

S1C17 シリーズ
EEPROM エミュレーション
ライブラリ
説明書

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ください。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ください。
3. 本評価ボード・キット、開発ツールに用いられる部品は、予告な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、予告な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します。
2. 弊社製品のご購入およびご使用にあたりましては、事前に弊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、弊社ホームページなどを通じて公開される最新情報に常にご注意ください。
3. 本資料に掲載されている応用回路、プログラム、使用方法などはあくまでも参考情報です。お客様の機器・システムの設計において、応用回路、プログラム、使用方法など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ならびに損害の発生に対し、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。
4. 弊社は常に品質、信頼性の向上に努めていますが、一般的に半導体製品は誤作動または故障する場合があります。弊社製品のご使用にあたりましては、弊社製品の誤作動や故障により生命・身体に危害を及ぼすこと又は財産が侵害されることのないように、お客様の責任において、お客様のハードウェア、ソフトウェア、システムに必要な安全設計を行うようお願いいたします。なお、設計および使用に際しては、弊社製品に関する最新の情報(本資料、仕様書、データシート、マニュアル、弊社ホームページなど)をご確認いただき、それに従ってください。また、上記資料などに掲載されている製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を使用する場合は、お客様の製品単独およびシステム全体で十分に評価を行い、お客様の責任において適用可否の判断をお願いいたします。
5. 弊社は、正確さを期すために慎重に本資料およびプログラムを作成しておりますが、本資料およびプログラムに掲載されている情報に誤りがないことを保証するものではありません。万一、本資料およびプログラムに掲載されている情報の誤りによってお客様に損害が生じた場合においても、弊社は一切その責任を負いかねます。
6. 弊社製品の分解、解析、リバースエンジニアリング、改造、改変、翻案、複製などは堅くお断りします。
7. 弊社製品は、一般的な電子機器(事務機器、通信機器、計測機器、家電製品など)および本資料に個別に掲載されている用途に使用されることを意図して設計、開発、製造されています(一般用途)。特別な品質、信頼性が要求され、その誤動作や故障により生命・身体に危害を及ぼす恐れ、膨大な財産侵害を引き起こす恐れ、もしくは社会に深刻な影響を及ぼす恐れのある以下の特定用途に使用されることを意図していません。お客様に置かれましては、弊社製品を一般用途に使用されることを推奨いたします。もし一般用途以外の用途で弊社製品のご使用およびご購入を希望される場合、弊社はおお客様の特定用途に弊社製品を使用されることへの商品性、適合性、安全性について、明示的・黙示的に関わらずいかなる保証を行うものではありません。お客様が特定用途での弊社製品の使用を希望される場合は、弊社営業窓口まで事前にご連絡の上、承諾を得てください。

【特定用途】

宇宙機器(人工衛星・ロケットなど) / 輸送車両並びにその制御機器(自動車・航空機・列車・船舶など)
医療機器(本資料に個別に掲載されている用途を除く) / 海底中継機器 / 発電所制御機器 / 防災・防犯装置
交通用機器 / 金融関連機器
上記と同等の信頼性を必要とする用途

8. 本資料に掲載されている弊社製品および当該技術を国内外の法令および規制により製造・使用・販売が禁止されている機器・システムに使用することはできません。また、弊社製品および当該技術を大量破壊兵器等の開発および軍事利用の目的その他軍事用途等に使用しないでください。弊社製品または当該技術を輸出または海外に提供する場合は、「外国為替及び外国為替法」、「米国輸出管理規則(EAR)」、その他輸出関連法令を遵守し、係る法令の定めるところにより必要な手続きを行ってください。
9. お客様が本資料に掲載されている諸条件に反したことに起因して生じたいかなる損害(直接・間接を問わず)に関して、弊社は一切その責任を負いかねます。
10. お客様が弊社製品を第三者に譲渡、貸与などをしたことにより、損害が発生した場合、弊社は一切その責任を負いかねます。
11. 本資料についての詳細に関するお問合せ、その他お気付きの点などがありましたら、弊社営業窓口までご連絡ください。
12. 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。

(Rev.J1.0, 2021.9)

©Seiko Epson Corporation 2022, All rights reserved.

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	1
1.1 機能	1
1.2 フォルダ構成.....	2
1.3 ファイル構成.....	2
2. ライブラリの使い方	3
2.1 アプリケーションプログラムへの適用方法	3
2.2 内蔵 RAM、フラッシュメモリ使用量	6
2.3 書き込み時間.....	6
2.4 ライブラリ使用上の注意	6
2.5 サンプルプログラム.....	7
3. ライブラリ仕様	8
3.1 ライブラリ関数の詳細	8
3.2 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(GNU17 Ver.2.x).....	11
B. ライブラリをプロジェクトへ組み込む方法(GNU17 Ver.3.x).....	18
改訂履歴表.....	21

