



AN-1471 アプリケーション・ノート

ADuCM4050 のフラッシュ・メモリを利用した EEPROM エミュレーション

著者: Pranit Jadhav, Rafael Lajara

はじめに

不揮発性のデータ・ストレージは、多くの組み込みシステムに欠くことのできないものです。ブートアップ構成、キャリブレーション定数、ネットワーク関連情報などのデータは、通常、EEPROM (Electrically Erasable Programmable Read Only Memory) デバイスに保存されます。これらのデータの保存に EEPROM を使用する利点は、EEPROM 上の他のロケーションの内容に影響を与えることなく、1 バイト単位でデータの書き換えや更新を行えることです。

ADuCM4050 は、フラッシュ・メモリを内蔵した超低消費電力のマイクロコントローラ・ユニット (MCU) です。ADuCM4050 には、72 ビット幅のデータ・バスを備えた 512kB の組み込みフラッシュ・メモリがあり、このデータ・バスにより 1 回のアクセスで、2 個の 32 ビット・ワードのデータと、それに対応する 8 ビットのエラー訂正コード (ECC) バイトを送ることができます。この ECC は ADuCM4050 のユーザ・スペース上でデフォルトで有効化されており、これによってフラッシュの初期化機能が

期待どおりに動作します。ECC チェックは、ADuCM4050 のフラッシュ・メモリの全ユーザ・スペースに対して有効化されています。ECC エラーが読み出し動作でレポートされると、ECC エンジンが自動的に 1 ビット・エラーの場合は訂正し、2 ビット・エラーの場合は検出レポートのみを行います。フラッシュで読み出しが行われると、適切なフラグが ADuCM4050 のステータス・レジスタにセットされます。割込みが生成されると、割込みの原因となった ECC エラーのソース・アドレスが FLCC0_ECC_ADDR レジスタで読み出し可能となり、割込みサービス・ルーチン (ISR) によって読み出しが行われます。

内蔵フラッシュ・メモリで EEPROM をエミュレートすれば、設計から EEPROM を除外できるので、部品表 (BOM) コストを削減でき、ユーザが書き込む必要のあるコードの複雑さが軽減されます。

目次

| | | | |
|-------------|---|------------|----|
| はじめに..... | 1 | Flash..... | 7 |
| 改訂履歴..... | 2 | 制約事項..... | 9 |
| 背景..... | 3 | まとめ..... | 10 |
| 動作原理..... | 4 | | |
| EEPROM..... | 4 | | |

改訂履歴

6/2018—Revision 0: Initial Version

背景

通常、フラッシュ・メモリはページの配列として構成されます。ADuCM4050の1ページには2kBの容量があります。ページの内容は、データを書き込む前に消去する必要があります。消去動作はページ全体に対して行われますが、読出しや書込みは、アドレス指定可能な1つのロケーション（バイトまたはワード）に対して行うことができます。

アドレス指定可能な1つのロケーションを対象とする読出しと書込みには、次のような課題があります。

- 通常、フラッシュ・メモリは書き込み動作や消去動作をブロックレベルで行うため、1バイト幅での読出し動作や書き込み動作は困難である。
- フラッシュ・メモリの消去はページ全体に対して行われるため、任意のロケーションにあるデータを消去または更新する一方で、他のロケーションにあるデータを維持することが困難である。

このアプリケーション・ノートでは、図1に示すように、ADuCM4050 デバイスと内蔵フラッシュ・メモリを使ってEEPROM をエミュレートするソフトウェアについて説明します。

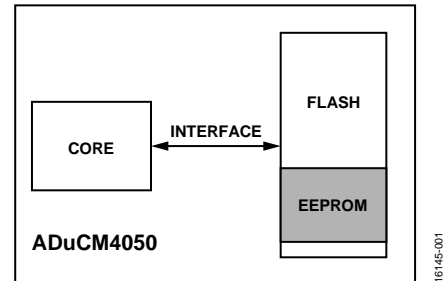


図 1. ADuCM4050 内蔵フラッシュと EEPROM システムの概要

EEPROM のエミュレーション・コードのブロック図を図2に示します。上位レベルの機能で EEPROM エミュレーションを実行し、下位レベルの機能ではフラッシュの管理とアドレス変換を行います。

