EPSON
EXCEED YOUR VISION

S1C17 シリーズ
EEPROM エミュレーション
ライブラリ
説明書

セイコーエプソン株式会社

Rev.1.3

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。

要旨

本資料は、S1C17 シリーズ EEPROM エミュレーションライブラリにより、内蔵フラッシュメモリを用いて EEPROM エミュレーションを実現するための参考資料です。

動作環境

・PC
GNU17（S5U1C17001C）開発ツールインストール済み※
ICDmini USB ドライバインストール済み
・ICDmini（S5U1C17001H2, S5U1C17001H3）

PC との接続には USB ケーブルが必要です。
・ターゲットシステム(ユーザーターゲットボードもしくは弊社評価ボード)
・S1C17xxxEEPROM エミュレーションライブラリパッケージ（本パッケージ）

使用上の注意事項

本パッケージに同梱されているライブラリは、サンプルライブラリです。本ライブラリ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。

本資料は S1C17 シリーズの共通資料です。
本資料では (xxx) = 該当機種名となります。
EEPROM エミュレーションライブラリは機種毎に提供されます。
EEPROM エミュレーションライブラリが提供されている機種は、弊社 Web サイトを参照下さい。

目　次

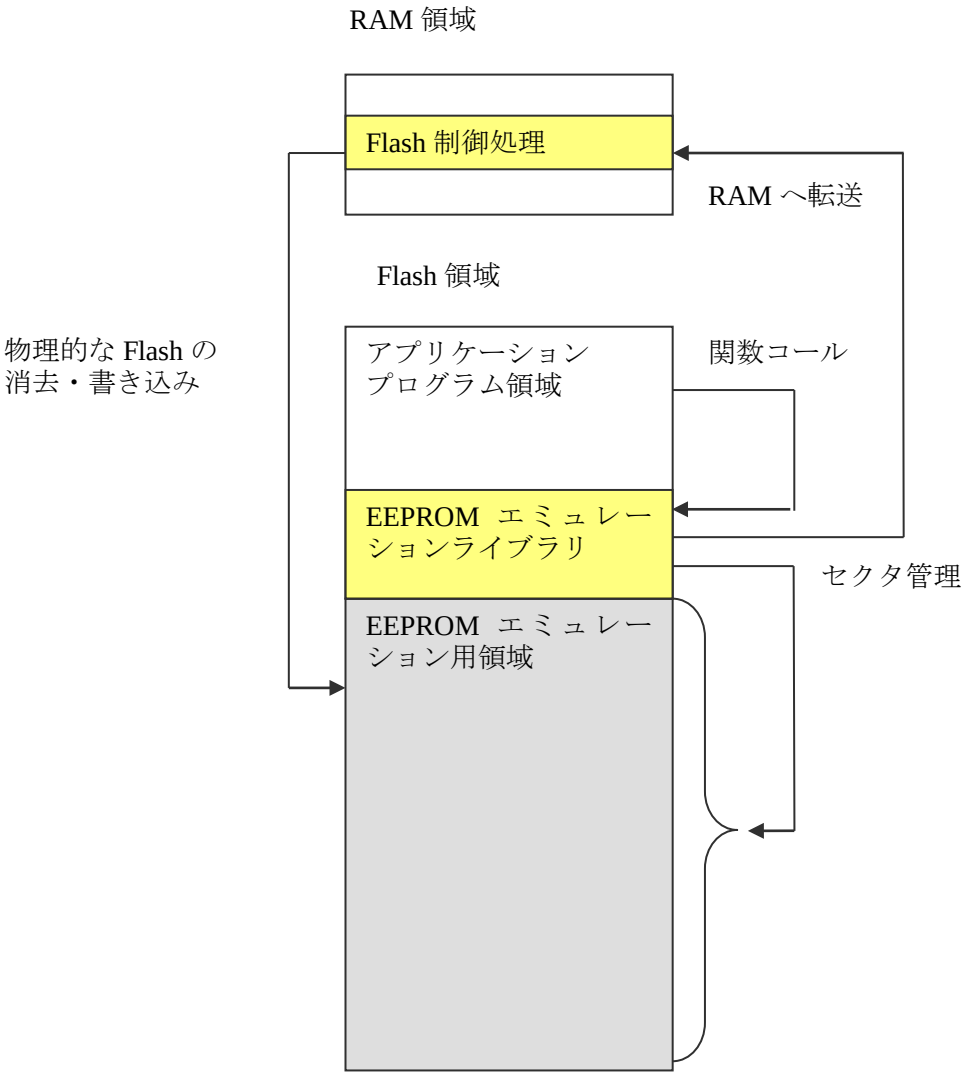
1. 概要

EEPROM エミュレーションライブラリパッケージは、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。

1.1 機能

本ライブラリは、S1C17 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。



1.2 フォルダ構成

本パッケージのフォルダ構成は以下の通りです。

```
+ s1c17(xxx)eeprom
    + eeprom                          : EEPROM エミュレーションライブラリ
    + s1c17(xxx)eeprom_gnu17v2        : GNU17 Ver.2.x 用サンプルプログラム
    + s1c17(xxx)eeprom_gnu17v3        : GNU17 Ver.3.x 用サンプルプログラム
    - s1c17(xxx)eeprom_j.pdf          : 説明書(日本語)
    - s1c17(xxx)eeprom_e.pdf          : 説明書(英語)
    - s1c17(xxx)eeprom_notes_j.txt    : 補足説明書(日本語)
    - s1c17(xxx)eeprom_notes_e.txt    : 補足説明書(英語)
    - License_e.txt                   : ソフトウェアライセンス契約書(英語)
```