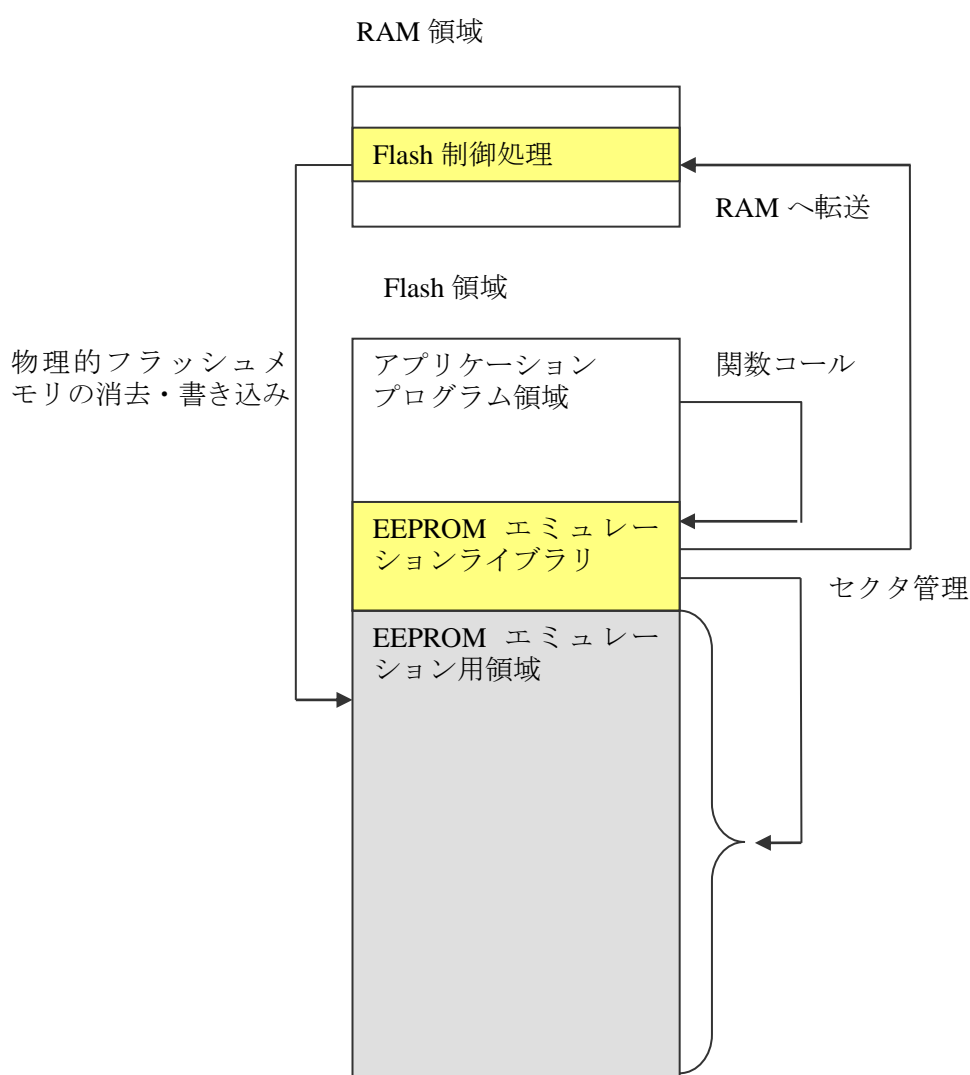
1. 概要

S1C17 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。2K バイト～128K バイトの EEPROM エミュレーション用領域を使用して、16～512 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. 概要

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_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 1.1 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 1.2 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

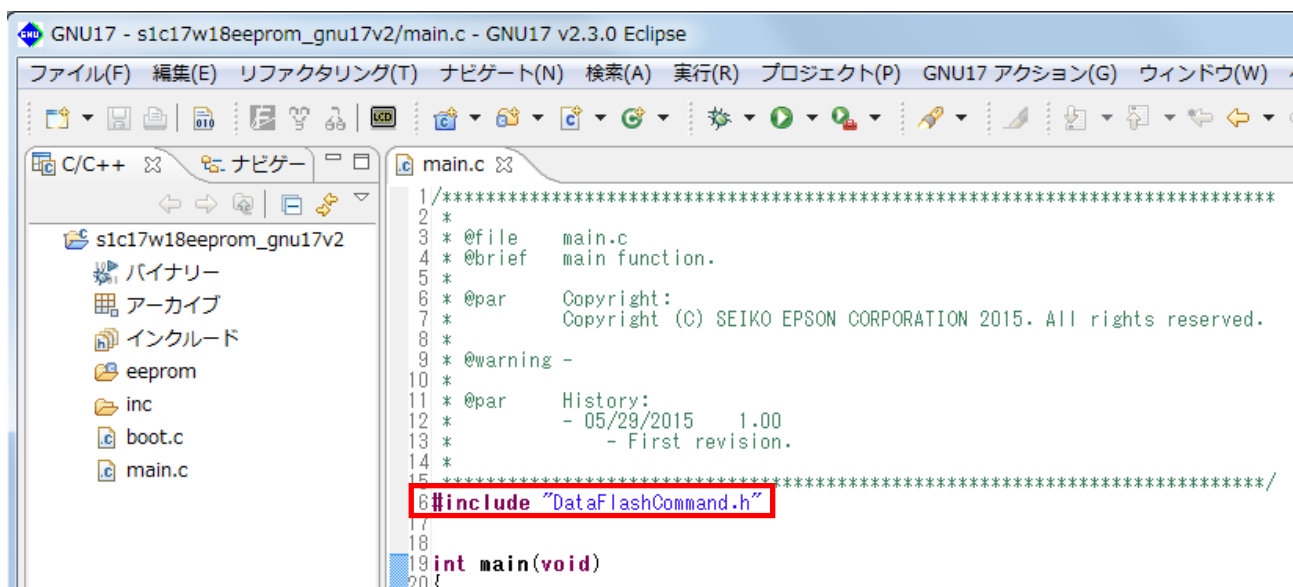
S1C17 EEPROM エミュレーションライブラリおよびサンプルソフトウェアの使用方法について説明します。