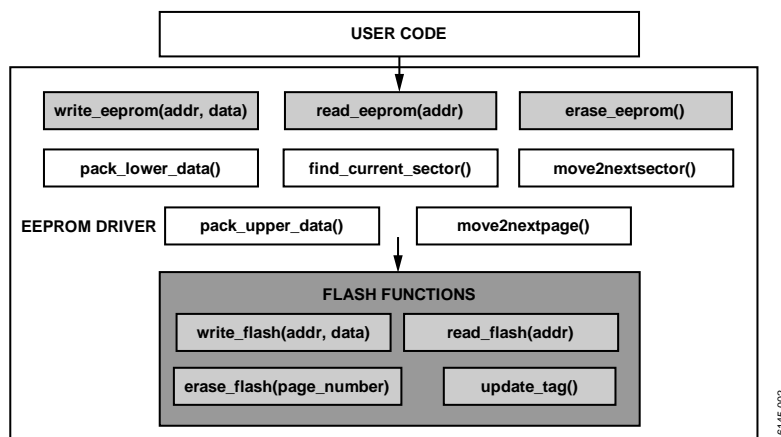


図 2. ADuCM4050 フラッシュ EEPROM エミュレーション・ソフトウェアの構造

動作原理

EEPROM をエミュレートするには、フラッシュ・メモリの一部をエミュレーション専用割り当てする必要があります。ほとんどの EEPROM では、1 回の書き込みコマンドで更新できるデータは 1 バイトだけです。フラッシュ・メモリ・デバイスには複数バイトを書き込む機能が備わっており、それに応じてデータを更新できますが、それは 2 回の書き込み動作の間に消去シーケンスを設定した場合に限られます。1 バイトの書き込みと読出しが可能な EEPROM をフラッシュ・メモリでエミュレートするには、EEPROM 動作と同様の読出し、変更、書き込みシーケンスを設定する必要があります。

このアプリケーション・ノートに示す手順では、フラッシュ・ページを 2 ページ使用しますが、これらのページを 3 ページ以上に拡張して、セクタ・タグで構成される複数のセクタに分割することができます。このセクタ・タグは、現在処理中のセクタと、そのセクタに書き込まれるデータのバイト数に関する情報を提供します。どのセクタも最後のロケーションがセクタ・タグ用に予約されており、そのサイズは、フラッシュ・メモリのデータ・バスのサイズと同じです。フラッシュ・ページ内のセクタ・サイズとセクタ数は、エミュレートされる EEPROM のサイズに依存します。

EEPROM

EEPROM の書き込みおよび読出し機能には、EEPROM のデータやアドレス情報など、アプリケーション・コード入力の処理が含まれます。EEPROM の API (Application Programming Interface) は、フラッシュ・インターフェースの条件に従ってデータやアドレス情報を処理し、提供します。

EEPROM の初期化

EEPROM 動作を初期化することで、EEPROM のサイズとエミュレートする EEPROM のデータのワード長が設定されます。このアプリケーション・ソフトウェアでは、EEPROM エミュレーションがデータの完全性を維持できるように 2 ページ分のフラッシュ・ページが予約されているため、エミュレートする EEPROM の最大サイズは 2kB (1 フラッシュ・ページ) です。同じアプリケーション・ソフトウェアでフラッシュ・ページの予約が 2 ページから 4 ページに拡張されている場合は、この制限も拡張できます。ユーザが設定できるワード長は、8 ビットから最大 64 ビットまでです。init_eeprom(uint16_t eeprom_size, uint8_t word_length)関数の詳細については、表 1 を参照してください。

表 1. EEPROM 初期化関数の説明

(init_eeprom(uint16_t eeprom_size, uint8_t word_length))¹

| Parameter | Description | Return Value |
|-------------|--|---|
| eeprom_size | エミュレートする EEPROM のサイズ。 | エラーなし。書き込みは正常に実行されました。 |
| word_length | エミュレートする EEPROM に書き込まれるデータ・ワード長の有効な値は、8、16、32、64 です。 | エラー。指定したアドレスは使用可能な EEPROM メモリ空間の外にあります。 |

