

# 外部フラッシュ定義エディタ カスタムプログラム作成方法

RJJ06J0093-0100  
Rev.1.00  
2010.04.16

## 要旨

本アプリケーションノートは、外部フラッシュ定義エディタ（EFE）において指定可能なカスタムプログラムの作成要領についてまとめたものです。

## 目次

|                              |    |
|------------------------------|----|
| 1. 概要 .....                  | 3  |
| 2. 開発環境 .....                | 4  |
| 3. カスタムプログラム作成の流れと注意事項 ..... | 5  |
| 4. エミュレータソフトウェアの動作 .....     | 6  |
| 5. 関数 .....                  | 9  |
| 6. サンプルプログラム構成 .....         | 19 |

## 【略称】

本アプリケーションノートでは、以下の略称・略語を用います。

| 略称・略語           | 説明   |
|-----------------|--|
| EFE             | External Flash Definition Editor                   |
| RFD タブ          | EFE の RFD ファイル生成タブ                                 |
| USD タブ          | EFE の USD ファイル生成タブ                                 |
| RFD ファイル        | Renesas Flash Definition file (フラッシュメモリデバイス定義ファイル) |
| USD ファイル        | User System Definition file (ユーザシステム定義ファイル)        |
| 外部フラッシュメモリ      | マイコンの外部バス上に接続するフラッシュメモリデバイス                        |
| 外部フラッシュダウンロード機能 | 外部フラッシュメモリに対してデータをダウンロードする機能                       |
| 書き込みプログラム       | 外部フラッシュメモリへの書き込み処理プログラム                            |
| 標準プログラム         | EFE にプリインストールしている標準の書き込みプログラム                      |
| カスタムプログラム       | EFE で指定可能な非標準の書き込みプログラム                            |
| JEDEC 方式        | JEDEC 標準コマンド準拠フラッシュ書き込み方式                          |
| CUI 方式          | Intel/Sharp CUI コマンドフラッシュ書き込み方式                    |
| エミュレータ          | ルネサス製のエミュレータシステム                                   |
| HEW             | 統合開発環境 High-performance Embedded Workshop          |

## 1. 概要

外部フラッシュ定義エディタ（EFE）では、EFE標準の書き込みプログラムでは書き込みができないコマンドセットを持ったフラッシュメモリデバイスを想定し、ご自身でプログラミングされた書き込みプログラムを「カスタムプログラム」として指定することができます。（図 1-1参照）

