

S1C31 Family Application Note

S1C31 Family

EEPROM エミュレーション ライブラリマニュアル

arm

評価ボード・キット、開発ツールご使用上の注意事項

1. 本評価ボード・キット、開発ツールは、お客様での技術的評価、動作の確認および開発のみに用いられることを想定し設計されています。それらの技術評価・開発等の目的以外には使用しないで下さい。本品は、完成品に対する設計品質に適合していません。
2. 本評価ボード・キット、開発ツールは、電子エンジニア向けであり、消費者向け製品ではありません。お客様において、適切な使用と安全に配慮願います。弊社は、本品を用いることで発生する損害や火災に対し、いかなる責も負いかねます。通常の使用においても、異常がある場合は使用を中止して下さい。
3. 本評価ボード・キット、開発ツールに用いられる部品は、予告無く変更されることがあります。

本資料のご使用につきましては、次の点にご留意願います。

本資料の内容については、予告無く変更することがあります。

1. 本資料の一部、または全部を弊社に無断で転載、または、複製など他の目的に使用することは堅くお断りいたします。
2. 本資料に掲載される応用回路、プログラム、使用方法等はあくまでも参考情報であり、これらに起因する第三者の知的財産権およびその他の権利侵害あるいは損害の発生に対し、弊社はいかなる保証を行うものではありません。また、本資料によって第三者または弊社の知的財産権およびその他の権利の実施権の許諾を行うものではありません。
3. 特性値の数値の大小は、数直線上の大小関係で表しています。
4. 製品および弊社が提供する技術を輸出等するにあたっては「外国為替および外国貿易法」を遵守し、当該法令の定める手続きが必要です。大量破壊兵器の開発等およびその他の軍事用途に使用する目的をもって製品および弊社が提供する技術を費消、再販売または輸出等しないでください。
5. 本資料に掲載されている製品は、生命維持装置その他、きわめて高い信頼性が要求される用途を前提としていません。よって、弊社は本（当該）製品をこれらの用途に用いた場合のいかなる責任についても負いかねます。
6. 本資料に掲載されている会社名、商品名は、各社の商標または登録商標です。
Arm, Cortex, Keil および μ Vision は、Arm Limited(またはその子会社)の US またはその他の国における登録商標です。IAR Systems, IAR Embedded Workbench, C-SPY, I-jet, IAR および IAR システムズのロゴタイプは、IAR Systems AB が所有権を有する商標または登録商標です。SEGGER および J-Link は、SEGGER Microcontroller GmbH & Co. KG の商標または登録商標です。All rights reserved.
“Reproduced with permission from Arm Limited. Copyright © Arm Limited”

目 次

1. 概要.....	2
1.1 機能.....	2
1.2 動作環境	3
1.3 使用上の注意事項.....	3
2. ライブラリの構成.....	4
2.1 フォルダ構成	4
2.2 ライブラリ機能	5
3. ライブラリの使い方.....	6
3.1 アプリケーションプログラムへの適用方法	6
3.2 内蔵 RAM、フラッシュメモリ使用量	7
3.3 書き込み時間	7
3.4 ライブラリ使用上の注意.....	7
3.5 サンプルソフトウェア	8
4. ライブラリ仕様	9
4.1 ライブラリ関数詳細.....	9
4.2 エラーコード定義.....	10
Appendix.....	11
A. ライブラリをプロジェクトへ組み込む方法(IAR EAWRM).....	11
B. ライブラリをプロジェクトへ組み込む方法(MDK-ARM)	14
改訂履歴表	17

1. 概要

1. 概要

S1C31 EEPROM エミュレーションライブラリは、対象機種の内蔵フラッシュメモリを用いて EEPROM をエミュレーションするためのライブラリを提供します。このライブラリをアプリケーションプログラムにリンクし、アプリケーションプログラムから関数コールすることによって、内蔵フラッシュメモリを EEPROM のように使用することが可能です。

本ライブラリおよびサンプルソフトウェアは、S1C31xxx 周辺回路サンプルソフトウェアパッケージに同梱されています。S1C31xxx 周辺回路サンプルソフトウェアパッケージは、弊社 Web より入手可能です。

また、本マニュアルに加えて、“S1C31xxx テクニカルマニュアル” も併せてご参照ください。

1.1 機能

本ライブラリは、S1C31 シリーズ内蔵フラッシュメモリを EEPROM として使用するエミュレーション機能を実現します。4K バイト～64K バイトの EEPROM エミュレーション用領域を使用して、32～512 バイトの容量の EEPROM をエミュレートします。

本ライブラリが対応する機種における内蔵フラッシュメモリの書き換え保証回数は 1000 回程度ですが、EEPROM の 1 アドレスのデータをフラッシュメモリの 1 セクタで管理することによって、エミュレートされた EEPROM 上の各アドレスの書き換え回数を理論上 100,000 回以上に増やします。