1.3 ファイル構成

ライブラリのファイル構成は以下の通りです。

表 1 s1c17(xxx)eeprom / eeprom

ファイル名	機能
dataFlash17(xxx).a	S1C17(xxx)用 EEPROM 動作（フラッシュメモリ上で動作）
dataFlash17(xxx)ram.a	S1C17(xxx)用 EEPROM 動作（RAM 上で動作）
FlashControlErase.o	フラッシュメモリ消去
FlashControlWrite.o	フラッシュメモリ書き込み
DataFlashConfig.h	EEPROM 設定用ヘッダファイル
DataFlashConfig.c	EEPROM 設定用ソースファイル
DataFlashCommand.h	関数宣言用ヘッダファイル
OscControl.h	クロックソース設定用ヘッダファイル
OscControl.c	クロックソース設定用ソースファイル
FlashArea.s	EEPROM エミュレーション用領域設定ファイル

サンプルプログラムのファイル構成は以下の通りです。

表 2 s1c17(xxx)eeprom / s1c17(xxx)eeprom_gnu17vx

ファイル / フォルダ名	機能
eeprom	EEPROM エミュレーションライブラリ(フォルダ)
boot.c	boot プログラム
main.c	main プログラム

2. ライブラリの使い方

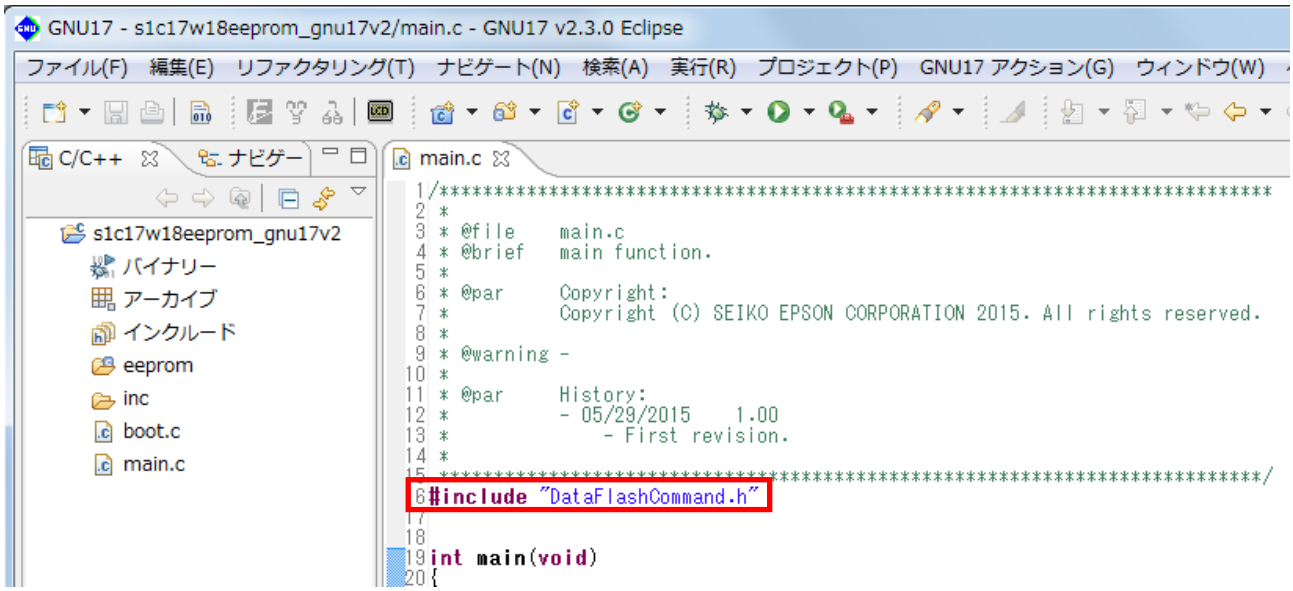
本ライブラリを使用するにあたり、必要な対応事項、並びに注意事項を説明します。また、本ライブラリを使用したサンプルプログラムについて説明します。

2.1 アプリケーションプログラムへの適用方法

以下では、本ライブラリを使用するアプリケーションプログラムのソースファイルに必要になる対応事項を説明します。
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込む方法については Appendix A. ライブラリをプロジェクトへ組み込む方法を参照下さい。

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内に、“DataFlashCommand.h”をインクルード宣言します。
注）インクルードパスを設定していない場合は、パスを指定してインクルードしてください。



2. EEPROM サイズ、書き込みリトライ回数の設定

“DataFlashConfig.h”に以下の値を設定して下さい。
エミュレートする EEPROM のサイズを“CONFIG_EEPROM_SIZE_MAX”に設定します。設定可能なサイズについては、補足説明書 s1c17(xxx)eeprom_notes_j.txt を参照してください。
書き込みに失敗した場合に、リトライを行う回数を“CONFIG_RETRY_COUNT”に設定します。
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。

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

3. クロックソースの設定

本ライブラリの書き込み関数内で CPU クロックと T16 ch.0 の設定をフラッシュメモリの書き込みに最適になるように変更しています。"OscControl.c"の関数“OscClockSourceInitialize()”、“OscClockSourceFinalize()”を書き換えることによって設定の変更ができます。
“OscClockSourceInitialize()”では書き込み時の CPU クロックと T16 ch.0 の設定を行っています。本関数を書き換える場合は以下の点に注意してください。