2.1 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内に、“DataFlashCommand.h”をインクルード宣言します。
ご注意：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。

```

#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（バイト）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)eprom_notes_j.txt を参照してください。

● 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“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。

2. ライブラリの使い方

3. クロックソースの設定

本ライブラリを使用するために、システムクロックおよび T16 ch.0 を適切な速度に設定する必要があります。システムクロックおよび T16 ch.0 は、OscControl.c にて設定します。各関数の処理内容は、以下の通りです。

表 2.1.1 “OscControl.c”の関数

関数名	処理内容
OscClockSourceInitialize()	本ライブラリを使用するために、システムクロックおよび T16 ch.0 を適切な速度に設定します。 ※ユーザプログラムの設定を変更します。
OscClockSourceFinalize()	システムクロックおよび T16 ch.0 を、ユーザプログラムの設定に戻します。

ご注意：本関数は、本ライブラリ内で使用されます。そのため、必ず本関数のソースコードを確認し、必要に応じて修正してください。

本ライブラリを使用するためには、上記関数を使用し、以下の適切な動作速度に設定してください。

表 2.1.2 システムクロックと T16ch.0 の動作速度

機種名	ライブラリバージョン	システムクロック	T16 Ch.0 ^{*1}
S1C17M23/24/25	Ver1.3x 以前	OSC3=16MHz 分周比 1/4	OSC3=16MHz
	Ver1.4x 以降	OSC3=16MHz 分周比 1/1	OSC3=16MHz
S1C17M30/31/32/34	Ver2.2 以前	OSC3=16MHz 分周比 1/4	OSC3=16MHz
	Ver2.3 以降	OSC3=16MHz 分周比 1/1	OSC3=16MHz
S1C17M33	Ver2. 2 以前	OSC3=16MHz 分周比 1/4	OSC3=16MHz
	Ver2.3 以降	OSC3=16MHz 分周比 1/1	OSC3=16MHz
S1C17W18	全バージョン	OSC3=4MHz 分周比 1/1	OSC3=4MHz
S1C17W36	全バージョン	OSC3=4MHz 分周比 1/1	OSC3=4MHz

*1 T16 ch.0 の分周比設定は本ライブラリ内で行われます。

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“3. 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");
    }
}
```

5. データの読み出し単位／書き込み単位について

データの読み出し単位および書き込み単位は、対象機種および本ライブラリのバージョンにより異なります。これらの単位は、本ライブラリ内の“DataFlashCommand.h”の以下の関数定義の型に左右されますので、ご確認ください。

■ 関数定義

```
extern int DataFlashWrite(unsigned short address, unsigned char data);
extern unsigned char DataFlashRead(unsigned short address);
extern unsigned char DataFlashReadCurrent(void);
```

■ データ読み出し単位／書き込み単位

表 2.1.3 型と単位

型	データ読み出し単位／書き込み単位
unsigned char	1 バイト
unsigned short	2 バイト
unsigned long	4 バイト

2. ライブラリの使い方

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、使用するクロックソース等に影響されます。参考として、S1C17W18 および S1C17M33 の 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。

- S1C17W18 (システムクロック : 内蔵 OSC3 4MHz)

Typ.値	7ms
Max.値	43ms
- S1C17M33 (システムクロック : 内蔵 OSC3 16MHz)

Typ.値	6ms
Max.値	38ms

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システムクロックと 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には“S1C17(xxx)テクニカルマニュアル”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 ICDmini の FLASH_VCC_OUT 端子と MCU の Vpp 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“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 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。

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. ライブラリ仕様” を参照してください。

3. ライブラリ仕様

3. ライブラリ仕様

3.1 ライブラリ関数の詳細

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 xxx 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。
	unsigned short	
	unsigned long	
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。		
・ 書き込みデータの型は、機種およびバージョンによって異なります。		

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 unsigned short unsigned long	読み出しデータ。	
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。		
・ 読み出しデータの型は、機種およびバージョンによって異なります。		

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 unsigned short unsigned long	読み出しデータです。	
機能		
<p>現在アドレスからデータ読み出しを行う。</p> <p>① 現在アドレスからデータを読み出す。</p> <ul style="list-style-type: none">- 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。 <p>② 戻り値を返す。</p>		
備考		
<ul style="list-style-type: none">・ 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。・ 読み出しデータの型は、機種およびバージョンによって異なります。		

クロックソースの設定変更

関数名		
OscClockSourceInitialize (void)		
引数		
なし		
戻り値		
なし		
機能		
<p>ライブラリが動作するための適切な速度にシステムクロック、T16Ch.0 の設定を行います。</p> <p>① システムクロック、T16Ch.0 の設定を保存します。</p> <p>② Flash Read Cycle を変更します。</p> <p>③ システムクロックの設定を行います。</p> <p>④ T16 Ch.0 の設定を行います。</p>		
備考		
<p>ライブラリ関数から書き込み処理前に呼び出されます。</p> <p>ソースファイルを公開しているため、ユーザーにて修正が可能です。</p>		

3. ライブラリ仕様

クロックソースの設定復元

関数名		
OscClockSourceFinalize (void)		
引数		
なし		
戻り値		
なし		
機能		
システムクロック、T16Ch.0 の設定をライブラリ関数呼び出し前の設定に戻します。 ① システムクロックの設定を復元します。 ② T16 Ch.0 の設定を復元します。 ③ Flash Read Cycle を復元します。		
備考		
ライブラリ関数から書き込み処理終了後呼び出されます。 ソースファイルを公開しているため、ユーザーにて修正が可能です。		

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)