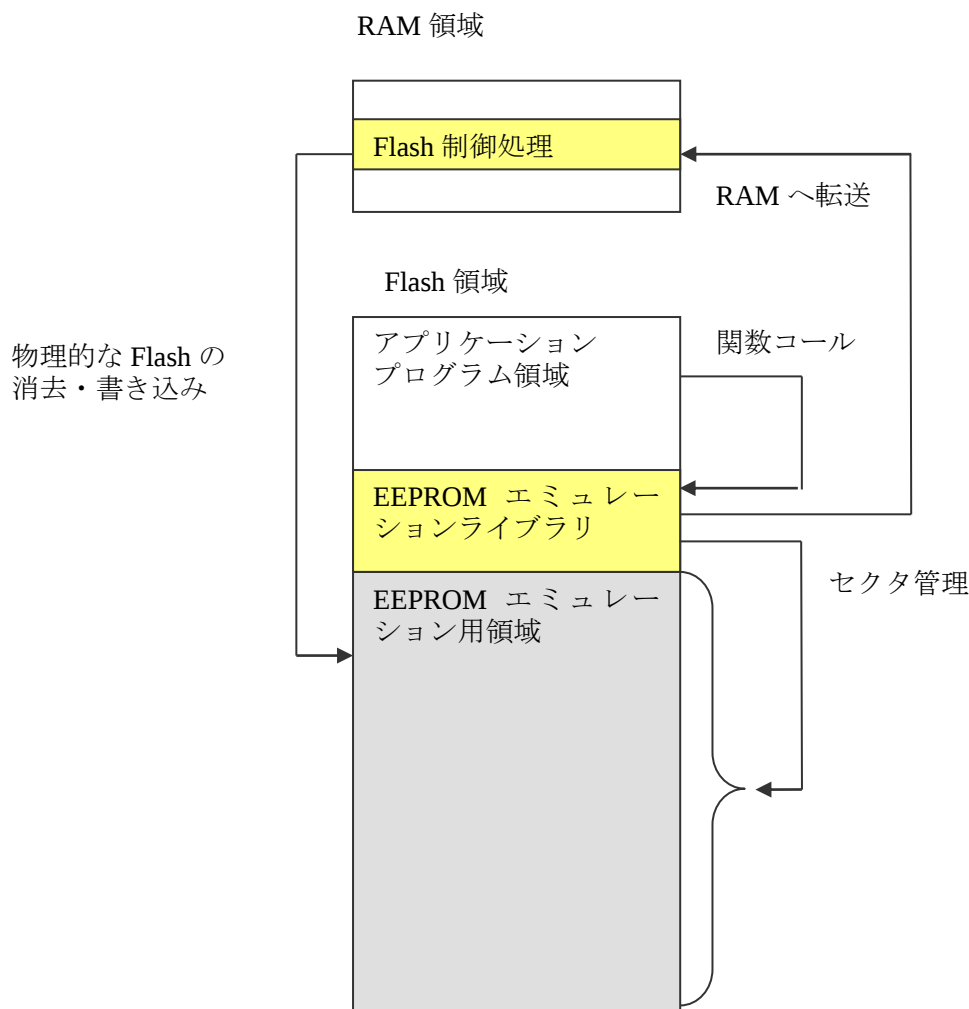


図 1.1.1 EEPROM エミュレーションライブラリの機能

1.2 動作環境

サンプルソフトウェアの書き込み時およびデバッグ時には、以下が必要です。

- 評価ボード
 - S1C31 シリーズ搭載 S5U1C31xxxTx 評価ボード
- デバッグプローブ *1*2
 - IAR Systems 社製 I-jet または SEGGER 社製 J-Link
- 統合開発環境
 - IAR Embedded Workbench for ARM® (IAR EWARM) または MDK-ARM® (μVision)
- S1C31SetupTool パッケージ
 - フラッシュローダおよび構成ファイル (.svd など)
- S1C31xxx 周辺回路サンプルソフトウェアパッケージ

*1: サンプルソフトウェアからライブラリの関数コールには、デバッグプローブは不要となります。

*2: I-jet は IAR EWARM でのみ使用可能です。J-Link は IAR EWARM と MDK-ARM の両方で使用可能です。

上記の詳細については、それぞれに付属するマニュアルをご参照ください。

1.3 使用上の注意事項

S1C31 EEPROM エミュレーションライブラリおよびサンプルソフトウェアは、参考資料です。これらに起因する不具合が発生した場合、弊社は如何なる責任についても負いません。製品上でお使いになる場合には、十分な動作検証を実施してください。

本マニュアルは、S1C31 シリーズの各機種に用意している EEPROM エミュレーションライブラリ共通の資料です。機種ごとに異なる仕様 (RAM 使用量等) については、S1C31xxx 周辺回路サンプルソフトウェアパッケージに同梱しているリードミーをご参照ください。

2. ライブラリの構成

2. ライブラリの構成

2.1 フォルダ構成

S1C31xxx 周辺回路サンプルソフトウェアパッケージに含まれる S1C31 EEPROM エミュレーションライブラリおよびサンプルソフトウェア、関連プログラムの構成は以下の通りです。

```
S1C31xxxSamplePKG_very_yy.zip
[S1C31xxxSamplePKG_very_yy]
|- [Licenses]
|- [Drivers] : ドライバ群
|   |- [board] : 評価ボード関連ドライバ
|   |- [CMSIS] : CMSIS ドライバ
|       |- [Device]
|           |- [S1C31xxx]
|               |- [Include]
|                   |- S1C31xxx.h : CMSIS 周辺回路アクセスレイヤヘッダファイル
|                   |- ...
|               |- [Source]
|                   |- [ARM]
|                   |- [IAR]
|                       |- startup_S1C31xxx.s : CMSIS スタートアッププログラム
|                       |- system_S1C31xxx.c : CMSIS 周辺回路アクセスレイヤプログラム
|           |- [Driver]
|               |- [Include]
|                   |- Driver_Common.h : 共通ドライバ定義
|                   |- Driver_EEPROM.h : CMSIS EEPROM エミュレーションライブラリドライバ定義
|                   |- ...
|               |- [Source]
|                   |- Driver_EEPROM.c : CMSIS EEPROM エミュレーションライブラリドライバ
|           |- [SVD]
|   |- [sePeripheralLibrary] : 周辺回路ライブラリ
|- [Middlewares] : ミドルウェア群
|   |- [seEepromLibrary] : EEPROM エミュレーションライブラリ
|       |- [Device]
|           |- [S1C31xxx]
|               |- seEepromLibraryS1C31xxx.a : IAR EAWRM 用ライブラリ
|               |- seEepromLibraryS1C31xxx.lib : MDK-ARM 用ライブラリ
|       |- [Include]
|           |- DataFlashCommand.h : EEPROM エミュレーションライブラリドライバ定義
|           |- DataFlashConfig.h : 設定用ヘッダファイル
|       |- eepromLibraryForS1c31xxx_readme_e.txt : リードミー
|       |- eepromLibraryForS1c31xxx_readme_j.txt
|   |- ...
|- [Projects] : サンプルソフトウェア群
|   |- [Applications] : 各種アプリケーションソフトウェア
|       |- [EEPROM] : EEPROM エミュレーションライブラリ用サンプルソフトウェア
|           |- [ARM] : MDK-ARM 用プロジェクト
|           |- [IAR] : IAR EWARM 用プロジェクト
|           |- main.c
|       |- ...
|   |- ...
README_e.txt
README_j.txt
```