・デフォルトでは最適なクロック（4MHz）で動作するように設定されています。
・クロックを変更する場合は 4MHz 以下になるように分周比を設定してください。
・CPU クロックソースと T16 ch.0 のクロックソースは同一になるようにしてください。

“OscClockSourceFinalize()”は書き込み終了時に呼び出されます。本関数を修正して、CPU のクロックと T16 ch.0 の設定を本ライブラリ使用前に戻すことができます。

```
void OscClockSourceInitialize(void)
{
        /// It doesn't do at all when having already started.
        if ( CLGSCLK_CLKSRC != 2 )
        {
                /// Disable write-protect.
                MSCPROT = 0x96;
                CLGOSC_OSC3EN = 0;    /// Stop OSC3.
                /// Clear interrupt flag(CLGINTF.OSC3STAIF).
                CLGINTF = 0x0004;
                //OSC3=Internal
                CLGOSC3_OSC3MD = 0;
                //OSC3=4MHz
                CLGOSC3_OSC3FQ = 3;
                //OSC3 enable
                CLGOSC_OSC3EN = 1;
                while ( CLGINTF_OSC3STAIF == 0 ) {
                        asm("nop");    // wait ...
                }
                //Clock=OSC3
                CLGSCLK_CLKSRC = 2;
        }
        //T16 setting
        T16_0CLK = 0x0112;    // T16 Debug mode run,Div=1/2,Clock=OSC3 4MHz
        T16_0CTL = 1;         // T16 enable
        T16_0MOD = 1;         // One shot mode
}

void OscClockSourceFinalize(void)
{
        return;
        MSCPROT = 0x96;
        /// Clear interrupt flag(CLGINTF.IOSCSTAIF).
        CLGINTF  = 0x0001;
        CLGOSC_IOSCEN = 1;           /// Start oscillation.
        //Clock=IOSC
        CLGSCLK_CLKSRC = 0;
        //OSC3 disable
        CLGOSC_OSC3EN = 0;
        //T16 setting
        T16_0CTL = 0;                // T16 disenable
        MSCPROT = 0x0;
}
```

4. EEPROM 読み出し／書き込み関数の追加

EEPROM の読み出し／書き込みを実行する、本ライブラリの関数をアプリケーションプログラムのソースコードに追加します。
関数の仕様については 03. ライブラリ仕様 を参照してください。

```
for(i=0;i<CONFIG_EEPROM_SIZE_MAX;i++)
{
        if(DataFlashWrite(i,i)!=DATAFLASH_SUCCESS)
        {
                asm("nop");
        }
        testdata[i] = DataFlashRead(i);
}
//compare
for(i=0;i<CONFIG_EEPROM_SIZE_MAX;i++)
{
        if(testdata[i] != (i&0xff))
        {
                asm("nop");
        }
}
```

2.2 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、フラッシュメモリ領域を使用します。
機種別の使用量については、補足説明書 s1c17(xxx)eeprom_notes_j.txt をご参照ください。

2.3 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、S1C17W18 で CPU のクロックソースとして内蔵 OSC3（4MHz）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時の書き込み時間を以下に示します。

Typ.値 7ms
Max.値 43ms

実際の書き込み時間は、本ライブラリを使用するターゲットシステムで確認して下さい。

2.4 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。

- 16 ビットタイマ（T16）の ch.0 をフラッシュメモリの書き込みタイミング制御に使用しています。
- 書き込み関数実行時に CPU クロックと T16 ch.0 の設定を変更しています。詳細は 2.1 章の「3. クロックソースの設定」を参照してください。
- 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 * 128）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 を設定して下さい。
- 本ライブラリ使用時は Vpp 端子はオープンとしてください。Vpp 端子が接続されたままだと、フラッシュメモリの消去／書き込みに失敗する場合があります。
- 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“S1C17(xxx)テクニカルマニュアル”を参照してください。
- 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 V_{DD} 動作電圧(V_{PP} 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。

2.5 サンプルプログラム

1. サンプルプログラム仕様

サンプルプログラムでは、本ライブラリを使用して以下の動作を行います。

- 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んだ後に、データを読みだして比較を行います。

2. 準備

IDE でサンプルプログラムを実行するには、以下の手順を参考にしてください。また、ライブラリの使用にあたっては、上記 2.1-2.4 の記載事項にも留意してください。

① プロジェクトのインポート
IDE を起動して、サンプルプログラムをインポートしてください。
② ビルド
IDE を使用してサンプルプログラムをビルドしてください。
③ 接続
ICDmini、ターゲットシステムを PC と接続してください。
④ Flashセキュリティの解除
Flashセキュリティ対応済みのICでサンプルプログラムをデバッグする場合、Flashセキュリティを解除してください。
⑤ プログラムロード
IDE を使用してサンプルプログラムをロードしてください。
⑥ 実行
ターゲットシステムをリセットするなどして、プログラムを実行させてください。

詳細は、“S1C17(xxx)テクニカルマニュアル”、”S5U1C17001C マニュアル” 及び "S5U1C17001H User Manual(ICDmini)" を参照してください。

