EPSON
EXCEED YOUR VISION

# S1C17 M01/W22/W23/W15
# 自己書き換え
# ソフトウェア(FLS)
# 説明書

セイコーエプソン株式会社

Rev.1.1

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、S1C17M01/W22/W23/W15 内蔵 FLASH 消去／書き込みソフトウェアである FLS17 M01/W22/W23/W15 のライブラリにより、アプリケーションプログラムでの内蔵 FLASH 自己書き換え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)
・S1C17xxx 自己書き換えソフトウェアパッケージ（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 S1C17M01/W22/W23/W15 の共通資料です。
本資料では (xxx) = 該当機種名(M01/W22/W23/W15) となります。

外部電圧供給を行う際には、安定するまで十分待ち時間を確保してください。
オペレーションが終了したら外部電圧供給は停止してください。

目　次

# 1. 概要

自己書き換えソフトウェアパッケージは、対象機種の内蔵フラッシュメモリ内のプログラムコードやデータを書き換え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の関数コールによって、フラッシュメモリ消去、書き込み処理を実現します。

アプリケーションプログラムからの関数コールによるフラッシュメモリ消去、書き込み処理時には、ICDmini（S5U1C17001H2, S5U1C17001H3）は不要となります。
＊ICDmini：デバッグ時および初期プログラム書き込みに使用します。

## 1.1 機能

ライブラリで提供する機能を以下に記します。

フラッシュメモリ消去機能：
flash_erase(char * flsTopAdd, unsigned short startSct, unsigned short endSct)

フラッシュメモリの先頭アドレス、消去開始セクタ、消去終了セクタを指定し、
S1C17(xxx)内蔵フラッシュメモリをセクタ消去します。
消去セクタは M01 = 1～16、W22/15 = 1～32、W23 = 1～48 まで指定可能です。

フラッシュメモリ書き込み機能：
flash_load(char * loadAdd, unsigned short loadSize, unsigned char* pData)

書き込み先アドレス、書き込みデータサイズ、書き込みデータへのポインタを指定し、
メモリ上のデータをフラッシュメモリへ書き込みます。
注） 書き込みデータサイズの有効範囲は、ライブラリ内ではチェックしません。

＜本パッケージの注意＞
1. 本パッケージは、GNU17（S5U1C17001C）によるプログラム開発用に作成されています。
2. ベリファイ機能については、フラッシュメモリ消去／書き込み機能に組み込まれています。

## 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)self_r1x
    + lib                                          : 自己書き換えライブラリ
    + c17(xxx)_sample_gnu17v2                      : GNU17 Ver.2.x 用サンプルプログラム
    + c17(xxx)_sample_gnu17v3                      : GNU17 Ver.3.x 用サンプルプログラム
    - s1c17m01_w22_w23_w15flsan_i_revXXX.pdf       : 説明書(日本語)
    - s1c17m01_w22_w23_w15flsan_e_revXXX.pdf       : 説明書(英語)
    - s1c17(xxx)self_notes_j.txt                   : 補足説明書(日本語)
    - s1c17(xxx)self_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)self_r1x /lib

| ファイル名 | 機能 |
|---|---|
| fls17(xxx)RAM.o | S1C17(xxx)用フラッシュメモリ消去／書き込み |
| fls17(xxx)ROM.o | S1C17(xxx)用フラッシュメモリ消去／書き込み |
| fl_self.h | 関数宣言用ヘッダファイル |

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　c17(xxx)_sample

| ファイル / フォルダ名 | 機能 |
|---|---|
| lib | 自己書き換えライブラリ(フォルダ) |
| boot.c | boot プログラム |
| main.c | main プログラム |
| data.c | （更新前）データ |
| update.c | 更新用データ |

# 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

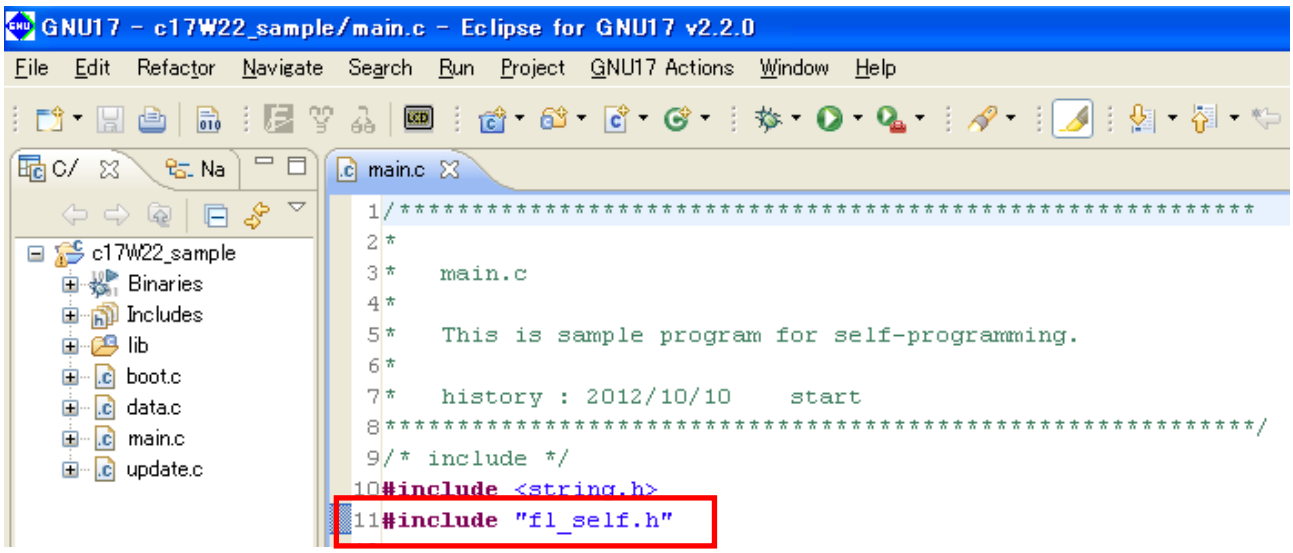
以下では、本ライブラリを使用するアプリケーションプログラムのソースファイルに必要になる対応事項を説明します。

