

ADUCM3027/ADUCM3029 のフラッシュ・メモリを利用した EEPROM エミュレーション

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、補正係数、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を書き換えたり更新したりできることです。

ADuCM3027/ADuCM3029 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。内蔵フラッシュで EEPROM のエミュレーションを行えば、設計から EEPROM をなくすことができるので、BOM コストが削減できます。したがって、ソフトウェアもより単純化されます。

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。ADuCM3027 の 1 ページには 2 kB 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。

- 1 バイト幅のデータの読出しと書込み。
- 他のロケーションにあるデータを維持しながら、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こと (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)。

このアプリケーション・ノートでは、図 1 に示すように、ADuCM3027/ADuCM3029 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。

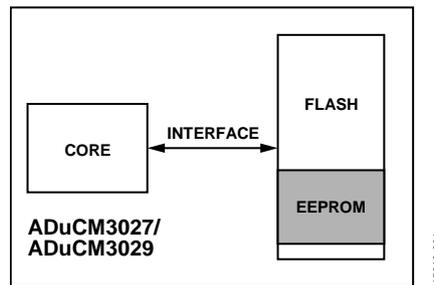


図 1. ADuCM3027/ADuCM3029 の内蔵フラッシュと EEPROM システムの概要

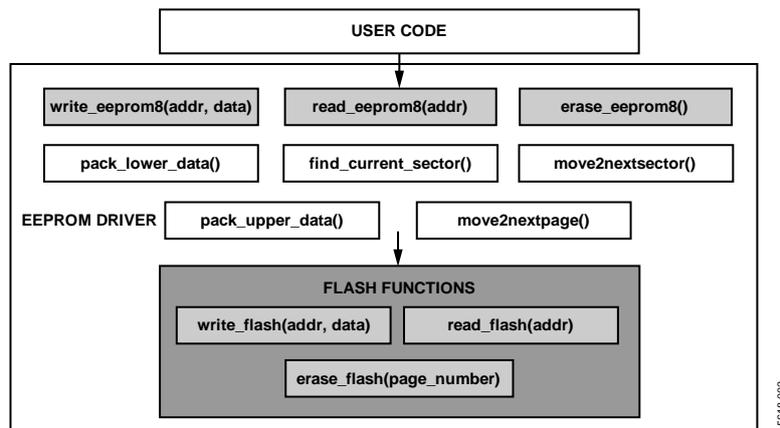


図 2. ADuCM3027/ADuCM3029 フラッシュ EEPROM エミュレーション・ソフトウェアの構造

アナログ・デバイス社は、提供する情報が正確で信頼できるものであることを期していますが、その情報の利用に関して、あるいは利用によって生じる第三者の特許やその他の権利の侵害に関して一切の責任を負いません。また、アナログ・デバイス社の特許または特許の権利の使用を明示的または暗示的に許諾するものでもありません。仕様は、予告なく変更される場合があります。本紙記載の商標および登録商標は、それぞれの所有者の財産です。※日本語版資料は REVISION が古い場合があります。最新の内容については、英語版をご参照ください。

目次

はじめに.....	1	フラッシュ.....	5
背景.....	1	制約事項.....	6
改訂履歴.....	2	まとめ.....	7
動作原理.....	3		
EEPROM.....	3		

改訂履歴

3/2017—Revision 0: Initial Version

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 回の書込みコマンドで更新できるデータは 1 バイトだけです。ただし、フラッシュ・メモリ・デバイスでは、2 回の書込み動作の間に消去シーケンスを設定した場合に限り、複数のバイトを書き込み、それに応じてデータを更新することができます。1 バイトの書込みと読み出しが可能な EEPROM をフラッシュ・メモリでエミュレートするには、EEPROM 動作と同様の読み出し、変更、書込みシーケンスを設定する必要があります。

このセクションに示す手順では 2 つのフラッシュ・ページを使用しますが、これらのページを 3 ページ以上に拡張して、セクター・タグで構成される複数のセクターに分割することができます。このセクター・タグは、現在処理中のセクターと、そのセクター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。どのセクターも最後のロケーションがセクター・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・バスに等しいサイズです。フラッシュ・ページ内のセクター・サイズとセクター数は、エミュレートされる EEPROM のサイズに依存します。