図 2.1.1 S1C31xxx サンプルソフトウェアパッケージの構成

2.2 ライブラリ機能

本ライブラリが提供する関数は、Drivers¥CMSIS¥Driver¥Include¥Driver_EEPROM.h で定義をしています。
本ライブラリが提供する関数は、以下の通りです。

表 2.2.1 S1C31 EEPROM エミュレーションライブラリが提供する関数

関数名	機能概要
int32_t ProgramData (uint32_t addr, unsigned char *data, uint32_t cnt)	内蔵フラッシュメモリを書き込む
int32_t ReadData (uint32_t addr, unsigned char *data, int32_t cnt)	内蔵フラッシュメモリを読む込む
ARM_DRIVER_VERSION GetVersion (void)	本ライブラリバージョンの取得
ARM_FLASH_INFO * GetInfo (void)	内蔵フラッシュメモリの情報取得

3. ライブラリの使い方

3. ライブラリの使い方

S1C31 EEPROM エミュレーションライブラリおよびサンプルソフトウェアの使用方法について説明します。

3.1 アプリケーションプログラムへの適用方法

アプリケーションプログラム上で本ライブラリを使用する方法を説明します。尚、ライブラリをアプリケーションプログラムのプロジェクトに組み込む方法については“Appendix x. ライブラリをプロジェクトへ組み込む方法”をご参照ください。

1. ヘッダファイル宣言

本ライブラリを使用するソースファイルに、“Driver_EEPROM.h”をインクルード宣言します。

```
#include <stdio.h>
#include <string.h>
#include "Driver_EEPROM.h"
```

2. EEPROM サイズ、書き込みリトライ回数の設定

“Driver_EEPROM.h”に以下の値を設定します。

```
#define CONFIG_EEPROM_SIZE_MAX          (512)
#define CONFIG_RETRY_COUNT              (4)
```

- CONFIG_EEPROM_SIZE_MAX

エミュレートする EEPROM のサイズを設定します。設定可能なサイズについては、リードミーをご参照ください。

- CONFIG_RETRY_COUNT

書き込みに失敗した場合に、リトライを行う回数を設定します。リトライ回数を増やすと書き込みルーチンの処理時間が増えることになりパフォーマンスが低下しますので、数回程度にしてください。

3. 関数の追加

本ライブラリを使用するソースファイルに、ライブラリが提供する関数を追加します。関数の仕様については“4 章 ライブラリ仕様”をご参照ください。

extern ARM_DRIVER_FLASH Driver_Eeprom:

```
...
int main(void) {
    unsigned char compbuf[CONFIG_EEPROM_SIZE_MAX];
    ARM_FLASH_INFO *Info = Driver_Eeprom.GetInfo();
    Driver_Eeprom.GetVersion();

    if (Driver_Eeprom.ProgramData(0, updateLineBit, CONFIG_EEPROM_SIZE_MAX) == ARM_DRIVER_OK)
    {
        printf("Program: OK\r\n");
        Driver_Eeprom.ReadData(0, compbuf, CONFIG_EEPROM_SIZE_MAX);
        if (memcmp(updateLineBit, compbuf, CONFIG_EEPROM_SIZE_MAX) == 0) {
            printf("Verify: OK\r\n");
        } else {
            printf("Verify: NG\r\n");
        }
    }
    else {
        printf("Program: NG\r\n");
    }
}
...
```

EEPROM の書き込み

EEPROM の読み込み

3.2 内蔵 RAM、フラッシュメモリ使用量

本ライブラリでは内蔵 RAM およびフラッシュメモリ領域を使用します。各機種の EEPROM エミュレーションライブラリの RAM 使用量については、S1C31xxx 周辺回路サンプルソフトウェアパッケージに同梱しているリードミーをご参照ください。

3.3 書き込み時間

本ライブラリを使用した書き込み時間は、EEPROM のサイズ、フラッシュメモリの書き込み回数、使用する CPU のクロックソース等に影響されます。参考として、S1C31W74 用 EEPROM エミュレーションライブラリを使用して、EEPROM の同一アドレスに 100,000 回書き込みを行った場合の書き込み時間を以下に示します。

Typ.値	4.6ms
Max.値	27.4ms

実際の書き込み時間は、本ライブラリを使用するアプリケーションプログラムでご確認ください。

3.4 ライブラリ使用上の注意

本ライブラリの使用にあたり、以下の点に注意してください。

- ・ S1C31D01/S1C31D5x/S1C31W74 では、16 ビットタイマ(T16)の ch.0 を使用します。そのため 16 ビットタイマ ch.0 のレジスタの内容が変更されます。16 ビットタイマを使用しているアプリケーションプログラムで、本ライブラリを使用する場合はご注意ください。
- ・ 本ライブラリでは、システムクロックを高速クロック(OSC3 or IOSC)に変更します。そのため、クロックジェネレータ (CLG) の制御レジスタ内容が変更されますのでご注意ください。
- ・ 本ライブラリを使用する前に EEPROM エミュレーション用領域として使用するフラッシュメモリは消去しておく必要があります。また EEPROM エミュレーション用領域のアドレスまたはサイズを変更した場合にもフラッシュメモリの消去が必要です。
- ・ EEPROM エミュレーション用領域として (CONFIG_EEPROM_SIZE_MAX * DATAFLASH_SECTOR_SIZE) バイトのサイズを使用します。フラッシュメモリのサイズを超えないように CONFIG_EEPROM_SIZE_MAX を設定してください。
- ・ 本ライブラリ使用時には“S1C31xxx テクニカルマニュアル”に記載の基本外部結線図のとおり Vpp 端子にコンデンサを接続し、Vpp 端子と他の端子との接続を切り離してください。
- ・ 本ライブラリを使用する際は、フラッシュメモリの書き換え可能回数に注意してください。フラッシュメモリの仕様については“S1C31xxx テクニカルマニュアル”をご参照ください。
- ・ 書き込み関数実行時には、各機種のデータシートで定められている Flash 書き換え時 VDD 動作電圧 (VPP 内部生成時)を安定供給してください。電圧が範囲外になった場合は書き込んだ値は保証されません。