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.B 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、“fl_self.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. フラッシュメモリ消去／書き込み関数の追加

   フラッシュメモリの消去／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。

```
int update(void) {
        int result;

        // erase 0xB000-0xBFFF
        result = flash_erase((char *)0x8000, 7, 7);

        if(result == FLS_E_SUCCESS) {
                // write data to 0xb000
                result = flash_load((char *)0xB000, 16, updateLineBit);
        }

        return result;
}
```

## 2.2 Flash プログラミング電圧 VPP

フラッシュメモリ消去/書き込み時は Flash プログラミング電源( VPP 電源 )を供給してください。
VPP 電源供給手順を以下に記します。尚、フラッシュメモリ消去/書き込み時以外は VPP 電源供給を停止してください。

① VDDが1.8V以上であることを確認します。
② 書き込みたいデータをRAMに配置します。
③ VPP電源を供給します。
④ VPP電圧が安定するのを待ちます。
⑤ flash_erase関数を実行し、指定した領域を消去します。
⑥ flash_load関数を実行し、消去されている領域の指定したアドレスにデータを書き込みます。
⑦ 消去されている領域の、別のアドレスにデータを書き込む場合は⑥を繰り返します。
⑧ VPP 電源供給を停止します。

“S1C17(xxx)テクニカルマニュアル”にて、Flash プログラミング電圧、基本外部結線図についても参照してください。

## 2.3 内蔵 RAM 使用量

本ライブラリでは内蔵 RAM 領域を使用します。
機種別の RAM 使用量については、補足説明書 s1c17(xxx)self_notes_j.txt をご参照ください。

## 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。

- flash_erase 関数, flash_load 関数の前で割込み禁止をしてください。
- ライブラリ実行時は、ライブラリを配置した領域を壊さないようにしてください。
- 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。
- 本ライブラリでは以下動作を行います。
  1. 16 ビットタイマ(T16)の ch.0 を使用しています。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プログラムで、本ライブラリを使用する場合は、注意してください。
  2. W22/W23/W15 では書き換え中 VD1 を 1.8V に変更。書き換え終了後、1.4V へ戻す処理を行っています。本ライブラリ使用時には、お客様が設定した全ての周辺回路を停止するようにしてください。
  3. システムクロックを M01=IOSC、W22/23/15=OSC3 に変更しています。
  4. W22/W23/W15 では、本ライブラリにて、クロックジェネレータ (CLG)の CGL OSC3 Control Register を初期値に戻しています。そのため、CLG OSC3 Control Register の内容が変更されます。CLG OSC3 Control Register を使用しているプログラムで、本ライブラリを使用する場合は、注意してください。

## 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 0xB000 の領域をセクタ消去後、16 バイト書き込みます。

**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 VPP電源供給
VPP電源を供給してください。
⑦ 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 内蔵フラッシュメモリのセクタ7（0xB000～0xB7FF）を消去します（main.c / flash_erase）。
- フラッシュメモリの先頭アドレス: 0x8000
- フラッシュメモリの消去開始セクタ: 7
- フラッシュメモリの消去終了セクタ: 7

② 消去した0xB000に更新用データupdateLineBit[]を書き込みます（main.c / flash_load）。
- 書き込み先アドレス: 0xB000
- 書き込みデータサイズ: 16 byte
- 書き込みデータへのポインタ: updateLineBit