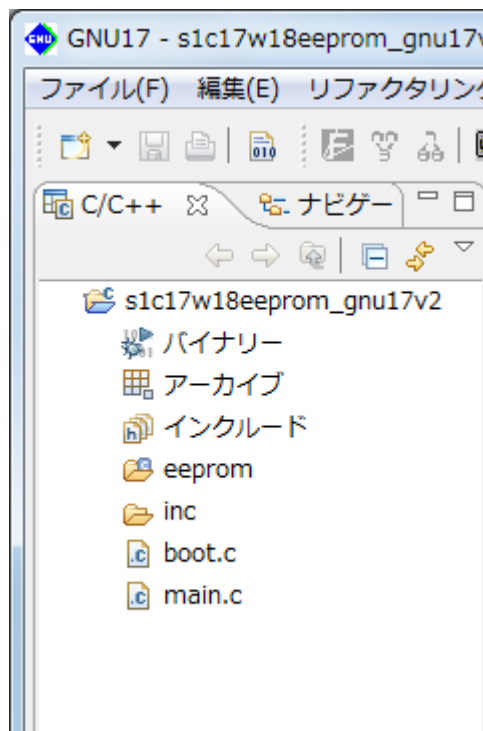
GNU17 Ver.2.x による本パッケージのライブラリを使用する方法を以下に記します。

(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。

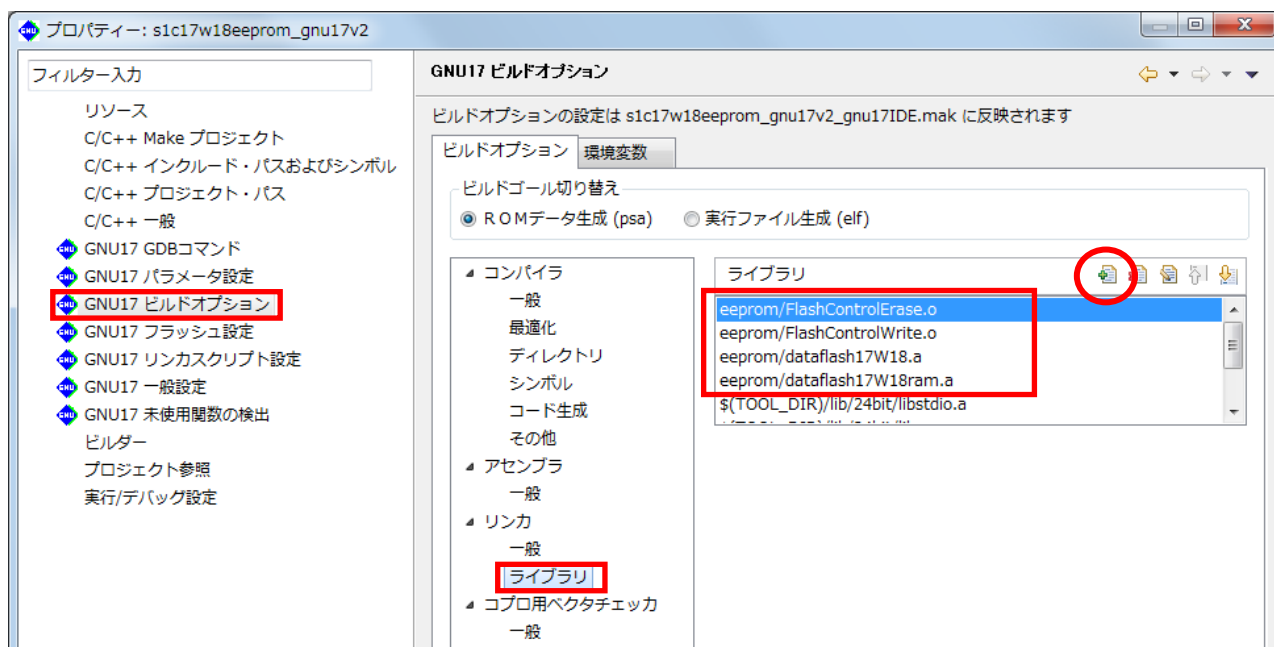
1. ライブラリ、ヘッダファイルの追加

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

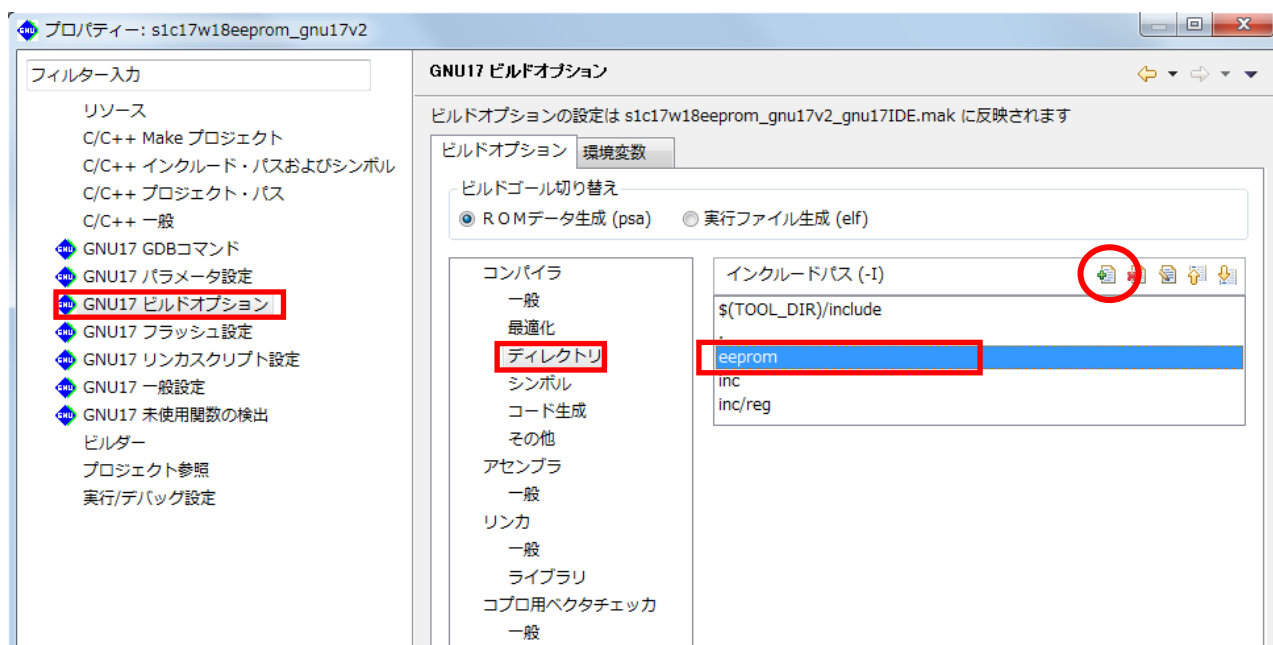
インポートしたライブラリを使用するために、ライブラリ設定に追加します。
プロジェクトの[プロパティ]-[GNU17 ビルドオプション]-[リンカ]-[ライブラリ]から下図の赤丸を選び、eeprom フォルダ内の “dataflash17W18.a”、“dataflash17W18ram.a”、“FlashControlErase.o”、“FlashControlWrite.o”を選択し、追加します。