3. 動作概要

① EEPROMアドレスの初期値 = 0、書き込みデータの初期値 = 0とします。
② EEPROM書き込み関数でデータを書き込みます（main.c / DataFlashWrite）。
③ 書き込んだアドレスからデータを読み出します（main.c / DataFlashRead）。
④ アドレスが CONFIG_EEPROM_SIZE_MAX より小さければアドレスと書き込みデータを 1 インクリメントして②に戻ります。
⑤ 読み出しデータと書き込みデータを比較します。

DataFlashRead, DataFlashWrite 関数については、3.1 EEPROM 読み出し／書き込み関数詳細 を参照してください。

3. ライブラリ仕様

3.1 EEPROM 読み出し／書き込み関数詳細

本ライブラリに記述された関数について記します。

EEPROM 書き込み

関数名		
DataFlashWrite(unsigned short address, unsigned char data)		
引数		
address	unsigned short	EEPROM アドレスを指定する。
data	unsigned char	書き込みデータ。
戻り値		
int	書き込み結果（エラーコード）を表す。	
機能		
引数で指定されたパラメータに従って、データ書き込みを行う。 ① 引数が正しいかチェックする。 ② 指定したアドレスにデータを書き込む。 ③ 戻り値を返す。		
備考		
第 1 引数の有効範囲は、0 ～ (CONFIG_EEPROM_SIZE_MAX - 1)です。		

EEPROM 読み出し

関数名		
DataFlashRead(unsigned short address)		
引数		
address	unsigned short	EEPROM アドレスを指定する。
戻り値		
unsigned char	読み出しデータです。	
機能		
引数で指定されたパラメータに従って、データ読み出しを行う。 ① 引数が正しいかチェックする。 ② 指定したアドレスからデータを読み出す。 ③ 戻り値を返す。		
備考		
第 1 引数の有効範囲は、0 ～ (CONFIG_EEPROM_SIZE_MAX - 1)です。 書き込み未実施のアドレスからは 0xFF が読み出されます。		

EEPROM シーケンシャル読み出し

関数名		
DataFlashReadCurrent(void)		
引数		
なし		
戻り値		
unsigned char	読み出しデータです。	
機能		
現在アドレスからデータ読み出しを行う。 ① 現在アドレスからデータを読み出す。 - 読み込み実施後、現在アドレスを 1 インクリメントします。 - 最終アドレスを読み込み後、現在アドレスは 0 になります。 - DataFlashWrite 関数実行後にコールされた場合は、DataFlashWrite 関数で指定したアドレスから読み出しを行います。 - DataFlashRead 関数実行後にコールされた場合は、DataFlashRead 関数で指定したアドレスの次のアドレスから読み出しを行います。 ② 戻り値を返す。		
備考		
現在アドレスの初期値は 0 になります。 書き込み未実施のアドレスからは 0xFF が読み出されます。		

3.2 エラーコード定義

表 3 エラーコード

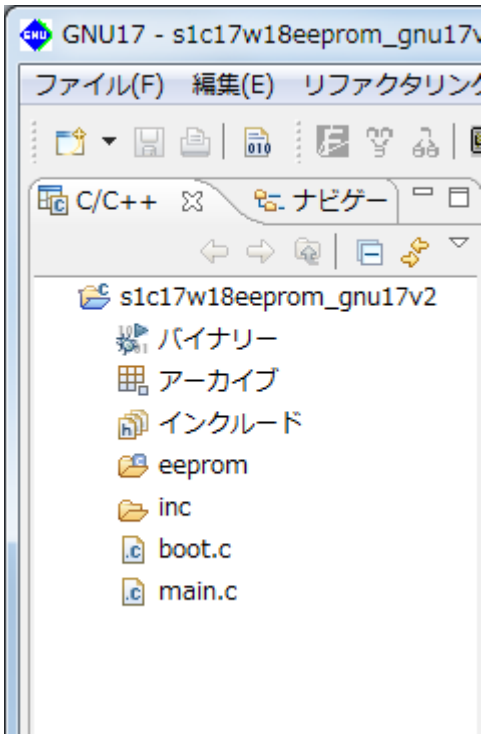
定義名	値	説明
DATAFLASH_SUCCESS	0	書き込み正常終了
DATAFLASH_ERROR_ERASE	1	消去エラー
DATAFLASH_ERROR_WRITE	2	書き込みエラー
DATAFLASH_ERROR_PARAMETER	3	パラメーターエラー