¹この関数は、EEPROM にデータを書き込みます。

EEPROM への書き込み

EEPROM の書き込み動作のフローチャートを図 3 に示します。
EEPROM の書き込み動作手順は以下のとおりです。

1. find_current_sector()関数を呼び出して、現在のセクタを検索します。この検索は、セクタ・タグとこれに対応するセクタ・タグ値に基づきます。戻り値は現在のセクタ開始アドレスで、これはフラッシュ・メモリの物理的な場所です。
2. 現在のセクタ開始アドレスに基づいて、EEPROM アドレスをフラッシュ・アドレスに変換します。ADuCM4050 のフラッシュ・メモリのデータ・バスは 64 ビット幅で、エミュレートする EEPROM のデータ・バスは 8 ビット幅なので、ソフトウェアが EEPROM アドレスから必要シフト数を決定します。
3. 取得したフラッシュ・アドレスのデータを読み出します。データが 0xFF に等しい場合、書き込むデータはまず 2 つの 32 ビット・データ (pack_lower_data() と pack_upper_data()) にまとめられ、64 ビット幅のデータがフラッシュ・メモリに書き込むために用意されます。
4. write_flash()関数を呼び出し、フラッシュ・コントローラに対して書き込みコマンドを実行します。この関数の入力パラメータは、フラッシュ・メモリ・アドレスとパッケージ化された 64 ビット幅のデータです。
5. フラッシュ・メモリへの書き込み動作が正常に終了した後、update_tag()関数を呼び出して、現在のセクタのセクタ・タグを更新します。