書き換え前後の 0xB000 のデータは以下の通りです。

書き換え前 (0xB000)

| 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F |
|---|

書き換え後 (0xB000)

| 0F 0E 0D 0C 0B 0A 09 08 07 06 05 04 03 02 01 00 |
|---|

書き換え結果はメモリダンプで確認してください。
セクタ番号と対応アドレスについてはAppendix.A フラッシュメモリ仕様 を参照してください。

flash_erase, flash_load 関数については、3.1 フラッシュメモリ消去／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 フラッシュメモリ消去／書き込み関数詳細

fl_17(xxx).o（fl_17(xxx).c）に記述された関数について記します。

フラッシュメモリ消去

| **関数名** | | |
|---|---|---|
| flash_erase(char * flsTopAdd, unsigned short startSct, unsigned short endSct) | | |
| **引数** | | |
| flsTopAdd | unsigned long | フラッシュメモリの先頭アドレスを表す。 |
| startSct | unsigned long | フラッシュメモリの消去開始セクタを表す。 |
| endSct | unsigned long | フラッシュメモリの消去終了セクタを表す。 |
| **戻り値** | | |
| int | フラッシュメモリ消去の結果（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、フラッシュメモリを消去する。<br>①引数が正しいかチェックする。<br>②消去を行う。<br>③戻り値を返す。 | | |
| **備考** | | |
| 第 2 引数、第 3 引数のセクタ番号の有効範囲は、M01=「1～16」 W22/W15=「1～32」W23=「1～48」です。 | | |

フラッシュメモリ書き込み

| **関数名** | | |
|---|---|---|
| flash_load(char * loadAdd, unsigned short loadSize, unsigned char* pData) | | |
| **引数** | | |
| loadAdd | unsigned long | 書き込み先アドレスを表す。 |
| loadSize | unsigned long | 書き込みデータサイズを表す。 |
| pData | unsigned char* | 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。 |
| **戻り値** | | |
| int | フラッシュメモリ書き込みの結果（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、フラッシュメモリに書き込みを行う。<br>①仮想アドレスを物理アドレスに変換する。<br>②書き込みを行う。<br>- 初期設定を行う。<br>- 書き込みたいデータと一致しているかチェックを行う。<br>- 一致していなければ書き込みを行う。<br>③戻り値を返す。 | | |
| **備考** | | |
| 書き込みデータ単位は「バイト（8bit）」単位です。 | | |

## 3.2 エラーコード定義

表 3 エラーコード

| 定義名 | 値 | 説明 |
|---|---|---|
| FLS_E_SUCCESS | 0 | フラッシュ操作正常終了 |
| FLS_E_PROGRAM_COUNT_OVER_ERR | 1 | プログラムカウントエラー<br>書き込み／消去時にベリファイ（書き込み時は書き込みたいデータ、消去時は 0xFFFF と比較する）を行い、ベリファイカウントが最大プログラムカウントを超えた場合に返る。 |

| | | |
|---|---|---|
| FLS_E_STRAT_SCT_ERR | 3 | 消去開始セクタが範囲外<br>消去開始セクタが 0 未満、もしくは最大セクタ数を超える場合に返る。 |
| FLS_E_END_SCT_ERR | 4 | 消去終了セクタが範囲外<br>消去終了セクタが 0 未満、もしくは最大セクタ数を超える場合に返る。 |
| FLS_E_TOP_ADDRESS_ERR | 5 | 先頭アドレスエラー<br>FLASH_ERASE 関数の第 1 引数（フラッシュメモリ先頭アドレス）が、フラッシュメモリの先頭アドレスとして決められた値と等しくない場合に返る。 |

# Appendix

## A. フラッシュメモリ仕様

以下に各機種の内蔵フラッシュメモリ仕様を記します。内蔵フラッシュメモリについては、”S1C17(xxx)テクニカルマニュアル”も参照してください。

### 1. S1C17M01 内蔵フラッシュメモリの構成

- セクタ数　　：　16sectors
- メモリサイズ　：　32Kbyte
- 消去単位　　：　セクタ単位（2Kbyte）
- 書き込み単位　：　バイト（8bit）単位
- 書き換え時間　：　初回 1 セクタ program 時間 109 msec (Typical)
  初回 1 セクタ erase 時間 121 msec (Typical)
- Flash プログラミング電圧（VPP）：消去 7.5V，書き込み 7.5V
  外部電圧供給を行う際には、安定するまで十分待ち時間を確保してください。
  オペレーションが終了したら外部電圧供給は停止してください。

表 4　S1C17M01 メモリアドレスと Flash セクタの対応

| S1C17M01 アドレス | Flash セクタ番号 |
|---|---|
| 0x0f800～0x0ffff | 16 |
| 0x0f000～0x0f7ff | 15 |
| 0x0e800～0x0efff | 14 |
| 0x0e000～0x0e7ff | 13 |
| 0x0d800～0x0dfff | 12 |
| 0x0d000～0x0d7ff | 11 |
| 0x0c800～0x0cfff | 10 |
| 0x0c000～0x0c7ff | 9 |
| 0x0b800～0x0bfff | 8 |
| 0x0b000～0x0b7ff | 7 |
| 0x0a800～0x0afff | 6 |
| 0x0a000～0x0a7ff | 5 |
| 0x09800～0x09fff | 4 |
| 0x09000～0x097ff | 3 |
| 0x08800～0x08fff | 2 |
| 0x08000～0x087ff | 1 |