EEPROM

EEPROM の書込みおよび読み出し機能には、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要求に従ってデータやアドレス情報を処理し、提供します。

EEPROM への書込み

EEPROM の書込み動作のフローチャートを図 3 に示します。EEPROM の書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、現在のセクターを検索します。この検索は、セクター・タグと、それに対応するセクター・タグ値に基づいて行われます。返される値は、現在のセクター開始アドレス (フラッシュ・メモリ上の物理ロケーション) です。
2. 現在のセクター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。
ADuCM3027/ADuCM3029 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. 得られたフラッシュ・アドレスのデータを読み出します。このデータが 0xFF に等しい場合、フラッシュ・メモリに書き込む 64 ビット幅のデータを作成するために、ビットをマスクして EEPROM データを左シフトすることにより、32 ビットの LSB (最下位ビット) および MSB (最上位ビット) データ・パケットを生成します。

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、LSB データ・パケットおよび MSB データ・パケットです。
5. フラッシュ・メモリへの書込み動作が正常に終了した後、`update_tag ()` 関数を呼び出して、現在のセクターのセクター・タグを更新します。

得られたフラッシュ・アドレスにデータがすでに存在する場合、データ読み取り関数は 0xFF を返しません。この場合は、得られたフラッシュ・アドレスの前後にあるデータは、`move2nextsector ()` 関数を呼び出すと、次のセクター、つまり隣接セクターに移動します。LSB および MSB データ・パケットに変換された EEPROM データは、次のセクターの新しいフラッシュ・アドレスに書き込まれます。したがって、すでに書き込み済みの EEPROM のロケーションに書込みが行われると、データはその都度、変更されたデータを格納するロケーション含む次のセクターに移されます。

新しいセクター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)` を呼び出すとフラッシュ・ページ消去コマンドが実行されます。すべてのアドレス・レジスタを更新するには、`move2nextpage ()` 関数を使用します。

`write_eeprom (uint16_t addr_eeprom, uint8_t data_eeprom)` 関数の詳細については、表 1 を参照してください。

EEPROM の読み出し

EEPROM の読み出し動作のフローチャートを図 4 に示します。EEPROM の読み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 読み出し要求において、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ーを最初に決めます。フラッシュ・アドレスは、EEPROM アドレスと現在のセクター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、読み出しコマンドを実行します。
4. フラッシュ・アドレスから受け取った 64 ビット幅データを処理します。次に、このアドレスのビットをマスクして右シフトし、アプリケーション・コードに提供します。