Appendix

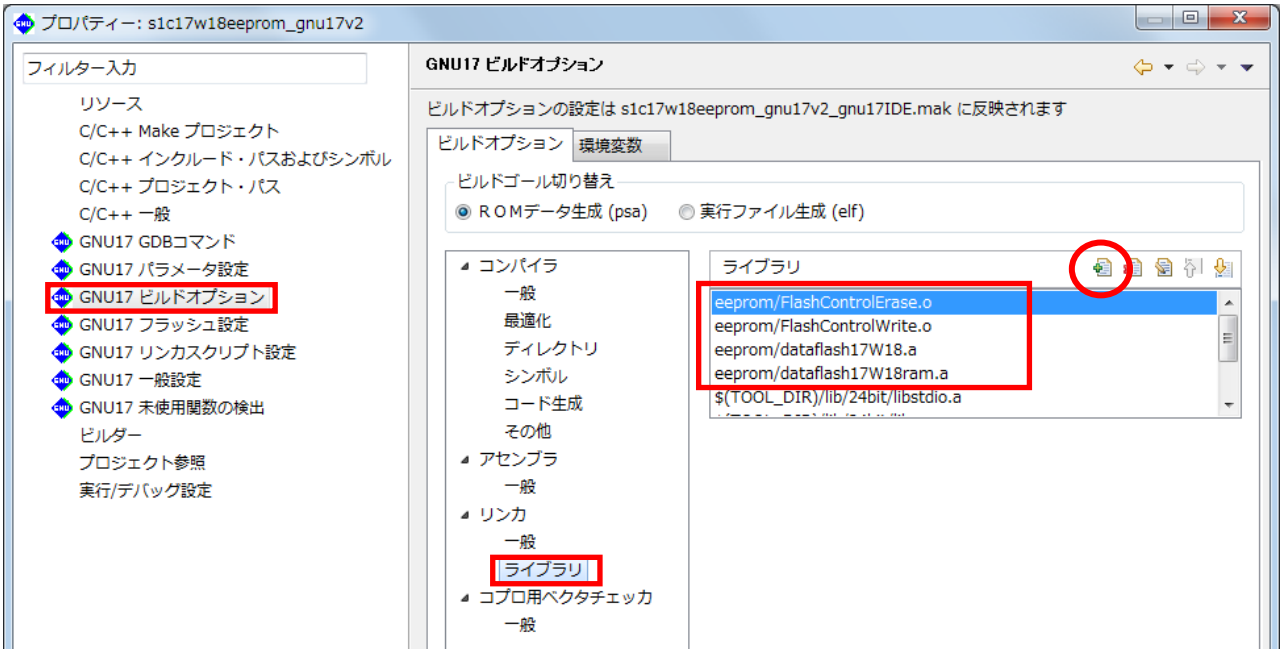
A. ライブラリをプロジェクトへ組み込む方法(GNU17 Ver.2.x)

GNU17 Ver.2.x による本パッケージのライブラリの使用方法を以下に記します。
（以下では S1C17W18 を例としています)
GNU17 Ver.2.x の詳しい使い方については、コンパイラマニュアルを参照してください。

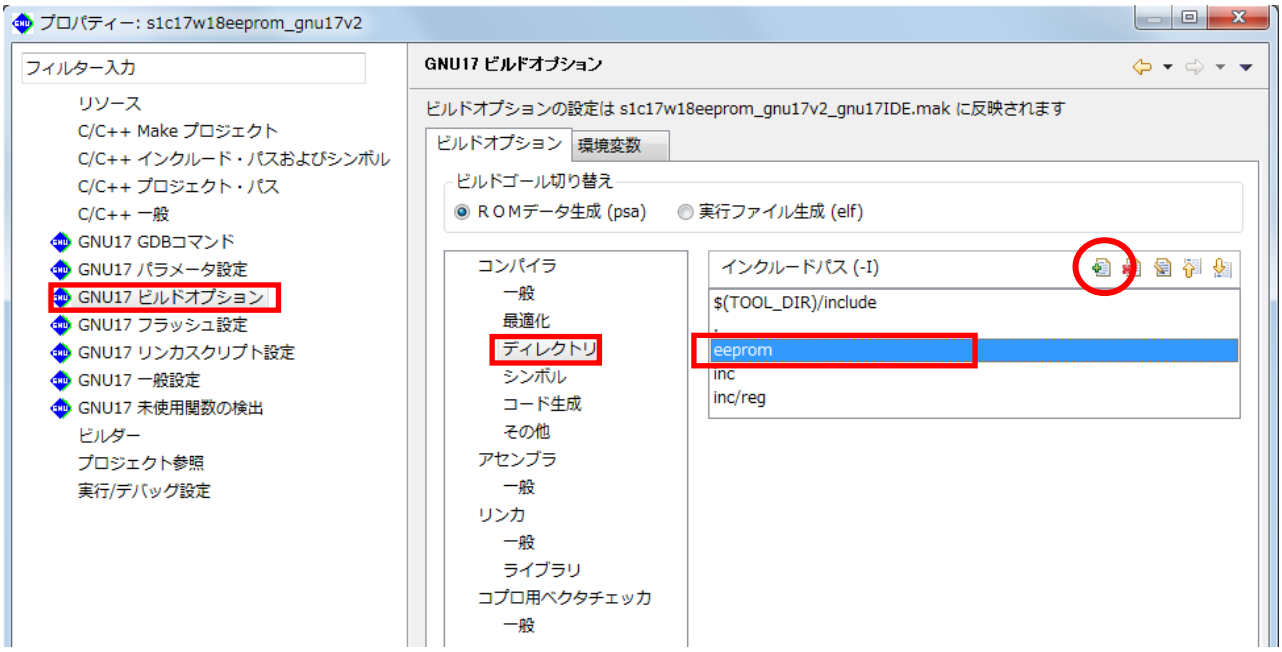
1. ライブラリ、ヘッダファイルの追加
   パッケージ内の eeprom フォルダをプロジェクトフォルダにインポートしてください。