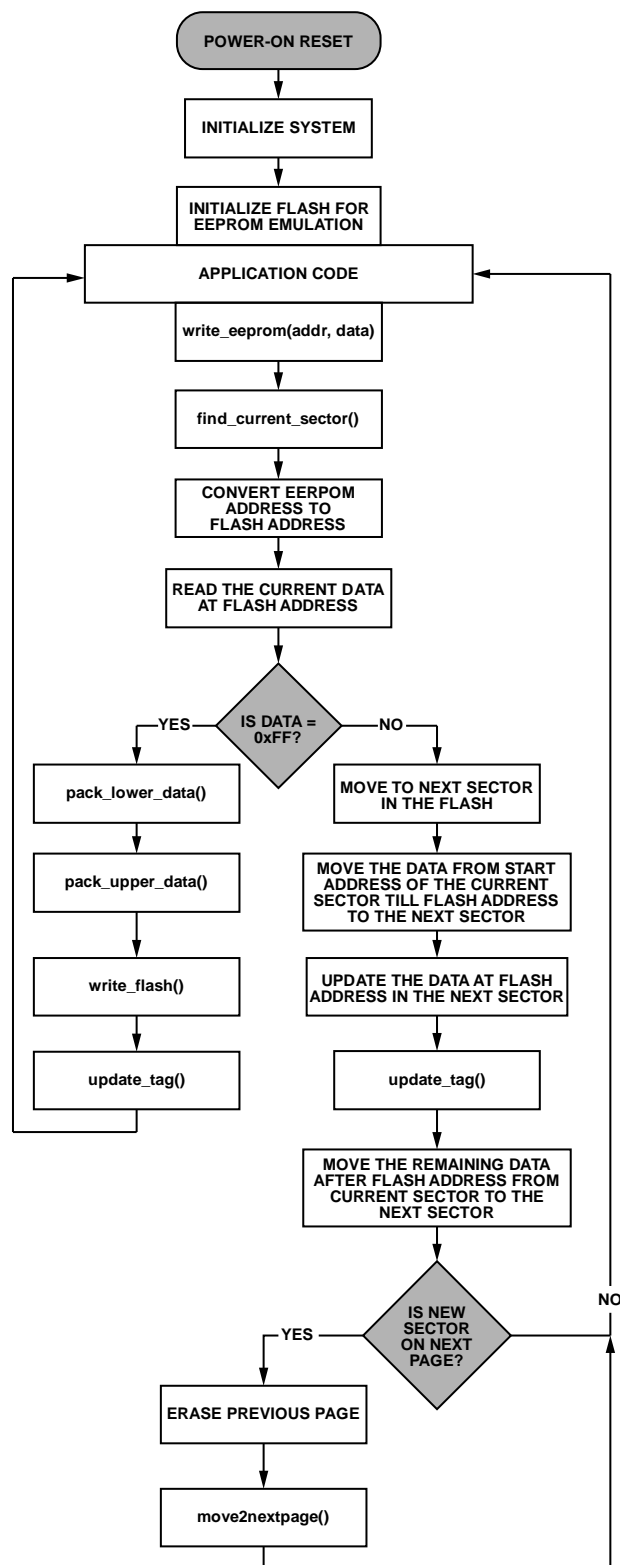


図 3. EEPROM 書き込み動作のフローチャート

16146-003

得られたフラッシュ・アドレスにデータが既に存在する場合、データ読取り関数は 0xFF を返しません。この場合、得られたフラッシュ・アドレスの前後にあるデータは、move2nextsector()関数を呼び出すことによって、次のセクタ、つまり隣接セクタに移動します。EEPROM データは、LSB および MSB データ・パケットに変換され、次のセクタの新しいフラッシュ・アドレスに書き込まれます。既に書き込み済みの EEPROM のロケーションに書き込み命令が送出されるたびに、データは、変更されたデータを格納するロケーションを含む次のセクタに移されます。

新しいセクタが次ページにある場合は、データ移動後に前ページ上で erase_flash (page_number) を呼び出すことによって、フラッシュ・ページ消去コマンドが送出されます。全てのアドレス・レジスタが、move2nextpage()関数によって更新されます。

write_eeprom(uint16_t addr_eeprom, uint8_t data_eeprom)関数の詳細については、表 2 を参照してください。

表 2. EEPROM 書き込み関数の説明

(write_eeprom(uint16_t addr_eeprom, uint8_t data_eeprom))¹

| Parameter | Description | Return Value |
|-------------|--|---|
| addr_eeprom | データが書き込まれる EEPROM 空間内の論理アドレス。 | エラーなし。書き込みは正常に実行されました。 |
| data_eeprom | addr_eeprom によってポイントされる EEPROM 空間に書き込まれるデータ。 | エラー。指定したアドレスは使用可能な EEPROM メモリ空間の外にあります。 |

¹ この関数は、EEPROM にデータを書き込みます。

EEPROM の読出し

EEPROM の読出し動作のフローチャートを図 4 に示します。EEPROM の読出し動作手順は以下のとおりです。

1. read_eeprom(addr)関数を呼び出して、アドレス・ロケーションに保存された EEPROM 値を読み出します。
2. アプリケーション・コードからの EEPROM 読出し要求において、ソフトウェアは、最新のデータで構成される現在のセクタを最初に決めます。フラッシュ・アドレスは、EEPROM アドレスと現在のセクタ開始アドレスでわかります。
3. 得られたフラッシュ・アドレスを使って read_flash()関数を呼び出し、読出しコマンドを実行します。
4. そのフラッシュ・アドレスから取得した 64 ビット幅のデータを処理します。その後、このアドレスのビットはマスクされ右シフトされて、アプリケーション・コードに返されます。