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

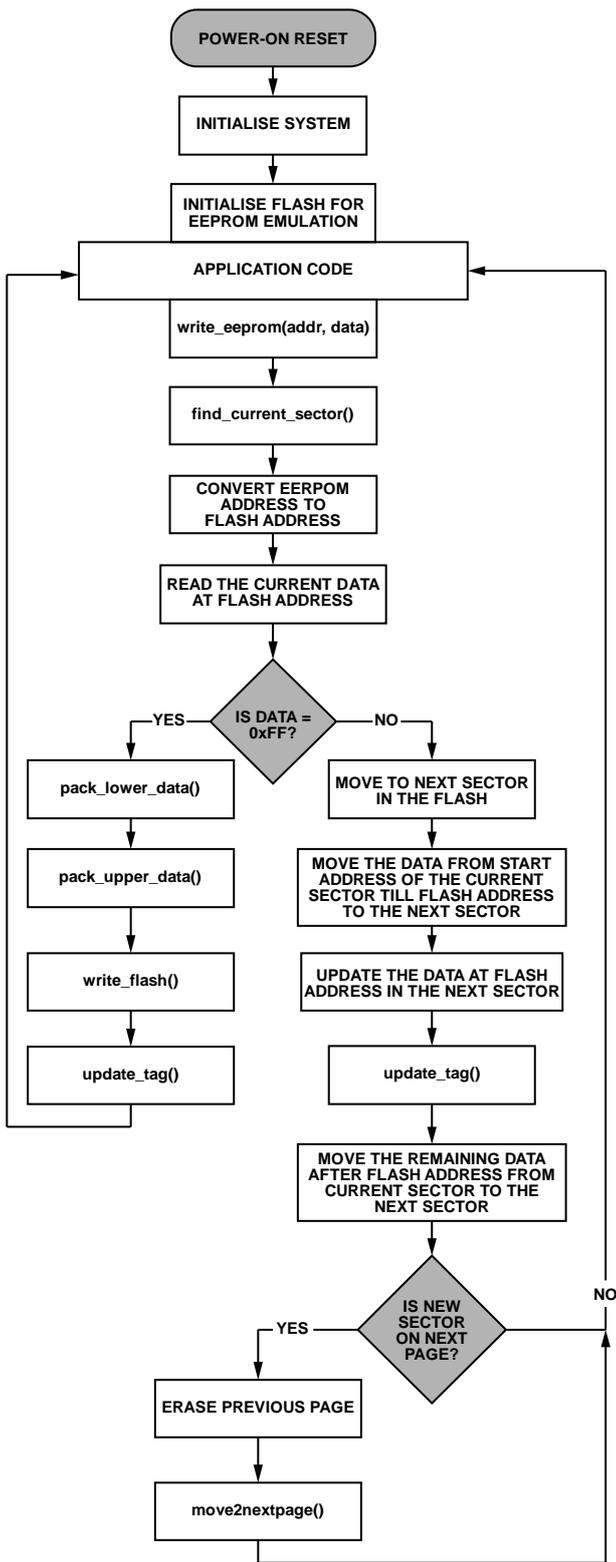


図 3. EEPROM 書込み動作

15618-003

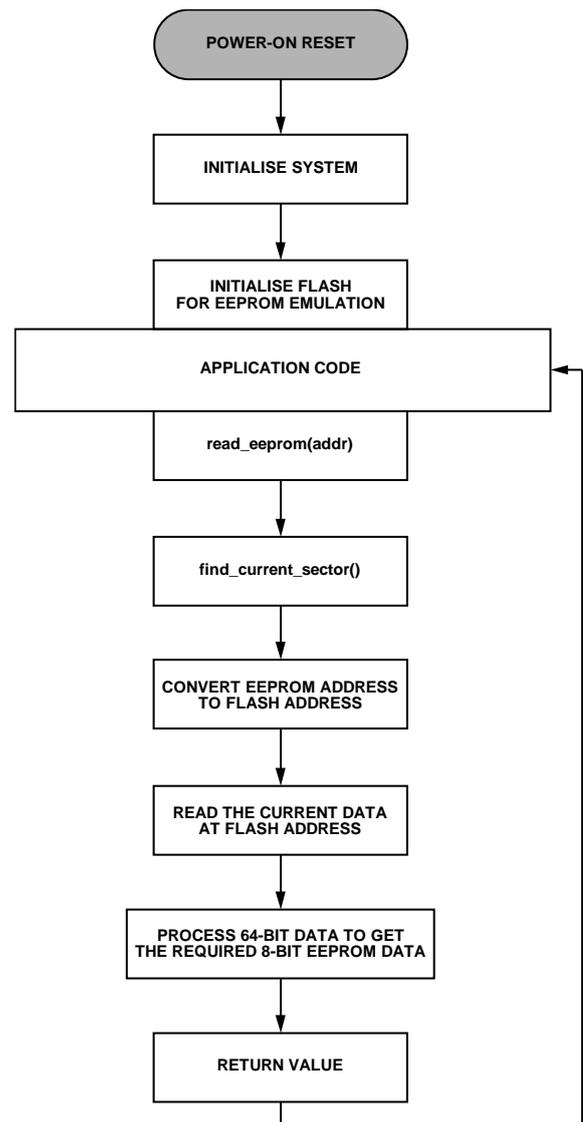


図 4. EEPROM 読出し動作

15618-004

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_eeprom ()` 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 スペース全体を消去します。