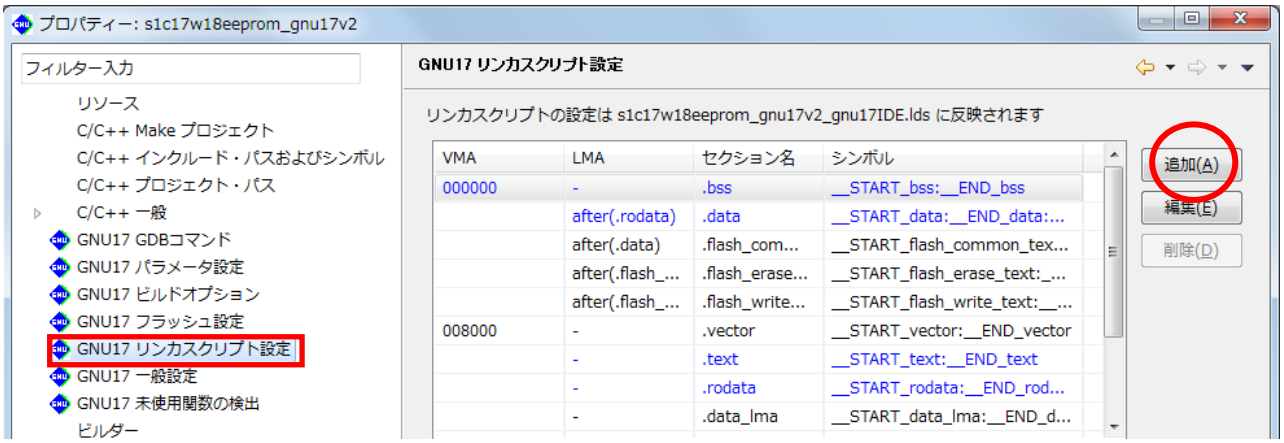
2. ライブラリ設定
インポートしたライブラリを使用するために、ライブラリ設定に追加します。
プロジェクトの[プロパティー]-[GNU17 ビルドオプション]-[リンカ]-[ライブラリ]から下図の赤丸を選び、eeprom フォルダ内の “dataflash17W18.a”、“dataflash17W18ram.a” 、“FlashControlErase.o” 、“FlashControlWrite.o”を選択し、追加します。



3. インクルードパス設定
eeprom フォルダにある”DataFlashCommand.h”を使用するために、インクルードパスを設定します。プロジェクトの[プロパティー]-[GNU17 ビルドオプション]-[ディレクトリ]から下図の赤丸を選び、プロジェクトに eeprom フォルダへのインクルードパスを設定します。
注）ソースファイル内で直接インクルードパスを指定する場合は必要ありません。

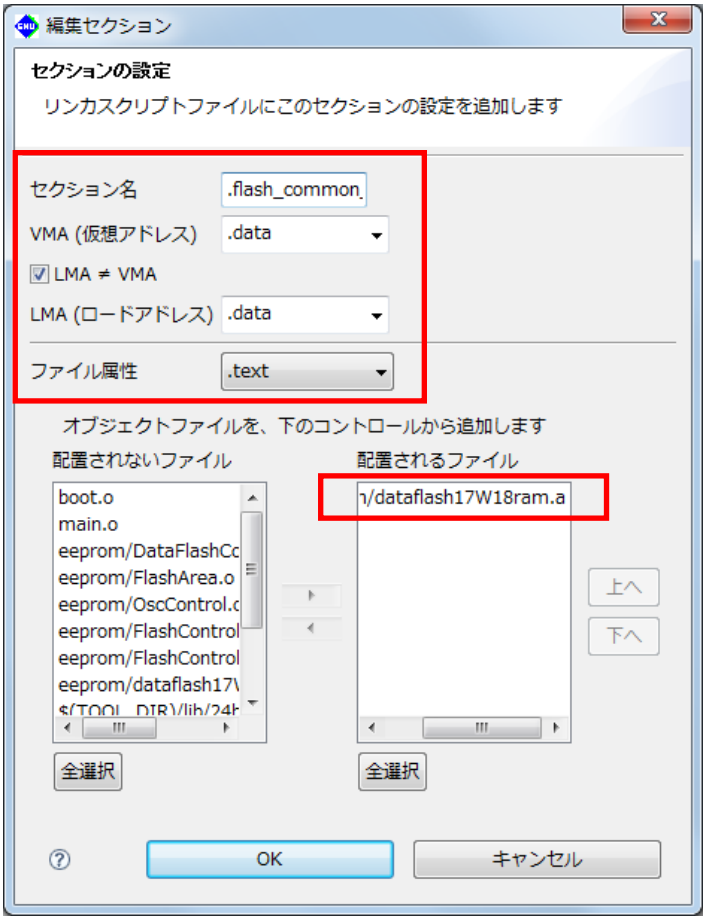


4. リンカスクリプト設定
インポートしたライブラリのリンカスクリプト設定をします。
プロジェクトの[プロパティー]-[GNU17 リンカスクリプト設定]から下図の赤丸を選び、ライブラリを配置するセクションを追加します。