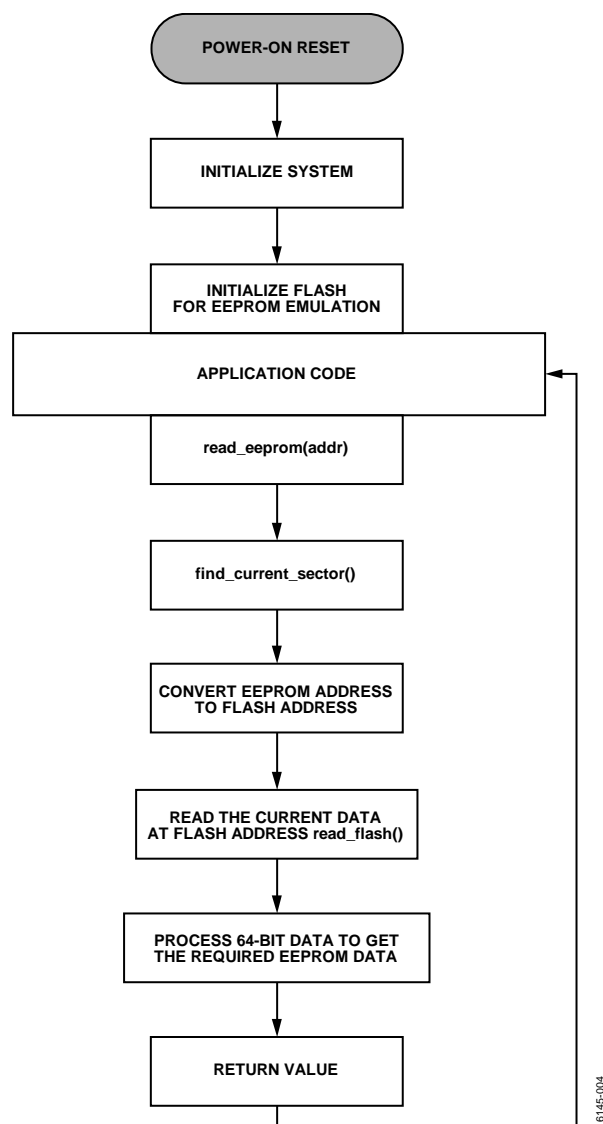


図 4. EEPROM 読出し動作のフローチャート

read_eeprom(uint16_t addr_eeprom)関数の詳細については、表 3 を参照してください。

表 3. EEPROM 読出し関数の説明

(read_eeprom(uint16_t addr_eeprom))¹

| Parameter | Description | Return Value |
|-------------|-------------------------------|---|
| addr_eeprom | データが読み出される EEPROM 空間内の論理アドレス。 | 値。アプリケーション・コードに 8 ビット・データが返されます。 エラー。指定したアドレスは使用可能な EEPROM メモリ空間の外にあります。 |

¹ この関数は、EEPROM にデータを書き込みます。

EEPROM の消去

EEPROM の消去動作のフローチャートを図 5 に示します。EEPROM の消去動作の手順は以下のとおりです。

1. `erase_eeeprom()`関数を呼び出して、フラッシュ・メモリに割り当てられた EEPROM 空間全体を消去します。

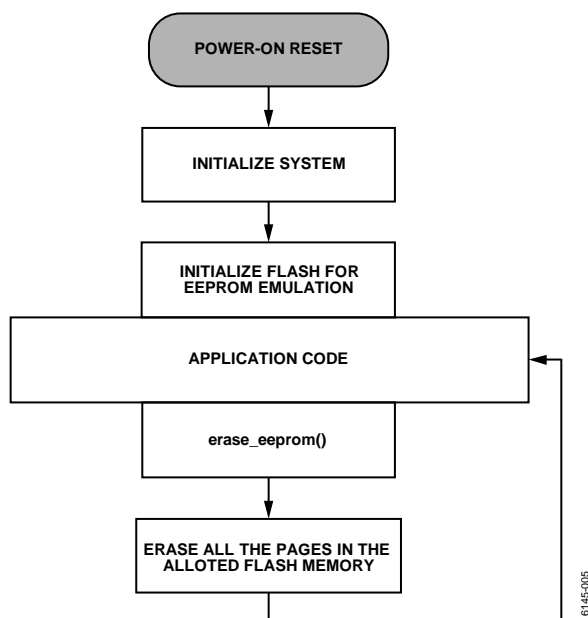


図 5. EEPROM 消去動作のフローチャート

フラッシュ・メモリ内の、EEPROM エミュレーション専用割り当てられた全てのページが消去されます。アプリケーション・コード内でこの操作を使用する場合は、注意が必要です。

`erase_eeeprom()`関数の詳細については、表 4 を参照してください。

表 4. EEPROM 消去関数の説明 (`erase_eeeprom()`)¹

| Parameter | Description | Return Value |
|----------------|--|--|
| Not applicable | この関数は EEPROM に割り当てられた全てのフラッシュ・ページを消去します。 | エラーなし。消去は正常に実行されました。 エラー。フラッシュ・コントローラがビジー状態で消去を実行できません。 |