3.5 サンプルソフトウェア

1. サンプルソフトウェア仕様

本サンプルソフトウェアでは、S1C31 EEPROM エミュレーションライブラリを使用して、アドレス 0 ～ (CONFIG_EEPROM_SIZE_MAX - 1) の範囲に 0 からインクリメントデータを書き込みます。

2. 準備

本サンプルソフトウェアのプロジェクトを統合開発環境で実行する方法については、“S1C31xxx 周辺回路サンプルソフトウェアマニュアル”をご参照ください。

3. 動作概要

- ① 内蔵フラッシュメモリの情報を取得します。(任意)
- ② S1C31 EEPROM エミュレーションライブラリのバージョンを取得します。(任意)
- ③ EEPROM に更新用データ updateLineBit [] (512byte) を書き込みます。
- ④ EEPROM から 512byte のデータを読み込みます。
- ⑤ 読み込み済みデータ cmpbuf[]と、更新用データ updateLineBit[]を比較し、結果を表示します。(ベリファイ)

4. ライブラリ仕様

4.1 ライブラリ関数詳細

S1C31 EEPROM エミュレーションライブラリが提供する関数の詳細について記します。

関数名		
int32_t ProgramData (uint32_t addr, unsigned char *data, uint32_t cnt)		
引数		
addr	uint32_t	書き込み先アドレス 有効範囲は、0 ~ (CONFIG_EEPROM_SIZE_MAX - 1)です。
data	const void *	書き込みデータ 書き込みデータへのポインタを表す。同ポインタは、RAM 空間を指していなければなりません。
cnt	uint32_t	書き込みデータサイズ
戻り値		
uint32_t	書き込み結果(エラーコード)	
機能		
EEPROM を書き込む。 ① 引数が EEPROM アドレス以内かチェックする。 ② 指定した書き込み先アドレスに対し書き込みデータの書き込みを行う。 ③ 書き込み先アドレスが書き込みデータであるか確認する。(ペリファイ) ④ 書き込み結果を返す。		
備考		
なし		

関数名		
int32_t ReadData (uint32_t addr, unsigned char *data, int32_t cnt)		
引数		
addr	uint32_t	読み込み先アドレス 0 ~ (CONFIG_EEPROM_SIZE_MAX - 1)です
data	const void *	読み込みデータサイズ 読み込みデータへのポインタを表す。同ポインタは、RAM 空間を指していなければなりません。
cnt	uint32_t	読み込みデータサイズ
戻り値		
uint32_t	ARM_DRIVER_OK (0)	
機能		
EEPROM を読み込む。 ① 引数が EEPROM アドレス以内かチェックする。 ② 指定した読み込み先アドレスに対し読み込みを行う。 ③ 読み込み結果を返す。		
備考		
なし		

関数名	
ARM_DRIVER_VERSION GetVersion (void)	
戻り値	
ARM_DRIVER_VERSION	本ライブラリのバージョン
機能	
本ライブラリバージョンを取得する。	
備考	
なし	

関数名

4. ライブラリ仕様

ARM_FLASH_INFO * GetInfo (void)	
戻り値	
ARM_FLASH_INFO *	内蔵フラッシュメモリの情報
機能	
内蔵フラッシュメモリの情報として以下を取得する。 ・セクタ数 ・EEPROM サイズ	
備考	
なし	

4.2 エラーコード定義

各関数の戻り値で使用されているエラーコードは、以下の通りです

表 4.2.1 エラーコード

定義	値	意味
ARM_DRIVER_OK	0	正常終了
ARM_DRIVER_ERROR_TIMEOUT	-3	タイムアウト／ペリフェイエラー
ARM_DRIVER_ERROR_UNSUPPORTED	-4	サポートしていない操作
ARM_DRIVER_ERROR_PARAMETER	-5	引数エラー

これらの定義は、Drivers¥CMSIS¥Driver¥Include¥Driver_Common.h で定義をしています。

Appendix

A. ライブラリをプロジェクトへ組み込む方法(IAR EWARM)

統合開発環境 IAR EWARM で作成したアプリケーションプログラムのプロジェクトに、S1C31 EEPROM エミュレーションライブラリを組み込む方法を以下に記します。IAR EWARM の詳細については、付属のマニュアルをご参照ください。

1. ライブラリの追加

- (1) IAR EWARM メニューの [プロジェクト] > [オプション] を選択します。
- (2) 表示されたダイアログの [カテゴリ] リストから [リンク] を選択します。
- (3) [ライブラリ] タブから、“追加ライブラリ” に S1C31xxx 周辺回路サンプルソフトウェアパッケージの以下に同梱されている本ライブラリを追加します。