3. インクルードパス設定

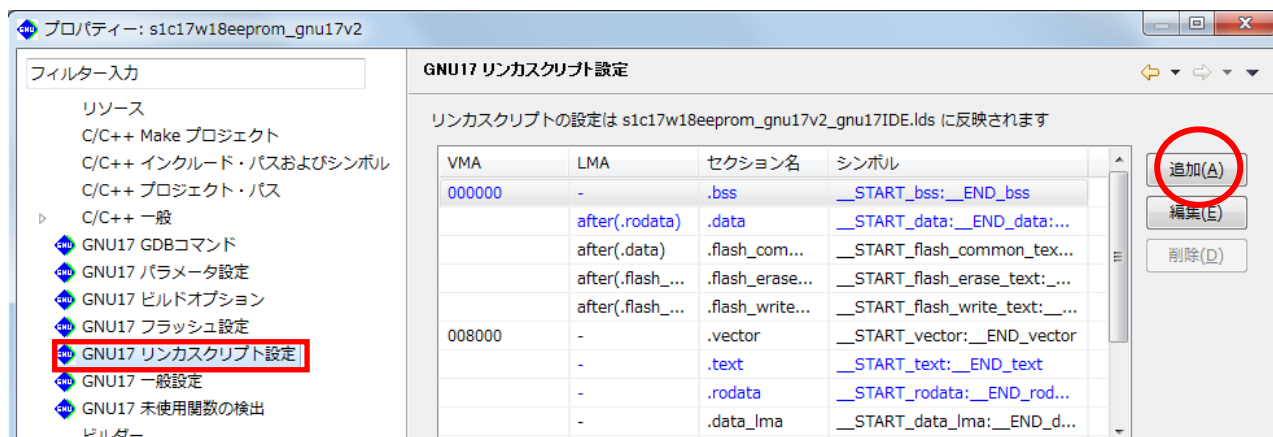
eeeprom フォルダにある”DataFlashCommand.h”を使用するために、インクルードパスを設定します。プロジェクトの[プロパティ]-[GNU17 ビルドオプション]-[ディレクトリ]から下図の赤丸を選び、プロジェクトに eeeprom フォルダへのインクルードパスを設定します。

ご注意：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。



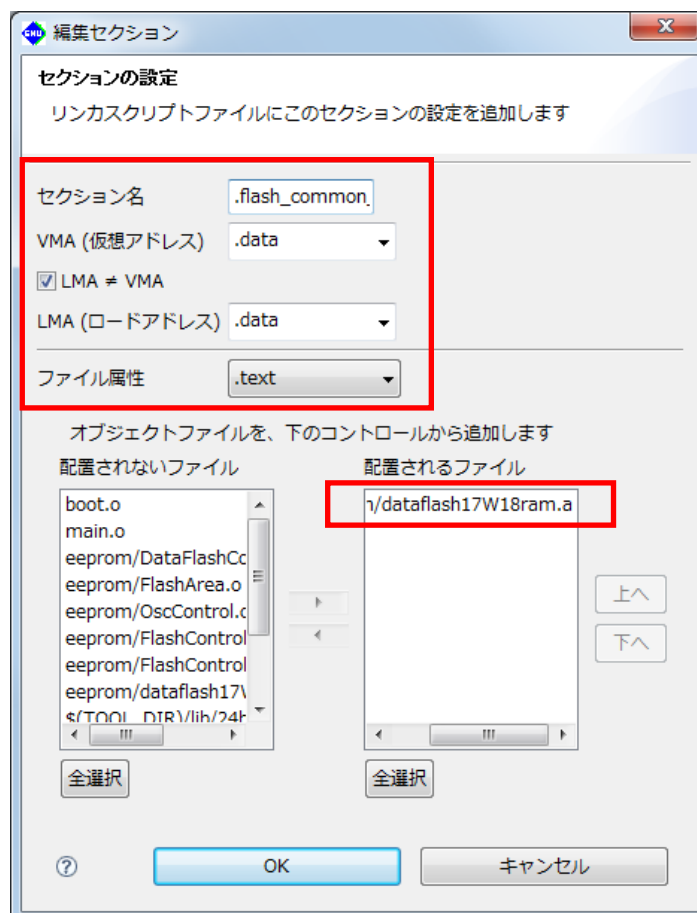
4. リンカスクリプト設定

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]-[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。

