61  void init_eeeprom(void); /* EEPROM の初期化 */
62  void copy_firm(void); /* FCU ファームの転送 */
63  char fcu_md_pe(void); /* FCU を P/E ノーマルモードへ移行 */
64  char fcu_eebck(void); /* ブランクチェックコマンド発行 */
65  char fcu_eerm(void); /* ブロックイレーズコマンド発行 */
66  char fcu_eeprg(void); /* データFlash 書き込み */
67  void fcu_res(void); /* FCU,ROM,EEPROM のリセット */
68  char fcu_chk0(void); /* FSTART0 のエラーフラグチェック */
69  char fcu_chk01(void); /* FSTART0 と FSTART1 のエラーフラグチェック */
70
71  /*****
72  * Function Name: main
73  * Description   : The main loop
74  * Arguments     : none
75  * Return Value  : none
76  *****/
77  void main(void)
78  {
79
80  _INITSCT();
81
82  EEPAddress = (unsigned char *) (0x80102000); /* コマンド入力先アドレス */
83  EEPData_w  = (unsigned short *) (0x80102080); /* ワード書き込み先アドレス */
84  fcu_end = 1;
85
86  init_eeeprom(); /* EEPROM 初期化 */
87
88  while (fcu_end != 0)
89  {
90  flag = fcu_chk01(); /* FCU ステータス確認 */
91  if (flag == 0)
92  {
93  copy_firm(); /* ファームウェア転送 */
94  flag = fcu_md_pe(); /* P/E ノーマルモードへ移行 */
95  }
96  if (flag == 0)
97  {
98  flag = fcu_eebck(); /* ブランクチェックコマンド発行 */
99  if (EEPROM.EEPBCSTAT.BIT.BCST == 1) /* BCST に 1 がセットされているならば */
100  {
101  flag = fcu_eerm(); /* ブロックイレーズコマンド発行 */
102  }
```

```

103 }
104 if (flag == 0)
105 {
106 fcu_end = fcu_eeprg(); /* データFlash 書き込み */
107 }
108 }
109 ROM.FENTRYR.WORD = 0xAA00; /* ROM/EEPROM リードモードへ移行 */
110 while(1);
111 }
112
113 /*****
114 * Function Name: init_eeprgm
115 * Description   : EEPROM 初期設定ルーチン
116 * Arguments    : none
117 * Return Value : none
118 *****/
119 void init_eeprgm(void)
120 {
121 /* Configure FAEINT
122 b7 ROMAEIE = 0 ROMAE=1 で FIFE 割り込み要求を発生しない
123 b6,5 reserved
124 b4 CMDLKIE = 0 CMDLK=1 で FIFE 割り込み要求を発生しない
125 b3 EEPAEIE = 0 EEPROM アクセス違反割り込みイネーブル
126 b2 EEPIFEIE = 0 EEPROM 命令フェッチ違反割り込みイネーブル
127 b1 EEPPEIE = 0 EEPROM リードプロテクト違反割り込みイネーブル
128 b0 EEPWEIE = 0 EEPROM 書き込み/消去プロテクト違反割り込みイネーブル*/
129 ROM.FAEINT.BYTE = 0x00; /* フラッシュインタフェース割り込みを禁止 */
130
131 /* Configure FPROTR
132 b15-8 FPKEY = H'55 キーコード
133 b7-1 reserved
134 b0 FPROTCN = 0 ロックビットによるプロテクト有効*/
135 ROM.FPROTR.WORD = 0x5500; /* ROM のロックビットプロテクトを有効に設定 */
136
137 EEPROM.EEPRE0.WORD = 0x2D02; /* DB01 ブロックの読み出しを許可 */
138 EEPROM.EEPRE1.WORD = 0xD200; /* DB15-08 ブロックの読み出しを禁止 */
139 EEPROM.EEPWE0.WORD = 0x1E02; /* DB01 ブロックへの書き込みを許可 */
140 EEPROM.EEPWE1.WORD = 0xE100; /* DB15-08 ブロックの書き込みを禁止 */
141 }
142
143 /*****
144 * Function Name: copy_firm
145 * Description   : FCU ファームの転送ルーチン
146 * Arguments    : none
147 * Return Value : none
148 *****/
149 void copy_firm(void)
150 {
151 INT_FCU_FIRM = 0x00402000; /*FCU ファームの転送元アドレス */
152 INT_FCU_RAM = 0x80FF8000; /*FCU ファームの転送先アドレス */
153 INT_FCU_SIZE = 0x00001FFF; /*FCU ファームの転送回数 */
154 EEPROM.EEPMAT.WORD = 0xB300; /*EEPROM のマットをデータマットに設定 */

```