## 2. S1C17W22 内蔵フラッシュメモリの構成

・ セクタ数　　：　32sectors
・ メモリサイズ　：　64Kbyte
・ 消去単位　　：　セクタ単位（2Kbyte）
・ 書き込み単位　：　バイト（8bit）単位
・ 書き換え時間　：　初回 1 セクタ program 時間 220 msec (Typical)
　　初回 1 セクタ erase 時間 208 msec (Typical)
・ Flash プログラミング電圧（VPP）：消去 7.5V，書き込み 7.5V
　　外部電圧供給を行う際には、安定するまで十分待ち時間を確保してください。
　　オペレーションが終了したら外部電圧供給は停止してください。

表 5　S1C17W22 メモリアドレスと Flash セクタの対応

| S1C17W22 アドレス | Flash セクタ番号 |
|---|---|
| 0x17800～0x17fff | 32 |
| 0x17000～0x177ff | 31 |
| 0x16800～0x16fff | 30 |
| 0x16000～0x167ff | 29 |
| 0x15800～0x15fff | 28 |
| 0x15000～0x157ff | 27 |
| 0x14800～0x14fff | 26 |
| 0x14000～0x147ff | 25 |
| 0x13800～0x13fff | 24 |
| 0x13000～0x137ff | 23 |
| 0x12800～0x12fff | 22 |
| 0x12000～0x127ff | 21 |
| 0x11800～0x11fff | 20 |
| 0x11000～0x117ff | 19 |
| 0x10800～0x10fff | 18 |
| 0x10000～0x107ff | 17 |
| 0x0f800～0x0ffff | 16 |
| 0x0f000～0x0f7ff | 15 |
| 0x0e800～0x0efff | 14 |
| 0x0e000～0x0e7ff | 13 |
| 0x0d800～0x0dfff | 12 |
| 0x0d000～0x0d7ff | 11 |
| 0x0c800～0x0cfff | 10 |
| 0x0c000～0x0c7ff | 9 |
| 0x0b800～0x0bfff | 8 |
| 0x0b000～0x0b7ff | 7 |
| 0x0a800～0x0afff | 6 |
| 0x0a000～0x0a7ff | 5 |
| 0x09800～0x09fff | 4 |
| 0x09000～0x097ff | 3 |
| 0x08800～0x08fff | 2 |
| 0x08000～0x087ff | 1 |

### 3. S1C17W23 内蔵フラッシュメモリの構成

・ セクタ数　　：　48sectors
・ メモリサイズ　：　96Kbyte
・ 消去単位　　：　セクタ単位（2Kbyte）
・ 書き込み単位　：　バイト（8bit）単位

・ 書き換え時間　：　初回 1 セクタ program 時間 220 msec (Typical)
　　初回 1 セクタ erase 時間 208 msec (Typical)
・ Flash プログラミング電圧（VPP）：消去 7.5V，書き込み 7.5V
　　外部電圧供給を行う際には、安定するまで十分待ち時間を確保してください。
　　オペレーションが終了したら外部電圧供給は停止してください。

表 6　S1C17W23 メモリアドレスと Flash セクタの対応

| S1C17W23 アドレス | Flash セクタ番号 |
|---|---|
| 0x17800～0x17fff | 32 |
| 0x17000～0x177ff | 31 |
| 0x16800～0x16fff | 30 |
| 0x16000～0x167ff | 29 |
| 0x15800～0x15fff | 28 |
| 0x15000～0x157ff | 27 |
| 0x14800～0x14fff | 26 |
| 0x14000～0x147ff | 25 |
| 0x13800～0x13fff | 24 |
| 0x13000～0x137ff | 23 |
| 0x12800～0x12fff | 22 |
| 0x12000～0x127ff | 21 |
| 0x11800～0x11fff | 20 |
| 0x11000～0x117ff | 19 |
| 0x10800～0x10fff | 18 |
| 0x10000～0x107ff | 17 |
| 0x0f800～0x0ffff | 16 |
| 0x0f000～0x0f7ff | 15 |
| 0x0e800～0x0efff | 14 |
| 0x0e000～0x0e7ff | 13 |
| 0x0d800～0x0dfff | 12 |
| 0x0d000～0x0d7ff | 11 |
| 0x0c800～0x0cfff | 10 |
| 0x0c000～0x0c7ff | 9 |
| 0x0b800～0x0bfff | 8 |
| 0x0b000～0x0b7ff | 7 |
| 0x0a800～0x0afff | 6 |
| 0x0a000～0x0a7ff | 5 |
| 0x09800～0x09fff | 4 |
| 0x09000～0x097ff | 3 |
| 0x08800～0x08fff | 2 |
| 0x08000～0x087ff | 1 |

| S1C17W23 アドレス | Flash セクタ番号 |
|---|---|
| 0x1f800～0x1ffff | 48 |
| 0x1f000～0x1f7ff | 47 |
| 0x1e800～0x1efff | 46 |
| 0x1e000～0x1e7ff | 45 |
| 0x1d800～0x1dfff | 44 |
| 0x1d000～0x1d7ff | 43 |
| 0x1c800～0x1cfff | 42 |
| 0x1c000～0x1c7ff | 41 |
| 0x1b800～0x1bfff | 40 |
| 0x1b000～0x1b7ff | 39 |
| 0x1a800～0x1afff | 38 |
| 0x1a000～0x1a7ff | 37 |
| 0x19800～0x19fff | 36 |
| 0x19000～0x197ff | 35 |
| 0x18800～0x18fff | 34 |
| 0x18000～0x187ff | 33 |