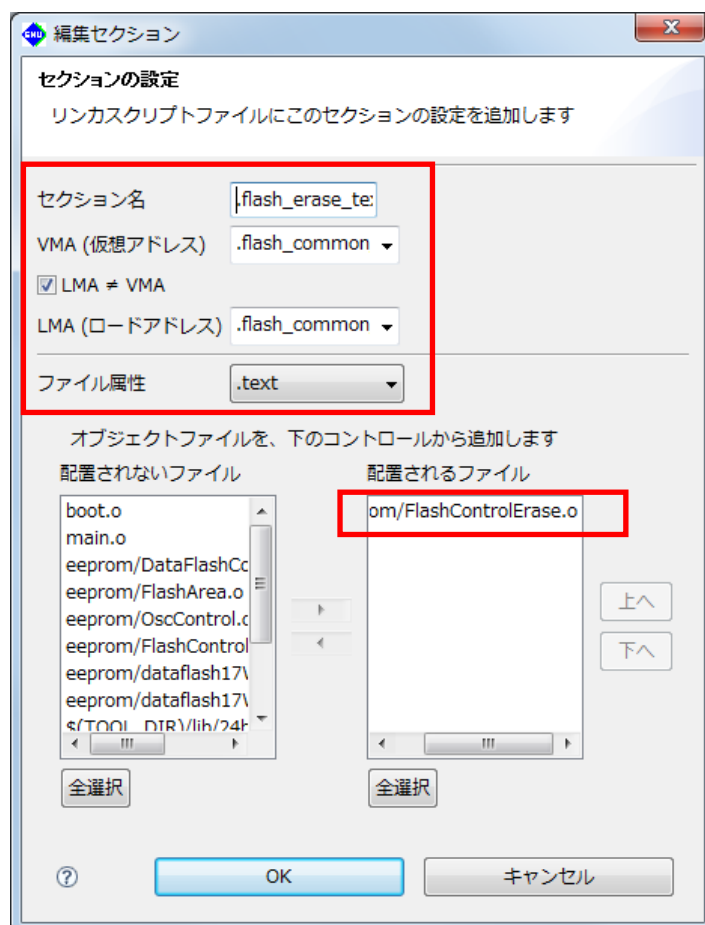


追加するのは、“.flash_common_text”、“.flash_erase_text”、“.flash_write_text”、“.flashdata_address”の各セクションです。先頭に”.”のついた上記の名前とし、下図に従って追加してください。

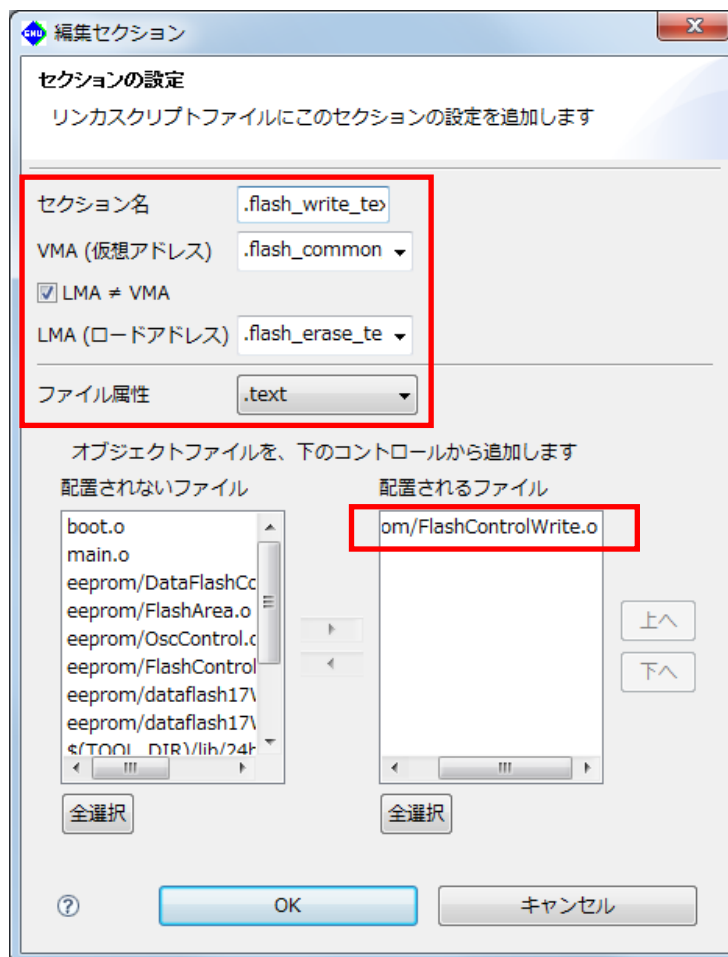
“.flash_common_text” セクションに “dataflashW18ram.a” を配置します。



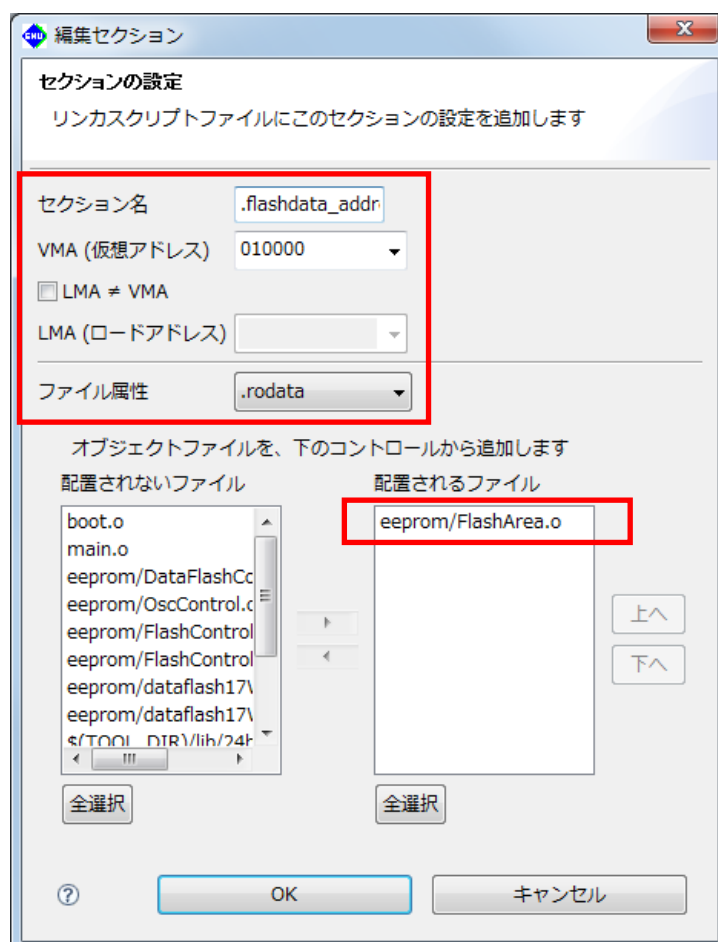
“flash_erase_text” セクションに “FlashControlErase.o” を配置します。
VMA、LMA は “.flash_common_text” とします。