```

155
156 /* Configure FCURAME
157 b15-8 KEY = H'C4 キーコード
158 b7-1 reserved
159 b0 FCRME = 1 FCU RAM へのアクセス許可*/
160 ROM.FCURAME.WORD = 0xC401; /*FCU RAM へのアクセスを許可 */
161
162 if(ROM.FENTRYR.WORD != 0x0000) /*FCU のモードがリードモードに設定されているか確認する */
163 {
164 /* Configure FENTRYR
165 b15-8 FPKEY = H'AA キーコード
166 b7 FENTRYD = 0 EEPROM P/E モードエントリビット
167 b6 reserved
168 b5 FENTRY5 = 0 ROM 0.25MB はリードモード
169 b4 FENTRY4 = 0 ROM 0.25MB はリードモード
170 b3 FENTRY3 = 0 ROM 0.25MB はリードモード
171 b2 FENTRY2 = 0 ROM 1MB はリードモード
172 b1 FENTRY1 = 0 ROM 1MB はリードモード
173 b0 FENTRY0 = 0 ROM 1MB はリードモード*/
174 ROM.FENTRYR.WORD = 0xAA00; /*他のモードならば ROM/EEPROM リードモードに移行する*/
175 }
176
177 sen = (unsigned int *)INT_FCU_FIRM; /* FCU ファームの転送元の値を変数 sen に入力*/
178 rec = (unsigned int *)INT_FCU_RAM; /* FCU ファームの転送先の値を変数 rec に入力*/
179
180 /* ファームウェア転送 */
181 while(sen <= (unsigned int *)(INT_FCU_FIRM + INT_FCU_SIZE))
182 {
183 *rec++ = *sen++;
184 }
185 }
186
187 /*****
188 * Function Name: fcu_md_pe
189 * Description : FCU を P/E ノーマルモードへ移行ルチン
190 * Arguments : none
191 * Return Value : 0:OK,1:NG
192 *****/
193 char fcu_md_pe(void)
194 {
195 unsigned char fcu_pe_flag;
196
197 fcu_pe_flag = 0;
198
199 ROM.FASTAT.BYTE = 0x00; /* コマンドロックビット以外をクリア */
200
201 /* Configure FENTRYR
202 b15-8 FPKEY = H'AA キーコード
203 b7 FENTRYD = 1 EEPROM P/E モードエントリビット
204 b6 reserved
205 b5 FENTRY5 = 0 ROM 0.25MB はリードモード
206 b4 FENTRY4 = 0 ROM 0.25MB はリードモード

```