¹ この関数は EEPROM メモリ空間を消去します。この関数を呼び出すと、全てのデータが失われます。

FLASH

ADuCM4050 プロセッサは 512kB の組み込みフラッシュ・メモリを内蔵しており、フラッシュ・コントローラを介してアクセスすることができます。組み込みフラッシュ・メモリは 72 ビット幅のデータ・バスを備えており、1 回のアクセスで、2 個の 32 ビット・ワードのデータと、それに対応する 8 ビットの ECC バイトを送ることができます。メモリはそれぞれ 2kB ずつのページで構成され、更に 256 バイトが ECC 用に予約されています。デフォルトでは、ECC は ADuCM4050 のフラッシュの全ユーザー空間で有効化されています。フラッシュのロケーションへの書き込みによって、そのロケーションのデータと ECC バイトが更新されます。フラッシュ・メモリは、消去手順を経て初めてビット値をロジック 0 からロジック 1 に変更できるので、1 回の書き込み動作しか許可されません。そのため、ECC バイトが更新される

のは 1 回のみです。消去手順を経ずに同じロケーションへの書き込みを再度実行すると、ECC エラーが発生します。

フラッシュへの書き込み

フラッシュ・メモリは、消去時には各ビットを 1 に設定し、データの書き込みやプログラム時には該当ビットを選択的に 0 にクリアすることで動作します。書き込み動作でビットを 0 から 1 に設定することはできません。このため、一般的な書き込みアクセスの際は、予め消去動作を行う必要があります。

キーホール書き込みは間接的な書き込み動作で、通常はユーザがプログラムするコードによって、ターゲット・アドレスとデータ値を使ってメモリ・マップ・レジスタがプログラムされ、その後バックグラウンドで書き込み動作を行うよう、フラッシュ・コントローラにコマンドが送られます。フラッシュ・コントローラは、キーホール書き込みによるフラッシュ・メモリへの書き込みアクセスだけをサポートしています。書き込みアクセスに関するこの制約により、フラッシュ・コントローラによる書き込みは、アトミックなダブル・ワード (64 ビット) 動作として正しく行われます。

EEPROM のデータを使って生成される LSB および MSB データ・パケットは、キーホール・データ・レジスタに送られます。書き込みコマンドがアサートされると、フラッシュ・コントローラは、所定のフラッシュ・アドレスへの 64 ビットのデュアル・ワード書き込みを開始します。

フラッシュ・メモリへのワード (32 ビット)、ハーフ・ワード (16 ビット)、バイト (8 ビット) の書き込みは、ADuCM4050 フラッシュ・メモリではサポートされていません。

`write_flash(uint32_t addr, uint32_t lower_data, uint32_t upper_data)`関数の詳細については、表 5 を参照してください。

表 5. フラッシュ書き込み関数の説明

(`write_flash(uint32_t addr, uint32_t lower_data, uint32_t upper_data)`)¹

| Parameter | Description | Return Value |
|------------|--|---|
| addr | EEPROM エミュレーション用に割り当てられたフラッシュ・メモリ空間内のアドレス。 | エラーなし。書き込みは正常に実行されました。 |
| lower_data | ダブル・ワードの下位 32 ビット。 | エラー。指定したアドレスは使用可能な EEPROM メモリ空間の外にあります。 |
| upper_data | ダブル・ワードの上位 32 ビット。 | |

¹ この関数は、変換された EEPROM アドレスとデータを `write_eeeprom()`関数から受信して、フラッシュ・コントローラへ書き込みコマンドを送出します。

フラッシュの読出し

フラッシュ・メモリは、自動初期化プロセス後にのみ読み出すことができます。フラッシュ・メモリを読み出すと、64 ビットのダブル・ワードが返されます。

フラッシュ・アドレス情報はフラッシュ・コントローラに提供され、それによって読出しデータが返されます。このデータが更に EEPROM インターフェースで処理され、EEPROM 値が得られます。

read_flash(uint32_t addr)関数の詳細については、表 6 を参照してください。

表 6. フラッシュ読み出し関数の説明
(read_flash(uint32_t addr))¹

| Parameter | Description | Return Value |
|-----------|--|--|
| addr | EEPROM エミュレーション用に割り当てられたフラッシュ・メモリ空間内のアドレス。 | データの読み出し。マスクング用に 64 ビット・データが返され、戻り値として read_eeprom()関数へ送信されます。 エラー。変換したアドレスは、割り当てられたフラッシュ・メモリ空間の外にあります。 |