### 4. S1C17W15 内蔵フラッシュメモリの構成

・ セクタ数　　：　32sectors
・ メモリサイズ　：　64Kbyte
・ 消去単位　　：　セクタ単位（2Kbyte）
・ 書き込み単位　：　バイト（8bit）単位

・ 書き換え時間　：　初回 1 セクタ program 時間 220 msec (Typical)
　　初回 1 セクタ erase 時間 208 msec (Typical)
・ Flash プログラミング電圧（VPP）：消去 7.5V，書き込み 7.5V
　　外部電圧供給を行う際には、安定するまで十分待ち時間を確保してください。
　　オペレーションが終了したら外部電圧供給は停止してください。

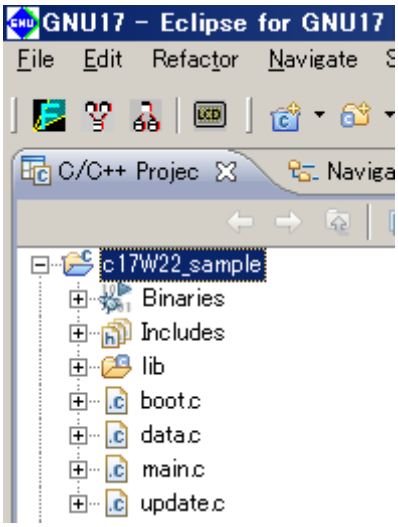
表 7　S1C17W15 メモリアドレスと Flash セクタの対応

| S1C17W15 アドレス | Flash セクタ番号 |
|---|---|
| 0x17800～0x17fff | 32 |
| 0x17000～0x177ff | 31 |
| 0x16800～0x16fff | 30 |
| 0x16000～0x167ff | 29 |
| 0x15800～0x15fff | 28 |
| 0x15000～0x157ff | 27 |
| 0x14800～0x14fff | 26 |
| 0x14000～0x147ff | 25 |
| 0x13800～0x13fff | 24 |
| 0x13000～0x137ff | 23 |
| 0x12800～0x12fff | 22 |
| 0x12000～0x127ff | 21 |
| 0x11800～0x11fff | 20 |
| 0x11000～0x117ff | 19 |
| 0x10800～0x10fff | 18 |
| 0x10000～0x107ff | 17 |
| 0x0f800～0x0ffff | 16 |
| 0x0f000～0x0f7ff | 15 |
| 0x0e800～0x0efff | 14 |
| 0x0e000～0x0e7ff | 13 |
| 0x0d800～0x0dfff | 12 |
| 0x0d000～0x0d7ff | 11 |
| 0x0c800～0x0cfff | 10 |
| 0x0c000～0x0c7ff | 9 |
| 0x0b800～0x0bfff | 8 |
| 0x0b000～0x0b7ff | 7 |
| 0x0a800～0x0afff | 6 |
| 0x0a000～0x0a7ff | 5 |
| 0x09800～0x09fff | 4 |
| 0x09000～0x097ff | 3 |
| 0x08800～0x08fff | 2 |
| 0x08000～0x087ff | 1 |

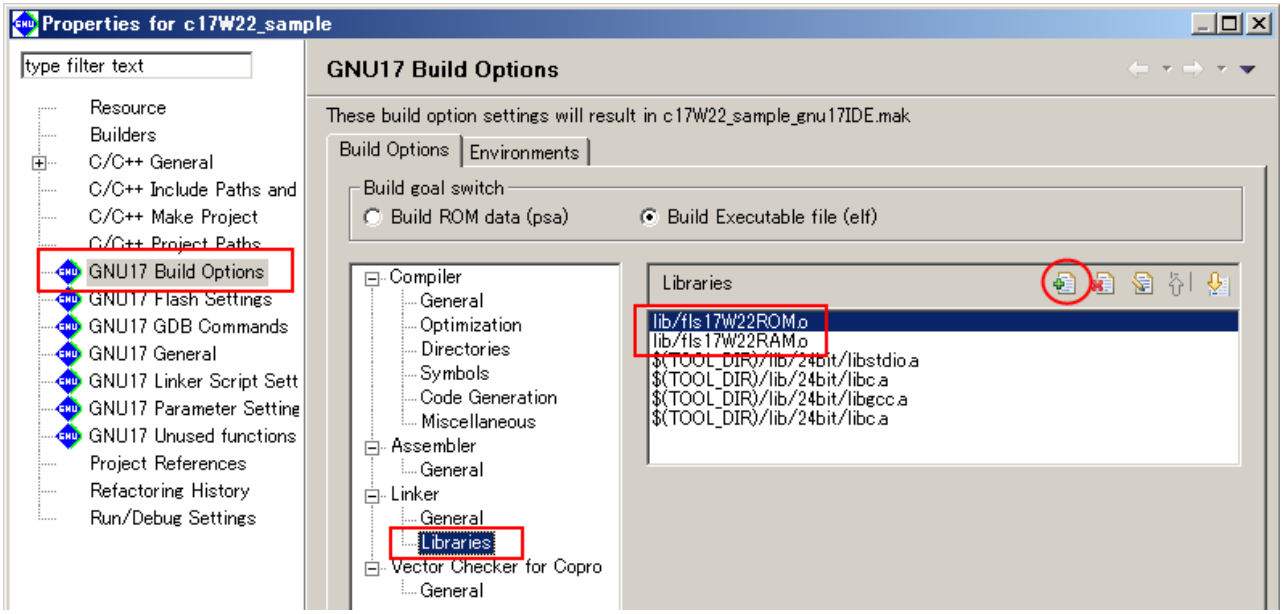
## B. ライブラリをプロジェクトへ組み込む方法(GNU17 Ver.2.x)