“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)

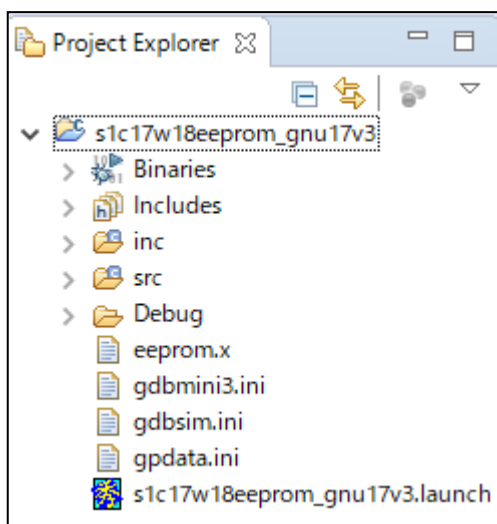
GNU17 Ver.3.x による本パッケージのライブラリの使用方法を以下に記します。

(以下では S1C17W18 を例としています)

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。

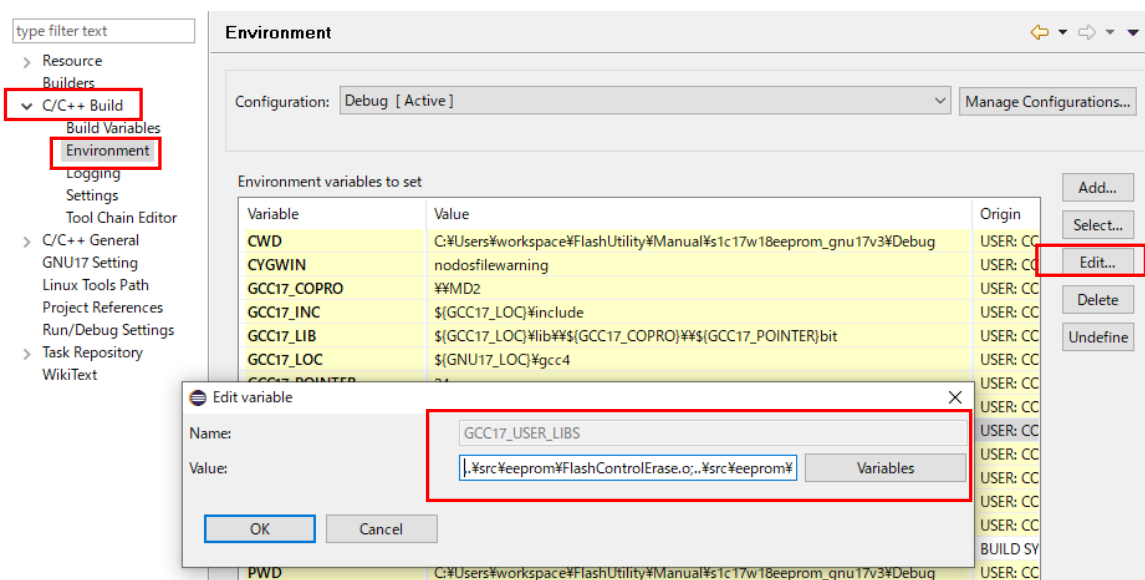


2. ライブラリ設定

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

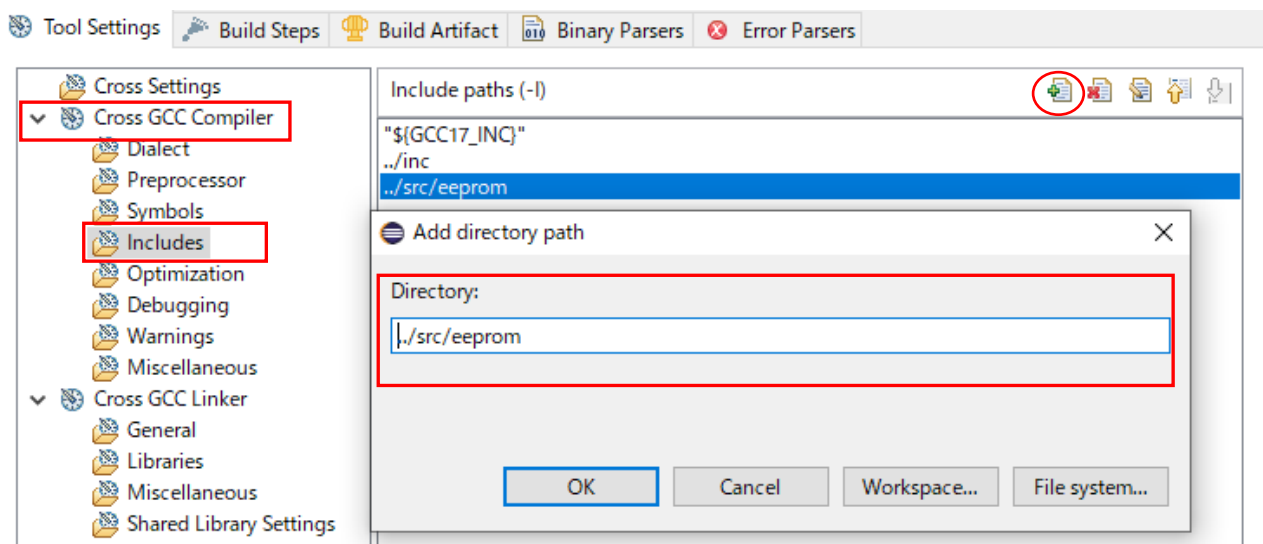


3. インクルードパス設定

eeeprom 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\eeeprom フォルダへのインクルードパスを設定します。

"../src/eeeprom"

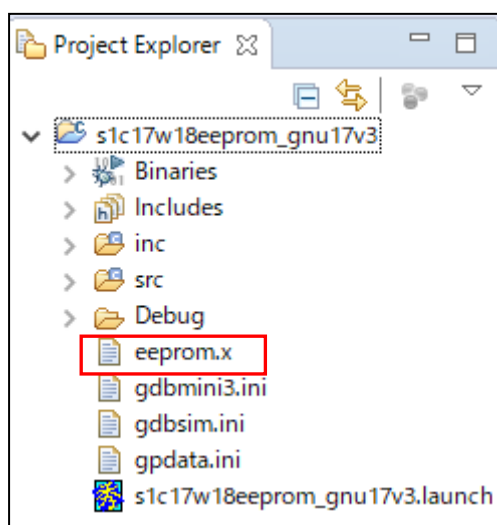


4. リンカスクリプト設定

ライブラリのリンカスクリプト設定をします。

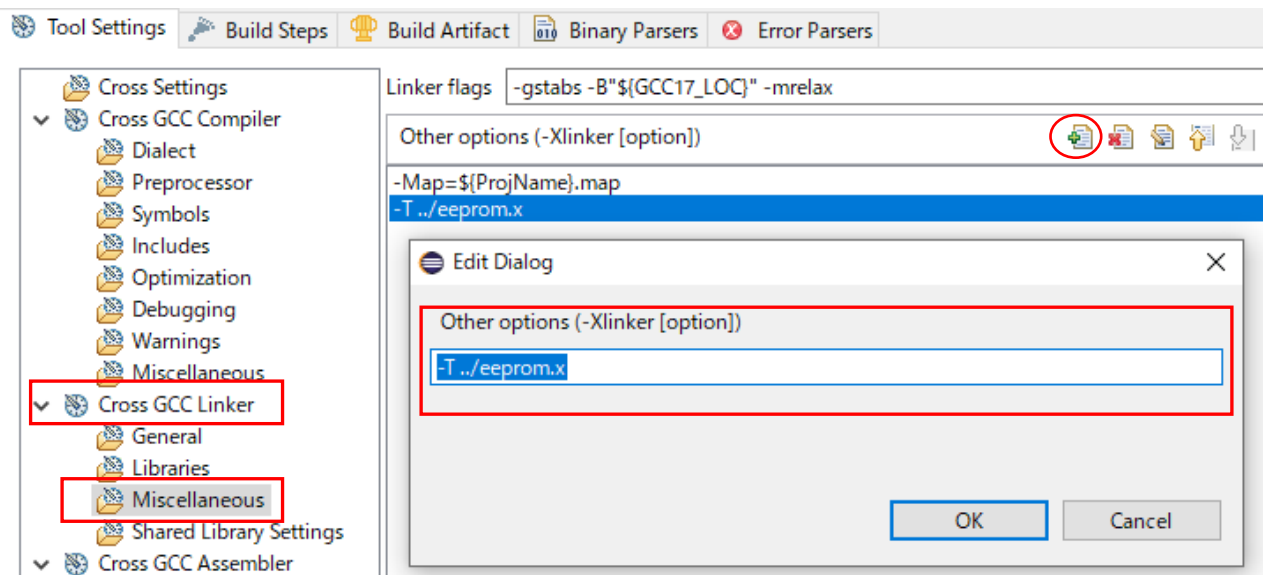
以下のフォルダに EEPROM エミュレーションライブラリ用に記述したリンカスクリプトがありますので、プロジェクトフォルダへコピーしてください。

¥c17(xxx)_sample_gnu17v3¥eeeprom.x



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

-T ../eeprom.x



このリンクスクリプトでは、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

付ー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。
Rev.1.4	2019/11/01	P16	変更	「Appendix B 2 ライブラリ設定」の記述を変更。
Rev.1.5	2020/05/26	P5	変更	「2.4 ライブラリ使用上の注意」の記述を変更。
Rev1.6	2022/03/04	P4	変更	「2.1 アプリケーションプログラムへの適用方法」の“3. クロックソースの設定”を変更。
		P5	追加	「2.1 アプリケーションプログラムへの適用方法」に“5.EEPROM 読み出し／書き込み関数のデータの型について”を追加。 「2.3 書き込み時間」の記述を追加。
		P9-10	追加	「3.1 ライブラリ関数の詳細」の記述を追加。
		P17-19	追加	「Appendix B.ライブラリをプロジェクトへ組み込む方法(GNU17 Ver.3.x)」の図を追加。

セイコーエプソン株式会社

営業本部 MD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 : 413120106

2015 年 11 月 作成

2020 年 5 月 改訂

2022 年 3 月 改訂