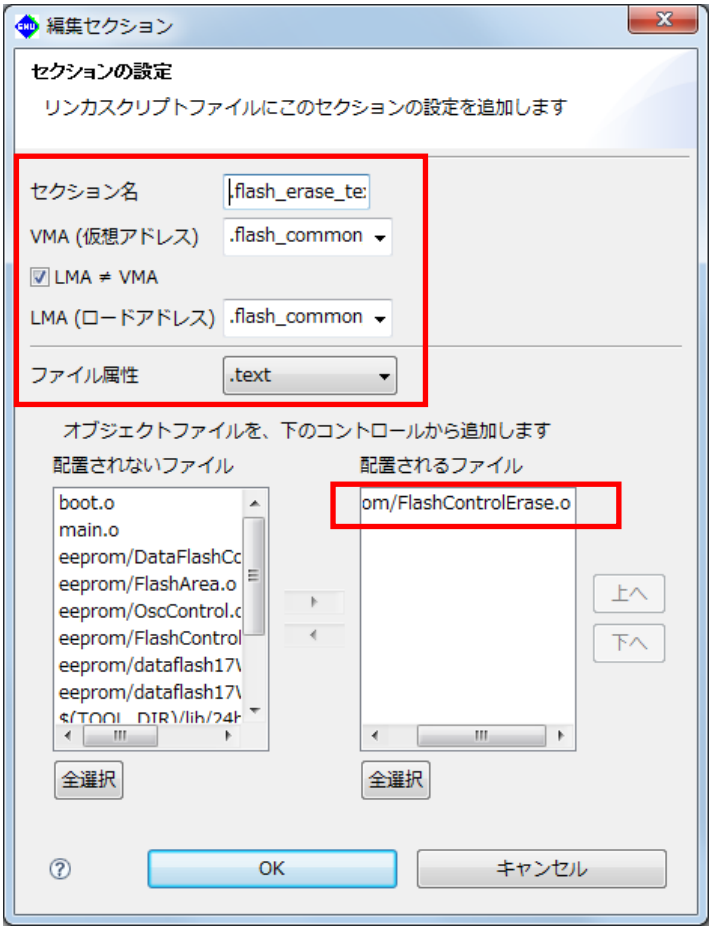


追加するのは、“.flash_common_text”、“.flash_erase_text”、“.flash_write_text”、“.flashdata_address”の各セクションです。先頭に”.”のついた上記の名前とし、下図に従って追加してください。

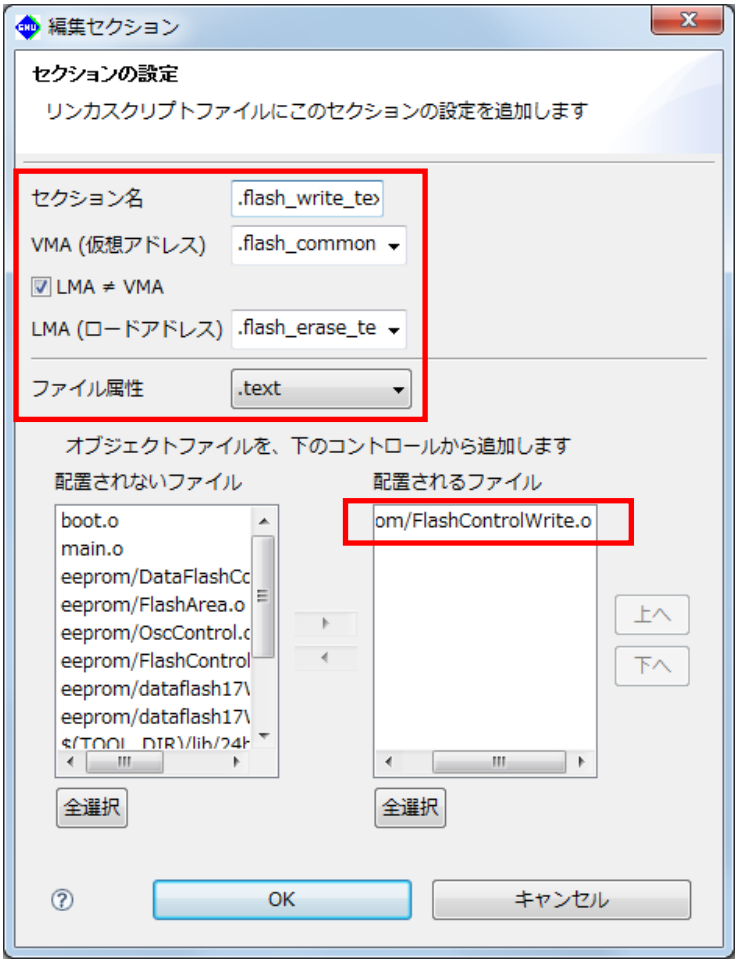
“.flash_common_text”セクションに“dataflashW18ram.a”を配置します。



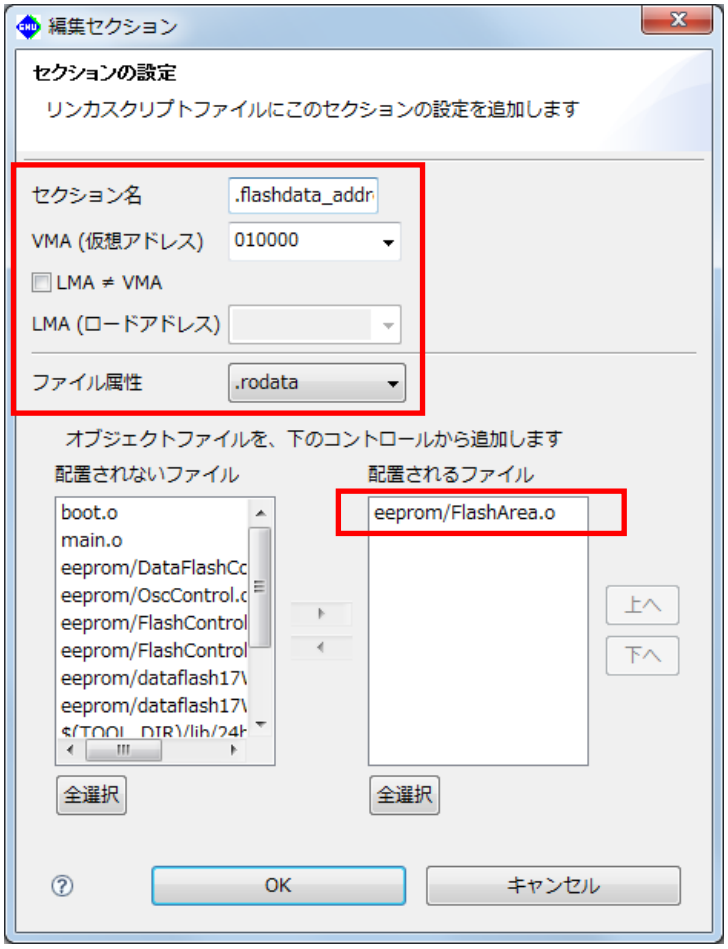
“.flash_erase_text”セクションに“FlashControlErase.o”を配置します。
VMA、LMA は“.flash_common_text”とします。