Middlewares¥seEepromLibrary¥Device¥S1C31xxx¥seEepromLibraryS1C31xxx.a

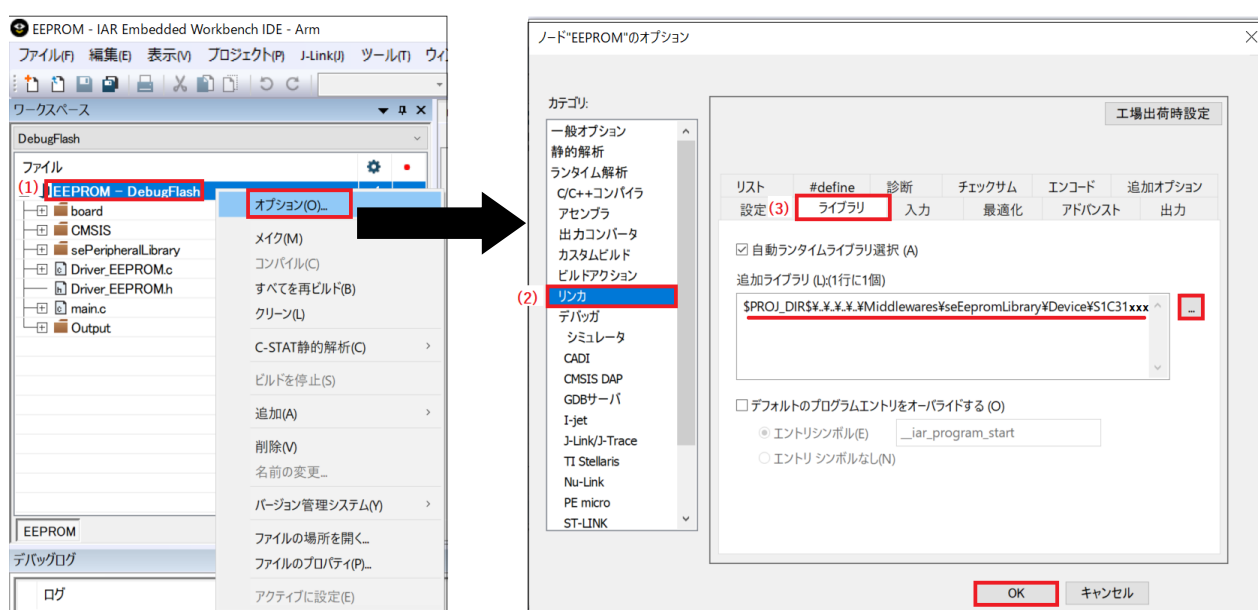


図 A.1 ライブラリの追加

- (4) IAR EWARM メニューの [プロジェクト] > [追加] > [ファイルの追加] を選択します。
- (5) 以下のドライバとドライバ定義を追加します。
 Drivers¥CMSIS¥Driver¥Source¥Driver_EEPROM.c
 Drivers¥CMSIS¥Driver¥Include¥Driver_EEPROM.h

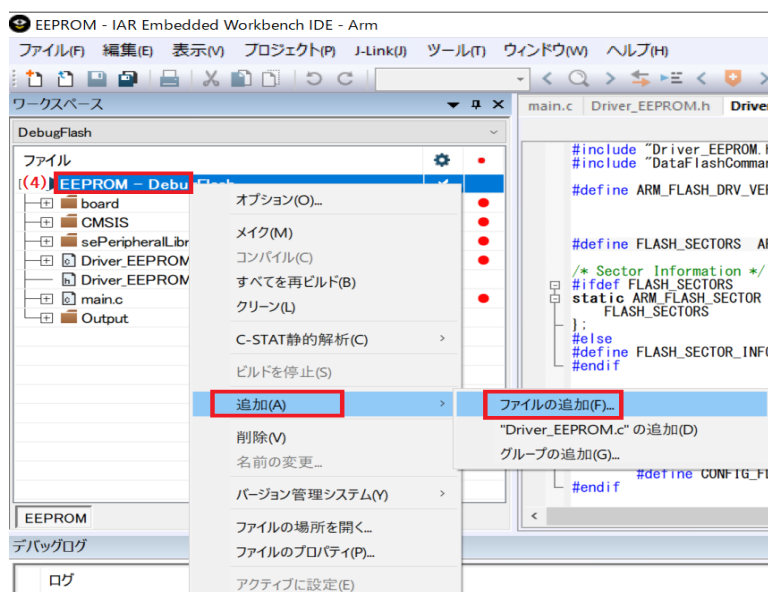


図 A.2 CMSISドライバの追加

2. インクルードパスの設定

- (1) IAR EWARM メニューの [プロジェクト] > [オプション] を選択します。
- (2) 表示されたダイアログの [カテゴリ] リストから [C/C++コンパイラ] を選択します。
- (3) [プリプロセッサ] タブから、“追加インクルードディレクトリ” にS1C31xxx周辺回路サンプルソフトウェアパッケージに同梱されているドライバ定義の以下のインクルードパスを追加します。
 - ・ Drivers¥CMSIS¥Driver¥Include
 - ・ Middlewares¥seEepromLibrary¥Include