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 W22 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。

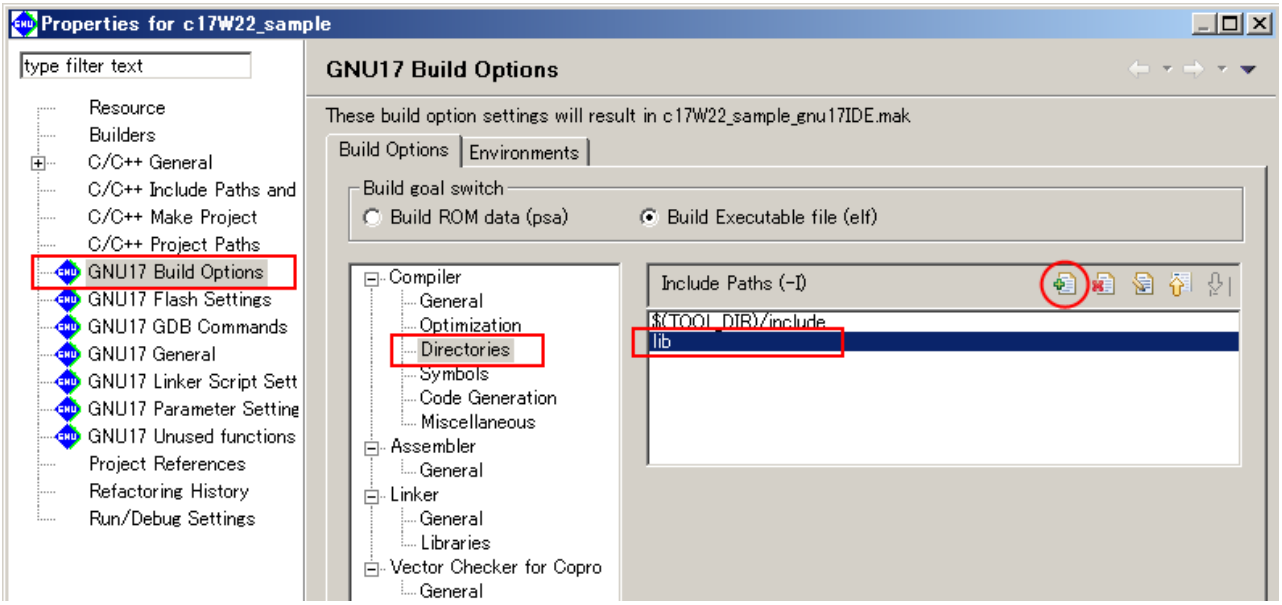
1. ライブラリ、ヘッダファイルの追加
   パッケージ内の lib フォルダをプロジェクトフォルダにインポートしてください。