“.flash_write_text”セクションに“FlashControlWrite.o”を配置します。
VMA は“.flash_common_text”、LMA は“.flash_erase_text”とします。



“.flashdata_address”セクションに“FlashArea.o”を配置します。
VMA に EEPROM エミュレーションで使用するフラッシュメモリの先頭アドレスを指定します。
アドレスは 0x80 単位で指定して下さい。



B. ライブラリをプロジェクトへ組み込む方法(GNU17 Ver.3.x)

GNU17 Ver.3.x による本パッケージのライブラリの使用方法を以下に記します。
GNU17 Ver.3.x の詳しい使い方については、コンパイラマニュアルを参照してください。

1. ライブラリ、ヘッダファイルの追加
プロジェクト中のフォルダ src にパッケージ内の eeprom フォルダをインポートしてください。

2. ライブラリ設定
インポートしたライブラリを使用するために、ライブラリ設定に追加します。
プロジェクトの[Properties]-[C/C++ Build]-[Environment]の Variable GCC17_USER_LIBS の Value に、src¥eeprom フォルダ内の“dataflash17W18.a”、“dataflash17W18ram.a”、“FlashControlErase.o”、“FlashControlWrite.o”を追加します。

```
..¥src¥eeprom¥ dataflash17W18.a;..¥src¥ eeprom¥ dataflash17W18ram.a;
..¥src¥ eeprom¥ FlashControlErase.o;..¥src¥ eeprom¥ FlashControlWrite.o
```

3. インクルードパス設定
eeprom フォルダにある“DataFlashCommand.h”を使用するために、インクルードパスを設定します。
プロジェクトの[Properties]-[C/C++ Build]-[Settings]-[Tool Settings]-[Cross GCC Compiler]-[Includes]を選択し、src¥eeprom フォルダへのインクルードパスを設定します。

```
"../src/eeprom"
```

4. リンカスクリプト設定
ライブラリのリンカスクリプト設定をします。
以下のフォルダに EEPROM エミュレーションライブラリ用に記述したリンカスクリプトがありますので、プロジェクトフォルダへコピーしてください。

```
¥c17(xxx)_sample_gnu17v3¥eeprom.x
```

[Properties]-[C/C++ Build]-[Settings]-[Tool Settings]-[Cross GCC Linker]-[Miscellaneous]を選択し、[Other options]にコピーしたリンカスクリプトファイルを指定します。

```
-T ..¥eeprom.x
```

このリンカスクリプトでは、ライブラリの動作に必要な以下のシンボルを定義して、ライブラリの実行アドレスを内蔵 RAM に配置しています。

```
__START_flash_common_text_lma
__START_flash_erase_text_lma
__START_flash_write_text_lma
```

また、以下の記述により、“FlashControlCommon.o”、“FlashControlWrite.o”、“FlashControlErase.o”を ROM に配置しないように設定しています。

```
*(EXCLUDE_FILE(*FlashTimeTable*.o*FlashControlCommon.o*FlashControlWrite.o
                *FlashControlErase.o) .text)
```

“.flashdata_address”で EEPROM エミュレーションで使用するフラッシュメモリの先頭アドレスを指定します。アドレスは 0x80 単位で指定して下さい。

```
.flashdata_address    0x010000 :
```

改訂履歴表

付－1

Rev. No.	日付	ページ	種別	改訂内容（旧内容を含む） および改訂理由
Rev 1.0	2015/11/16	全ページ	新規	新規制定
Rev.1.1	2017/04/07	P1,P5, P15,P16	変更	S1C17M30/M31/M32/M34 対応に伴う変更。
Rev.1.2	2018/07/01	P1,P5, P15,P16	変更	S1C17M30/M31/M32/M34 対応は別マニュアルにするため変更は戻す。
		P3	変更	「2. EEPROM サイズ、書き込みリトライ回数の設定」の記述を変更。
		P4	変更 追加	「3. クロックソースの設定」の記述を変更、追加。
		P5	変更 追加	「4. EEPROM 読み出し／書き込み関数の追加」内のソースコードを変更。 「2.4 ライブラリ使用上の注意」の記述を変更、追加。
Rev.1.3	2018/10/18	P5	追加	「2.4 ライブラリ使用上の注意」の記述を追加。

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　〒191-8501　東京都日野市日野 421-8
TEL（042）587-5313（直通）　FAX（042）587-5116

大阪　〒530-6122　大阪市北区中之島 3-3-23　中之島ダイビル 22F
TEL（06）7711-6770（代表）　FAX（06）7711-6771

ドキュメントコード：413120103
2015 年 11 月 作成
2018 年 10 月 改訂