```
207 b3 FENTRY3 = 0 ROM 0.25MB はリードモード
208 b2 FENTRY2 = 0 ROM 1MB はリードモード
209 b1 FENTRY1 = 0 ROM 1MB はリードモード
210 b0 FENTRY0 = 0 ROM 1MB はリードモード*/
211 ROM.FENTRYR.WORD = 0xAA80; /* FCU のモードを EEPROM P/E ノーマルモード に移行*/
212
213 fcu_pe_flag = fcu_chk0(); /* FCU ステータスチェック */
214
215 *EEPAddress = 0x50; /* EEPROM 領域に FCU コマンド 0x50 をバイト書き込み*/
216
217 return fcu_pe_flag;
218 }
219
220 /*****
221 * Function Name: fcu_eebck
222 * Description   : ブランクチェックコマンド発行ルーチン
223 * Arguments     : none
224 * Return Value  : 0:OK,1:NG
225 *****/
226 char fcu_eebck()
227 {
228     unsigned char eebck_flag;
229     eebck_flag = 0;
230     ROM.FMODR.BIT.FRDMD = 1; /* FCU リードモード:レジスタリードモード */
231     EEPROM.EEPBCNT.WORD = 0x0001; /* ブランクチェックサイズ 8KB */
232
233     *EEPAddress = 0x71; /* ブランクチェックコマンド発行 */
234     *EEPAddress = 0xD0; /* FCU 動作開始 */
235
236     Timeout = 0x2800; /* 待機時間を約 2.5ms に設定 (TBD) */
237     while(!(ROM.FSTATR0.BIT.FRDY))
238     { /* FRDY ビットに 1 がセットされるまで待機 */
239         if(Timeout == 0)
240         { /* 待機中に一定期間経過時 */
241             eebck_flag = 1; /* エラー設定 */
242             fcu_res();
243             break;
244         }
245         Timeout--; /* Timeout の値をデクリメント */
246     }
247     if (eebck_flag == 0){
248         eebck_flag = fcu_chk0(); /* 書き込み後のエラー確認 */
249     }
250     return eebck_flag;
251 }
252
253 /*****
254 * Function Name: fcu_eerm
255 * Description   : データ Flash のブロック消去ルーチン
256 * Arguments     : none
257 * Return Value  : 0:OK,1:NG
258 *****/
```

```

259 char fcu_eerm()
260 {
261 unsigned char eerm_flag;
262 eerm_flag = 0;
263 *EEPAddress = 0x20; /* EEPROM 領域に FCU コマンド 0x20 をバイト書き込み */
264 *EEPAddress = 0xD0; /* 消去/書き込み領域に FCU コマンド 0xD0 をバイト書き込み */
265
266 Timeout = 0x000F4240; /* 待機時間を約 250ms に設定 (TBD) */
267 while(!(ROM.FSTATR0.BIT.FRDY)) /* FRDY ビットに 1 がセットされるまで待機 */
268 {
269 if(Timeout == 0)
270 { /* 待機中に一定期間経過時 */
271 eerm_flag = 1; /* エラー設定 */
272 fcu_res();
273 break;
274 }
275 Timeout--; /* Timeout の値をデクリメント */
276 }
277 if (eerm_flag == 0)
278 {
279 eerm_flag = fcu_chk0(); /* 消去後のエラー確認 */
280 }
281 return eerm_flag;
282 }
283
284 /*****
285 * Function Name: fcu_eeprg
286 * Description : データ Flash 書き込み(8/128 バイト)ルーチン
287 * Arguments : none
288 * Return Value : 0:OK,1:NG
289 *****/
290 char fcu_eeprg()
291 {
292
293 unsigned char eeprg_flag;
294 eeprg_flag = 0;
295 *EEPAddress = 0xE8; /*EEPROM 領域に FCU コマンド 0xE8 をバイト書き込み*/
296 // *EEPAddress = 0x04; /*8 バイト EEPROM 領域に書き込みデータ量をバイト書き込み */
297 *EEPAddress = 0x40; /*128 バイト EEPROM 領域に書き込みデータ量をバイト書き込み */
298
299 // for (cnt = 0;cnt < 0x04; cnt++ )
300 for (cnt = 0;cnt < 0x40; cnt++ )
301 {
302 *EEPData_w = cnt;
303 }
304 *EEPAddress = 0xD0; /* EEPROM 領域に FCU コマンド 0xD0 をバイト書き込み*/
305 Timeout = 0x4E20; /* 待機時間を約 5ms に設定 (TBD) */
306 while(!(ROM.FSTATR0.BIT.FRDY)) /* FRDY ビットに 1 がセットされるまで待機 */
307 {
308 if(Timeout == 0)
309 { /* 待機中に一定期間経過時 */
310 eeprg_flag = 1; /* エラー設定 */

```