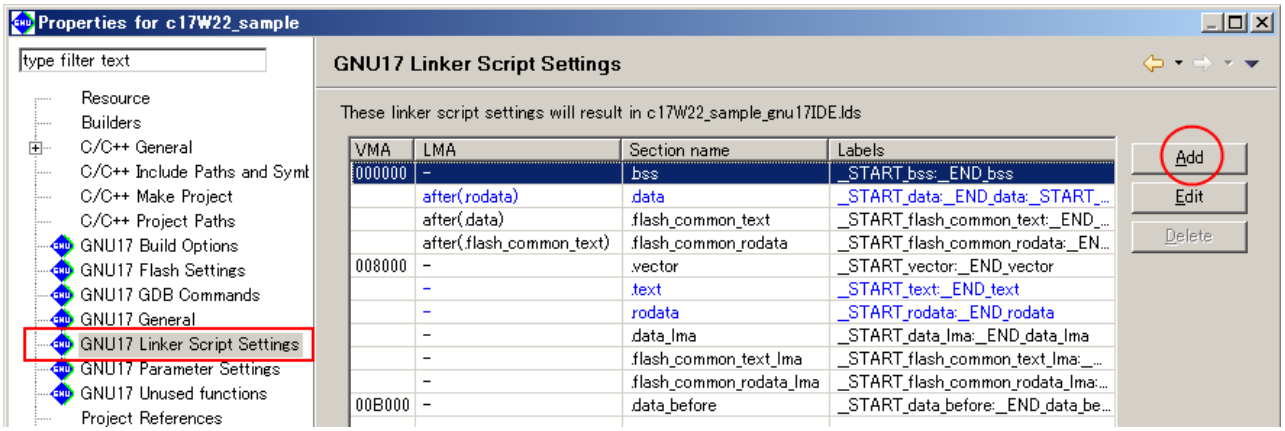

2. ライブラリ設定
   インポートしたライブラリを使用するために、ライブラリ設定に追加します。
   プロジェクトの[Properties]-[GNU17 Build Options]-[Linker]-[Libraries]から下図の赤丸を選び、lib フォルダ内の “fls17(xxx)ROM.o”、“fls17(xxx)RAM.o”を選択し、追加します。



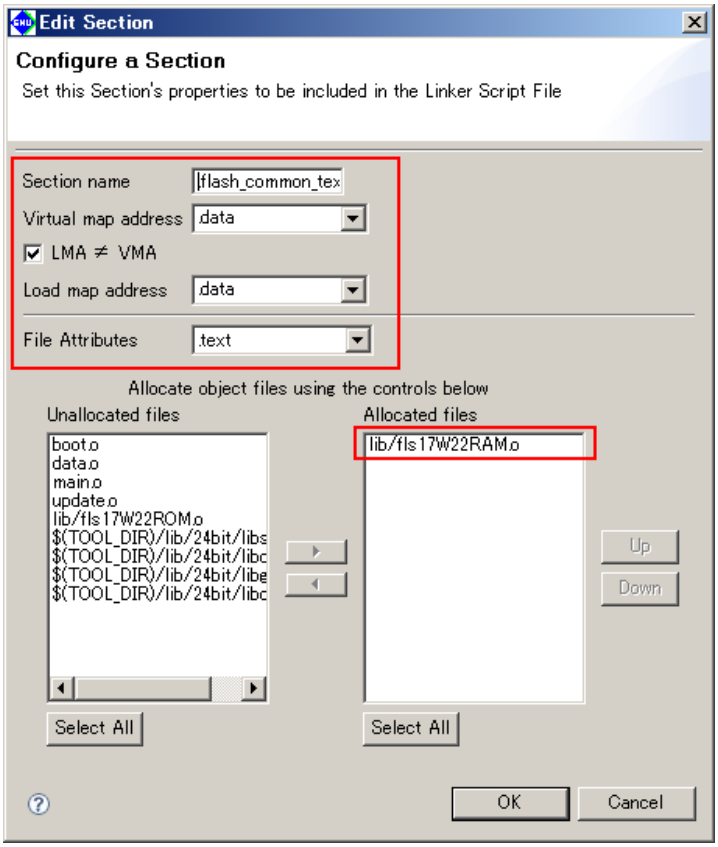
3. インクルードパス設定
lib フォルダにある”fl_self.h”を使用するために、インクルードパスを設定します。
プロジェクトの[Properties]-[GNU17 Build Options]-[Directories]から下図の赤丸を選び、プロジェクトに lib フォルダへのインクルードパスを設定します。
注）ソースファイル内で直接インクルードパスを指定する場合は必要ありません。