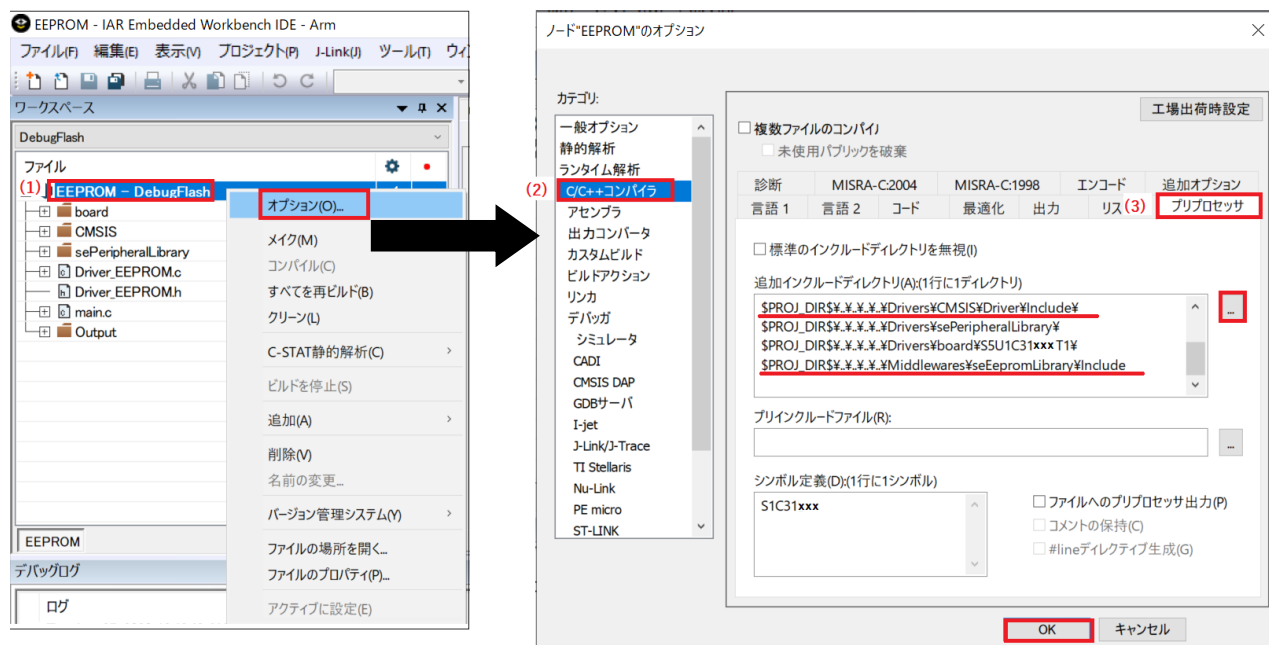


図 A.3 インクルードパスの設定

3. リンカスクリプト設定

- (1) プロジェクト内に同梱されているリンカスクリプトファイル (.icf) を編集します。
- (2) S1C31xxx周辺回路サンプルソフトウェアパッケージ同梱のサンプルソフトウェアのリンカスクリプトファイル (S1C31xxx_flash.icf) を参考に、以下のセクションを追加します。

...

```
initialize by copy { readwrite };
```

flash common text セクションの生成

```
initialize manually with packing = none { section .flash_common_text};
```

```
//initialize by copy with packing = none { section __DLIB_PERTHREAD }; // Required in a multi-threaded application
```

```
do not initialize { section .noinit };
```

```
place at address mem:__ICFEDIT_intvec_start__ { readonly section .intvec };
```

```
place in ROM_region { readonly };
```

```
place in RAM_region { readwrite, block CSTACK, block HEAP };
```

```
place in ROM_region { section .flash_common_text_init};
```

```
place in RAM_region { section .flash_common_text };
```

ROM 領域のコピー元セクションの指定

RAM 領域のコピー先セクションの指定

```
/"ROM2":place in ROM2_region { section EEPROM1 };
```

```
place at address mem: 0x20000 { readonly section EEPROM1 };
```

EEPROM エミュレーション領域として使用する内蔵フラッシュメモリの先頭アドレスを指定

上記を追加することで、本ライブラリ内で、本ライブラリのコードを RAM 領域に配置します。また、EEPROM エミュレーション領域として使用する内蔵フラッシュメモリを指定します。先頭アドレスは、リードミーに記載の単位で指定してください。

- (3) IAR EWARMメニューの [プロジェクト] > [オプション] を選択します。
- (4) 表示されたダイアログの [カテゴリ] リストから [リンカ] を選択します。
- (5) [設定] タブから、“デフォルトのオーバーライド” にチェックを入れ、編集したリンカスクリプトファイルを指定します。

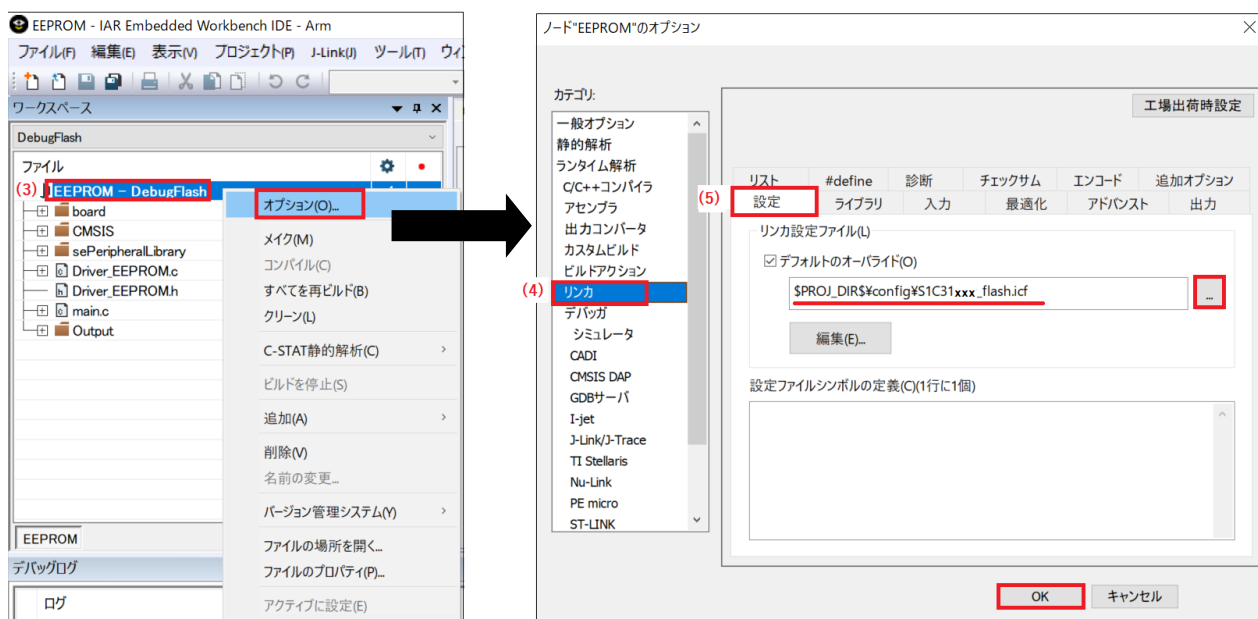


図 A.4 リンカスクリプトファイルの設定

B. ライブラリをプロジェクトへ組み込む方法(MDK-ARM)

統合開発環境 MDK-ARM (uVision) で作成したアプリケーションプログラムのプロジェクトに、S1C31 自己書き換えライブラリを組み込む方法を以下に記します。MDK-ARM の詳細については、付属のマニュアルをご参照ください。