```

311 fcu_res();
312 break;
313 }
314 Timeout--; /* Timeout の値をデクリメント */
315 }
316 if (eeprg_flag == 0)
317 {
318 eeprg_flag = fcu_chk0(); /* FCU 書き込みエラー確認 */
319 }
320 return eeprg_flag;
321 }
322
323 /*****
324 * Function Name: fcu_res
325 * Description   : FCU,EEPROM のリセットルーチン
326 * Arguments     : none
327 * Return Value  : none
328 *****/
329 void fcu_res(void)
330 {
331
332 unsigned long t;
333 t = 0;
334
335 /* Configure FRESETR
336 b15-8 RXPR1[15:0] = H'CC キーコード
337 b7-1 reserved
338 b0 FRESET = H'1 FCU はリセットされる*/
339 ROM.FRESETR.WORD = 0xCC01; /* FCU,ROM,EEPROM を初期化 */
340 for(t = 0; t <= 0x320; t++); /* 20μS ウェイトになるよう設定*/
341
342 /* Configure FRESETR
343 b15-8 RXPR1[15:0] = H'CC キーコード
344 b7-1 reserved
345 b0 FRESET = 0 FCU はリセットされない*/
346 ROM.FRESETR.WORD = 0xCC00;
347
348 return;
349 }
350
351 /*****
352 * Function Name: fcu_chk0
353 * Description   : FCU コマンド実施後 status 確認ルーチン
354 * Arguments     : none
355 * Return Value  : 0:OK,1:NG
356 *****/
357 char fcu_chk0(void)
358 {
359 unsigned char fcu_chk0_flag;
360 fcu_chk0_flag = 0;
361 /* いずれかのエラーフラグが発生しているなら */
362 if(ROM.FSTATR0.BYTE & (FSTATR0_ILGLERR | FSTATR0_PRGERR | FSTATR0_ERSERR))

```

```
363  {
364  fcu_res();
365  fcu_chk0_flag = 1; /* エラー設定 */
366  }
367  return fcu_chk0_flag;
368  }
369
370  /*****
371  * Function Name: fcu_chk01
372  * Description   : FCU 初期化 check ルーチン
373  * Arguments     : none
374  * Return Value  : 0:OK,1:NG
375  *****/
376  char fcu_chk01(void)
377  {
378
379  unsigned char fcu_chk01_flag;
380  fcu_chk01_flag = 0;
381  /* Flash ステータスをチェックし、異常なら FCU と Flash リセット */
382  if((ROM.FSTATR0.BYTE & (FSTATR0_ILGLERR | FSTATR0_ERSERR | FSTATR0_PRGERR))
383  || (ROM.FSTATR1.BYTE & (FSTATR1_FCUERR | FSTATR1_ECCERR)))
384  {
385  fcu_res();
386  fcu_chk01_flag = 1; /* エラー設定 */
387  }
388  return fcu_chk01_flag;
389  }
```

6. SH7253 グループでご使用の場合の変更点

本文 1.~5.及びサンプルソフトは SH7254R グループ用に作成されたものです。SH7253 グループでご使用される場合は SH7254R グループのサンプルソフトを下記に従い変更してください。

変更内容

(1)製品ヘッダファイル iodefine.h を SH7253 グループ用のものに差替えてください

(2)SH7253 グループの EEPROM のデータマットは 8KB ブロック x 4(合計 32kB)で、SH7254R グループと容量が異なります。(SH7254R グループは、8KB ブロック x16(合計 128kB)です)

SH7253 グループには EEPROM 読み出し許可レジスタ 1、EEPROM 書き込み/消去レジスタ 1 はありません。

具体的には

5.サンプルプログラムリストの 138 行目と 140 行目を削除してください。

138 行目 : EEPROM.EEPRE1.WORD=0xD200

140 行目 : EEPROM.EEPWE1.WORD=0xE100

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/>
- お問い合わせ先
<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

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、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

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、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違くと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

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用途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。



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/inquiry>