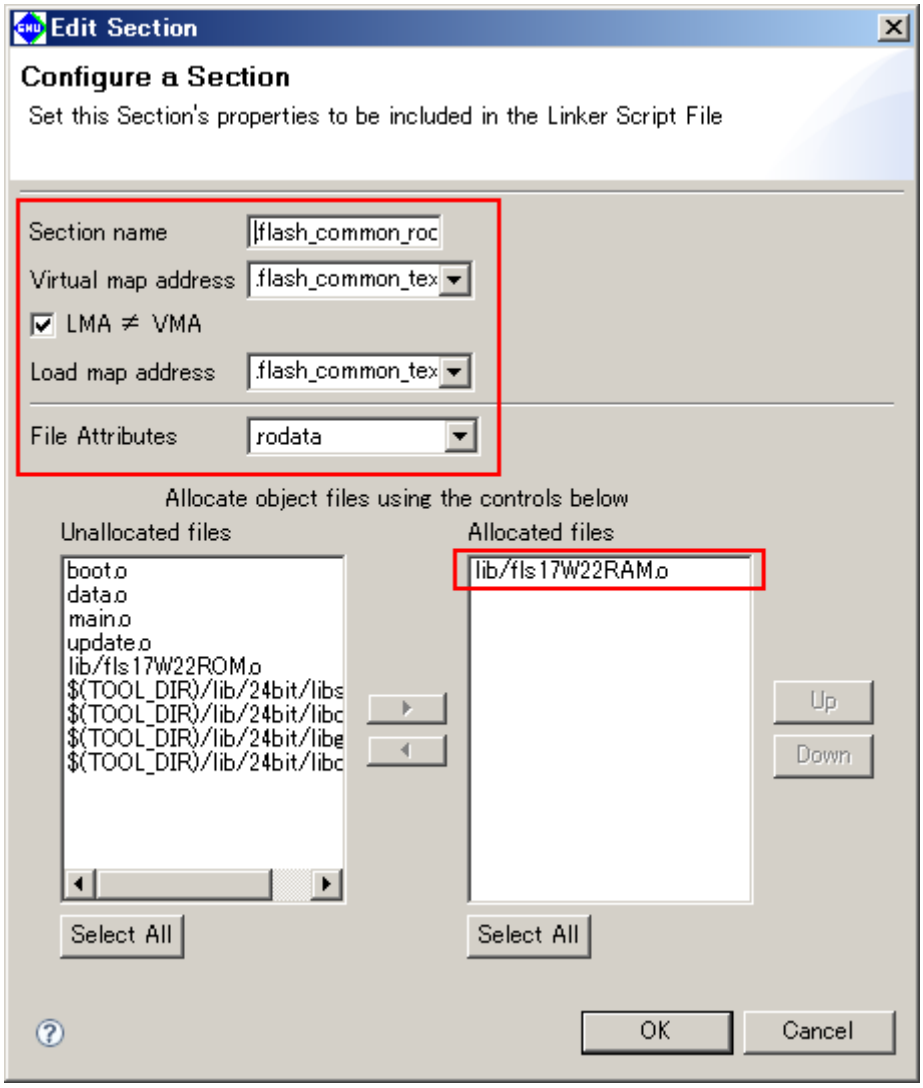


4. リンカスクリプト設定
インポートしたライブラリのリンカスクリプト設定をします。
プロジェクトの[Properties]-[GNU17 Linker Script Settings]から下図の赤丸を選び、ライブラリを配置するセクションを追加します。




追加するのは、.flash_common_text、.flach_common_rodata、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。
上記各セクションに下図のように “fls17(xxx)RAM.o”を配置します。“fls17(xxx)RAM.o”はフラッシュメモリに対して制御コマンドを発行するライブラリで、内蔵 RAM で実行する必要があるため、実行アドレス(VMA)を内蔵 RAM に配置してください。尚、”fls17(xxx)ROM.o”は、RAM 上に配置する必要はありません。




Edit Section
Configure a Section
Set this Section's properties to be included in the Linker Script File
Section name
.flash_common_roc
Virtual map address
.flash_common_tex
LMA ≠ VMA
Load map address
.flash_common_tex
File Attributes
rodata
Allocate object files using the controls below
Unallocated files
boot.o
data.o
main.o
update.o
lib/fls17W22ROM.o
$(TOOL_DIR)/lib/24bit/libs
$(TOOL_DIR)/lib/24bit/libc
$(TOOL_DIR)/lib/24bit/libg
$(TOOL_DIR)/lib/24bit/libc
Allocated files
lib/fls17W22RAM.o
Up
Down
Select All
Select All
OK
Cancel

## C. 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 lib 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¥lib フォルダ内の”fls17(xxx)ROM.o”、”fls17(xxx)RAM.o”を追加します。

```
..¥src¥lib¥fls17(xxx)RAM.o;..¥src¥lib¥fls17(xxx)ROM.o
```

**3. インクルードパス設定**
lib フォルダにある”fl_self.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¥lib フォルダへのインクルードパスを設定します。

```
"${workspace_loc:/${ProjName}/src/lib}"
```

**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。
以下のフォルダに内蔵 FLASH 自己書き換え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¥selfmodifying.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 ..¥selfmodifying.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”fls17(xxx)RAM.o”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_common_text
__END_flash_common_text
__START_flash_common_rodata_lma
__START_flash_common_rodata
__END_flash_common_rodata
```

また、以下の記述により、”fls17(xxx)RAM.o”を ROM に配置しないように設定しています。

```
*(EXCLUDE_FILE (*fls17*RAM.o ) .text)
*(EXCLUDE_FILE (*crt0.o *fls17*RAM.o) .rodata)
```

このリンカスクリプトでは.updatable セクションに書換え対象となるデータを配置していますが、このようなセクションが不要な場合は.udatable セクションを削除してください。

# 改定履歴

| Date | revision | modification | note |
|---|---|---|---|
| 13/12/4 | 1.0 | 新規作成 | |
| 15/8/6 | 1.1 | GNU17V3 対応追加 | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

セイコーエプソン株式会社
マイクロデバイス事業部　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　〒541-0059　大阪市中央区博労町 3-5-1　エプソン大阪ビル 15F
TEL（06）6120-6000（代表）　FAX（06）6120-6100

ドキュメントコード：412668601
2013 年 12 月　作成
2015 年 8 月　改訂