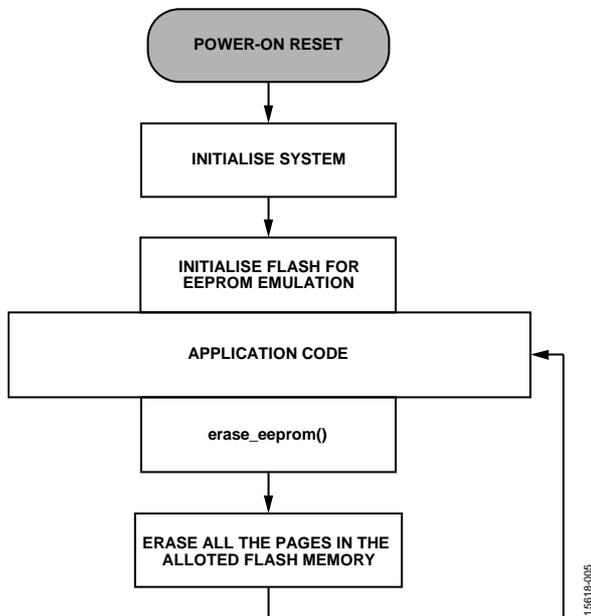


図 5. EEPROM 消去動作

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すべてのページが消去されます。このため、アプリケーション・コード内でこの操作を使用するときは注意が必要です。

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

フラッシュ

ADuCM3027/ADuCM3029 プロセッサは 128 kB と 256 kB の組込みフラッシュ・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 ビットのエラー訂正コード (ECC) を送ることができます。メモリはそれぞれ 2 kB ずつのページで構成され、さらに 256 バイトが ECC 用に予約されています。

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、データ書き込み (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、前もって消去動作を行う必要があります。

キーホール書き込みは、ユーザー・コードがターゲット・アドレスとデータ値を使ってメモリ・マップ・レジスタをプログラムし、その後に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を送る間接的な書き込み動作です。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ることが保証されています。

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。

ワード (32 ビット)、ハーフ・ワード (16 ビット)、およびバイト (8 ビット) 書き込み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 4 を参照してください。

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。このデータがさらに 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。

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。ページを消去する場合は、前述のように消去前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 5 を参照してください。

制約事項

実際の 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。

この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ーから隣のセクター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。したがって、ロケーションの更新の都度、データが次のセクターに移動し、そのセクターが次ページにある場合はページ消去が実行されることになります。この動作によって、フラッシュ・メモリの有効書換え回数は減少します。

これらの制約に対処するには、以下の点に留意します。

- エミュレートする EEPROM のサイズを選択する際は注意が必要です。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・メモリの書換え回数が間接的に増加します。
- EEPROM への不要な書き込みは避けるようにします。不要な書き込みをなくすことによって、有効書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 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常データが 0xFF の場合、フラッシュ・メモリへの書き込みは行われません。

まとめ

このアプリケーション・ノートの目的は、ADuCM3027/ADuCM3029 を使って、EEPROM とフラッシュ・メモリの物理的な差を埋めることです。このエミュレートされた EEPROM は実際の EEPROM 同様に機能し、シリコン面積、入出力バス・リソース、製造コストなどに関する課題を解決します。

このアプリケーション・ノートでは、エミュレーション用に大きい EEPROM サイズ（64 バイト～1024 バイト）を設定しています。エミュレートされた EEPROM とフラッシュ・メモリの書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。この他、ADuCM3027/ADuCM3029 のフラッシュ・デバイスへの不要な書込みの一部をソフトウェアで処理することで、有効書換え回数は増加します。

表 1. EEPROM 書込み関数の説明

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。書込みは正常に終了しました。
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。

表 2. 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 からデータを読み出します。

表 3. EEPROM 消去関数の説明

erase_eeprom() ¹	
Parameter	Return Value
Not applicable	エラーなし。消去が正常に終了しました。 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、すべてのデータが失われます。

表 4. フラッシュ書込み関数の説明

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。書込みは正常に終了しました。
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_eeprom () 関数から受信して、フラッシュ・コントローラへ書込みコマンドを送出します。

表 5. フラッシュ消去関数の説明

erase_flash(uint8_t PAGE) ¹	
Parameter	Return Value
ページ。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すべて消去します。

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。