1. ライブラリの追加

- (1) uVision の [Project] ウィンドウから対象のソースフォルダを右クリックし、[Add Existing Files to Group 'xxx'...] を選択します。
- (2) 表示されたダイアログから S1C31xxx 周辺回路サンプルソフトウェアパッケージの以下に同梱されている本ライブラリおよびドライバを追加します。
 - ・ Middlewares¥seEepromLibrary¥Device¥S1C31xxx¥seEepromLibraryS1C31xxx.lib
 - ・ Drivers¥CMSIS¥Driver¥Source¥Driver_EEPROM.c

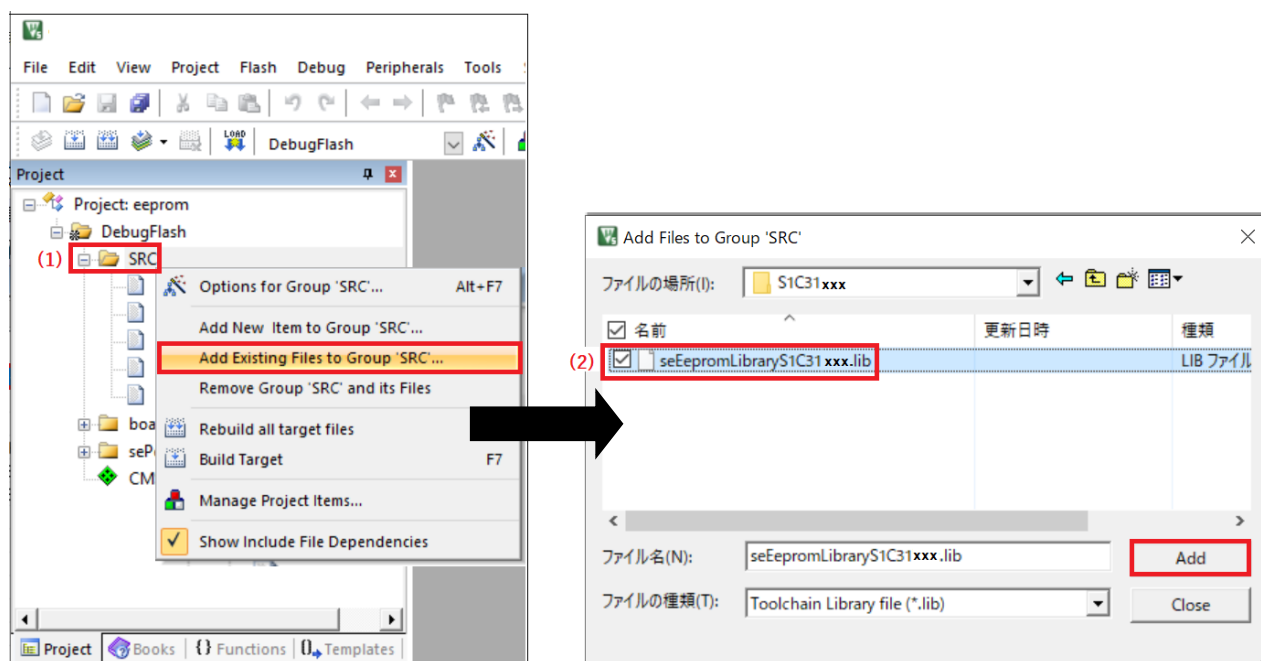


図 B.1 ライブラリの追加

2. インクルードパスの追加

- (1) uVision メニューの [Project] > [Options for Target 'xxx'...] を選択します。
- (2) 表示されたダイアログの [C/C++] > 'Include Paths' からフォルダを参照します。
- (3) [New (Insert)] から、S1C31xxx 周辺回路サンプルソフトウェアパッケージに同梱されているドライバ定義の以下のインクルードパスを追加します。
 - ・ Drivers¥CMSIS¥Driver¥Include
 - ・ Middlewares¥seEepromLibrary¥Include