¹この関数は、変換された EEPROM アドレスを read_eeprom()関数から受信して、フラッシュ・コントローラへ読み出しコマンドを送出します。

フラッシュの消去

EEPROM への書き込み中にページが変更されると、erase_flash(page)関数を呼び出すことによって、前ページ上でページ消去コマンドがアサートされます。ページ消去の前には、EEPROM への書き込みのセクションで説明したように、データの移動が行われます。

erase_flash(uint8_t PAGE)関数の詳細については、表 7 を参照してください。

表 7. フラッシュ消去関数の説明
(erase_flash(uint8_t PAGE))¹

| Parameter | Description | Return Value |
|-----------|---------------------------|--|
| PAGE | 割り当てられたフラッシュ・メモリ空間のページ番号。 | エラーなし。ページ消去は正常に終了しました。 エラー。指定したページ値は、割り当てられたフラッシュ・メモリ空間の外にあります。 |

¹この関数は、割り当てられたフラッシュ・メモリ空間内にあるページの内容を全て消去します。

制約事項

一般的な EEPROM では、1 つのロケーションが更新されると、消去サイクルを 1 サイクルだけカウントした後、そのアドレスに書き込みを行います。他のロケーションは変更しません。

本アプリケーション・ノートで説明したエミュレートされた EEPROM では、1 つのロケーションを更新すると現在のセクタから隣のセクタへデータが移動されて、書き込みサイクルの数だけ EEPROM サイズが消費されます。ロケーション更新の都度、データが次のセクタに移動し、そのセクタが次ページにある場合はページ消去が実行されます。複数回の書き込みを含むデータ移動によって、フラッシュ・メモリの実効的な書き換え回数は減少します。

これらの制約に対処するには、2 通りの方法があります。1 つ目は、エミュレートする EEPROM のサイズを選択する際に注意を払うことです。EEPROM サイズを小さくするとデータ移動時の書き込みサイクルが短くなり、そのため、フラッシュ・メモリの書き換え回数が間接的に増加します。2 つ目は、EEPROM への不要な書き込みを避けることです。不要な書き込みをなくすことによって、有効書き換え回数が増加します。例えば、システムが書き込み命令を出す必要があるのは、電源異常シーケンス時のみです。通常動作時のデータ保存には、RAM（ランダム・アクセス・メモリ）バッファを使用することができます。

エミュレートされた EEPROM への不要な書き込みには、ソフトウェアによって処理できるものもあります。例えば、書き込みデータが 0xFF で、書き込み先ロケーションの現状データが 0xFF の場合、フラッシュ・メモリへの書き込み命令は送出されません。

まとめ

本アプリケーション・ノートは、ADuCM4050 を使って EEPROM とフラッシュ・メモリの物理的な差を埋めるものです。このエミュレートされた EEPROM は一般的な EEPROM と同様に機能し、シリコン面積、入出力バス・リソース、製造コスト、部品表 (BOM) などに関する課題を解決します。

このアプリケーション・ソフトウェアでは、EEPROM エミュレーション用に 2 ページ分のフラッシュ・ページが予約されているため、8B~2kB (1 ページ) の大きな EEPROM サイズがユーザに提供されます。ソフトウェアが 4 フラッシュ・ページ分に拡張されている場合、最大 EEPROM サイズは、8~64 ビットに設定可能なワード長を含め、4kB となります。

エミュレートする EEPROM のサイズとフラッシュ・メモリの書き換え回数はトレードオフの関係にあるので、ハードウェアの効率を向上させるには、適切なサイズを選択する必要があります。ソフトウェアは ADuCM4050 のフラッシュ・デバイスへの不要な書込みの一部を処理し、これによって書き換え回数を実効的に増加します。