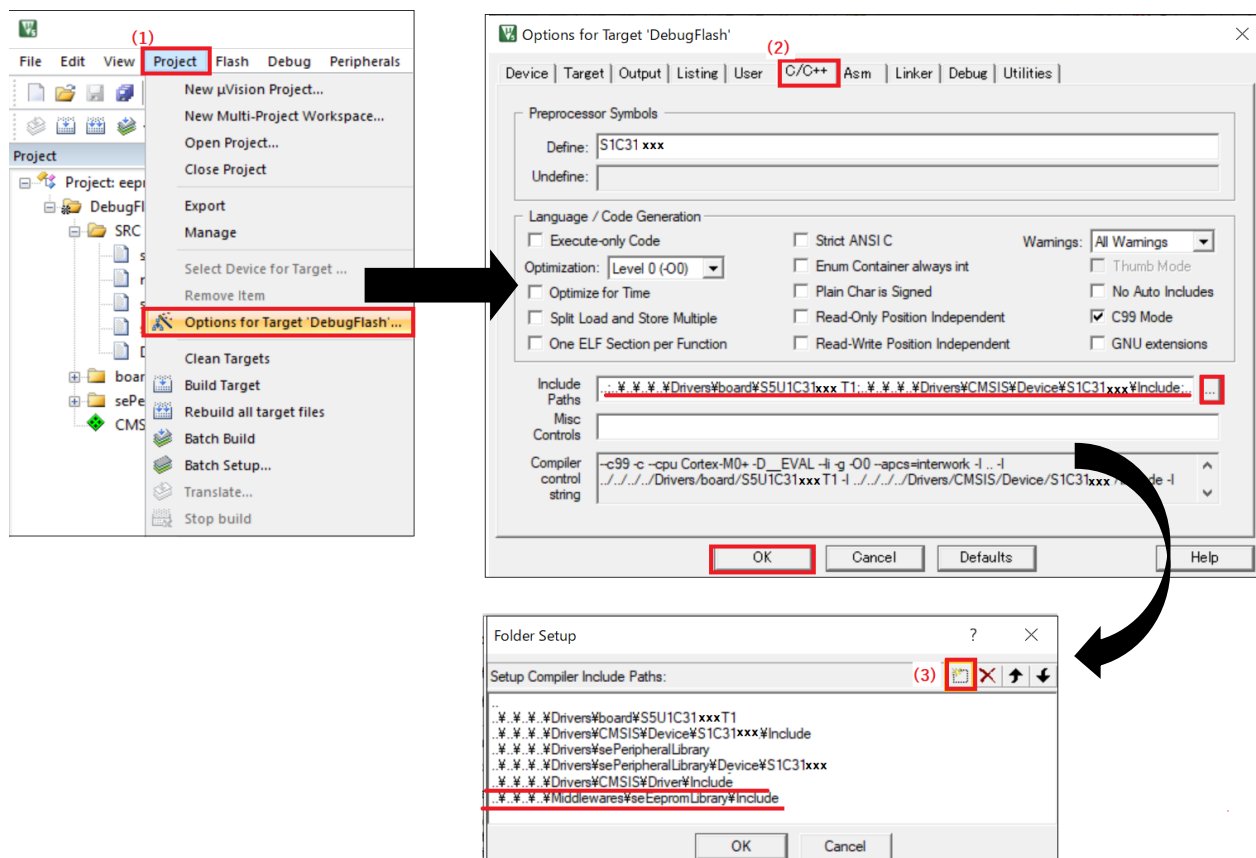


図 B.2 インクルードパスの設定

3. リンカスクリプトの設定

- (1) プロジェクト内に同梱されているリンカスクリプトファイル (.sct) を編集します。
- (2) S1C31xxx周辺回路サンプルソフトウェアパッケージ同梱のサンプルソフトウェアのリンカスクリプトファイル (eeprom_flash.sct) を参考に、以下のセクションを追加します。

```

...
LR_IROM1 0x00000000 0x00020000 {    ; load region size_region
...
RW_IRAM2 +0 {    ; RW data
*(.flash_common_text)
}
ARM_LIB_HEAP 0x00151800 EMPTY 0x400 ; Heap region growing up
{ }
ARM_LIB_STACK 0x00152000 EMPTY -0x400 ; Stack region growing down
{ }
}

LR_IROM2 0x00020000 0x00010000 {    ; load region size_region
ER_IROM2 0x00020000 0x00010000 {    ; load address = execution address
*(.eeprom)
}

```

RAM 領域に flash_common_text セクションを生成

EEPROM エミュレーション領域として使用する内蔵フラッシュメモリの先頭アドレスを指定

上記を追加することで、本ライブラリ内で、本ライブラリのコードを RAM 領域に配置します。また、EEPROM エミュレーション領域として使用する内蔵フラッシュメモリを指定します。先頭アドレスは、リードミーに記載の単位で指定してください。

- (3) uVisionメニューの [Project] > [Options for Target 'xxx'...] を選択します。

- (4) 表示されたダイアログの [Linker] > ‘Scatter File’ から編集したリンクスクリプトファイルを指定します。

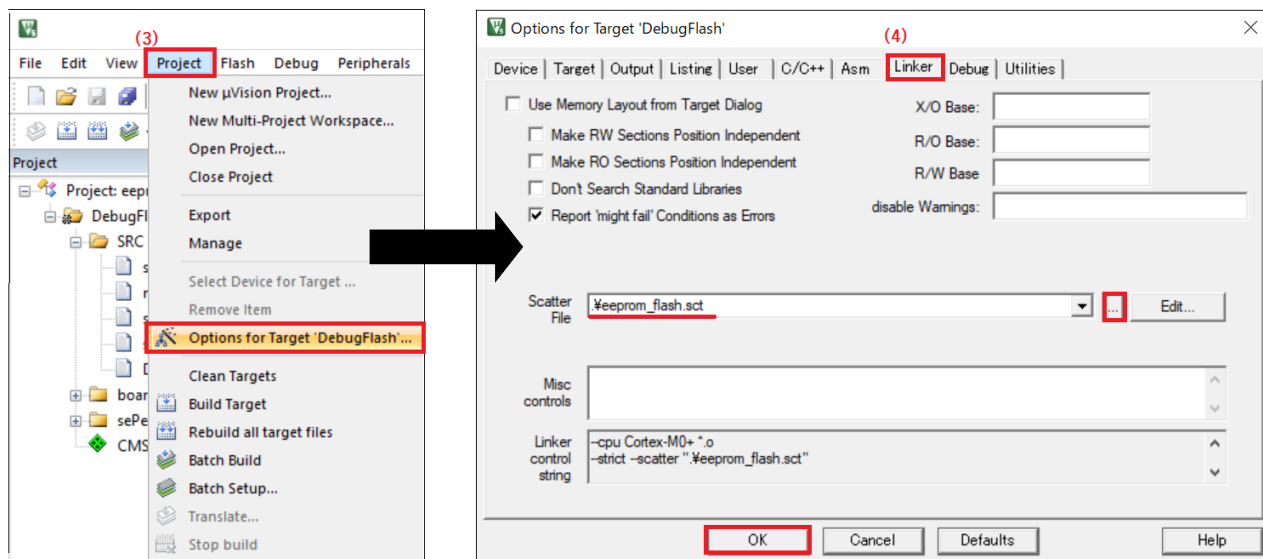


図 B.3 リンカスクリプトファイルの設定

[illegible]

セイコーエプソン株式会社
営業本部 デバイス営業部

東京 〒160-8801 東京都新宿区新宿 4-1-6 JR 新宿ミライナタワー29F

大阪 〒530-6122 大阪市北区中之島 3-3-23 中之島ダイビル 22F

ドキュメントコード : 414142000
2021 年 1 月 作成