本アプリケーションノートでは、カスタムプログラム作成方法について説明します。

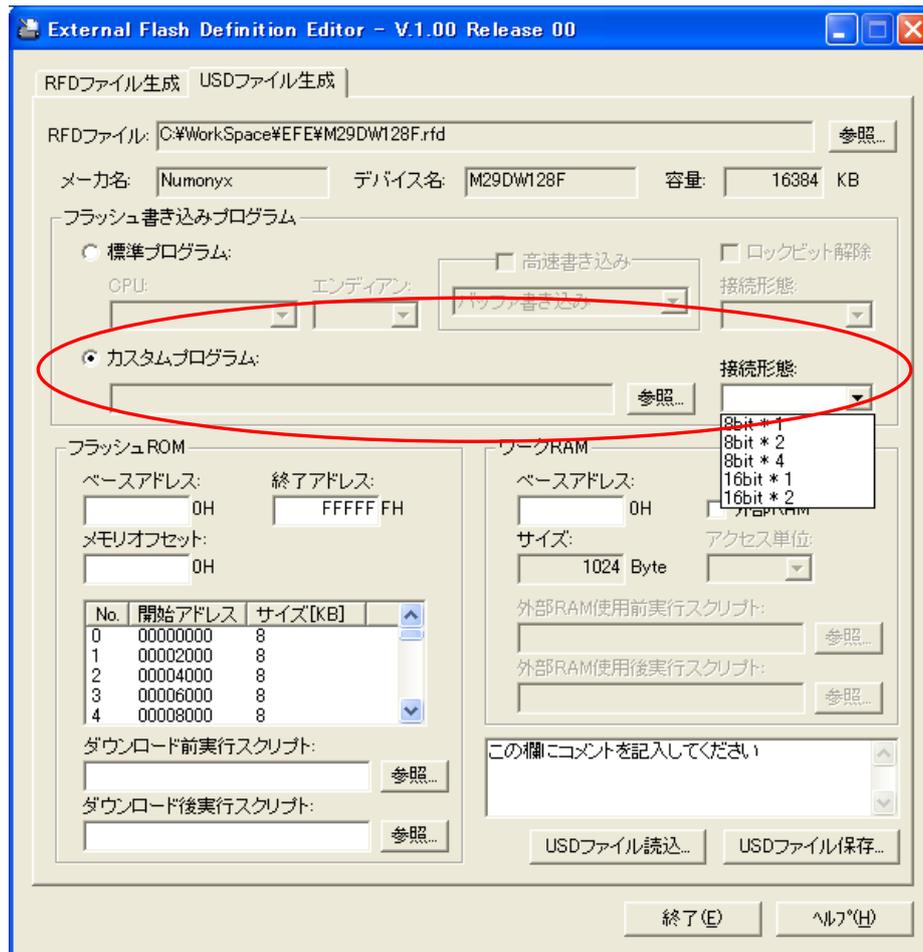


図 1-1 カスタムプログラム選択メニュー

## 2. 開発環境

カスタムプログラムの作成に必要な環境を以下に示します。

### 2.1 作成環境

#### ■書き込みプログラムサンプル

以下のファイルを EFE 製品サイトよりダウンロードしてください。

[http://japan.renesas.com/products/tools/emulation\\_debugging/onchip\\_debuggers/efe/efe.jsp](http://japan.renesas.com/products/tools/emulation_debugging/onchip_debuggers/efe/efe.jsp)

#### 1. サンプル C ソースプログラム

JEDEC 方式      ・ ・ ・ ・ EFE\_JEDEC\_Sample.c

CUI 方式        ・ ・ ・ ・ EFE\_CUI\_Sample.c

#### 2. ヘッダファイル      ・ ・ ・ ・ EFE\_fwif.h

#### 3. ライブラリファイル      ・ ・ ・ ・ EFE\_\*\*\*.lib      (\*\*\*は、マイコンファミリ名)

※ ライブラリファイルはマイコンファミリ毎に提供されます。  
サンプル C ソースプログラム、およびヘッダファイルは全マイコンファミリで共通です。

#### ■C/C++コンパイラ, シミュレータ

ご使用のマイコンに対応するコンパイラパッケージをご用意ください。

### 2.2 デバッグ・評価環境

#### ■外部フラッシュ定義エディタ (EFE)

以下のサイトよりダウンロードしてインストールしてください。

[http://japan.renesas.com/products/tools/emulation\\_debugging/onchip\\_debuggers/efe/efe.jsp](http://japan.renesas.com/products/tools/emulation_debugging/onchip_debuggers/efe/efe.jsp)

#### ■エミュレータソフトウェア (デバッガ)

ご使用のエミュレータに適合するものをダウンロードしてインストールしてください。

#### ■エミュレータ

#### ■ターゲットシステム (外部フラッシュメモリ搭載)

### 3. カスタムプログラム作成の流れと注意事項

#### 3.1 作成の流れ

カスタムプログラムの作成は以下の手順に沿って行ってください。

1. サンプルCソースプログラムをベースとして、修正が必要な関数をカスタマイズします。
2. ヘッダファイルとライブラリファイルをリンクしてビルドします。
3. EFEを使用し、RFDファイルおよびUSDファイルを作成します。
4. エミュレータソフトウェアにUSDファイルを登録し、外部フラッシュメモリに対するダウンロードが正常に動作することを実機にて確認します。

#### 3.2 作成上の注意事項

以下にカスタムプログラム作成する上での注意事項を示します。

- 提供サンプル（サンプルソースプログラムファイル、ヘッダファイル、ライブラリファイル）のうち、サンプルCソースプログラムに定義されている関数の処理をカスタマイズ可能です。
- 関数の実行を終了する時には、必ずReturn\_Result関数を使用して、規定の実行ステータスを返送してください。
- カスタムプログラムの占有メモリサイズ（スタックを除く）は、8192バイト（2000hバイト）以下としてください。
- カスタムプログラムが使用するスタックサイズは、256バイトを越えないようにしてください。

#### 4. エミュレータソフトウェアの動作

外部フラッシュメモリへのダウンロード動作は、エミュレータソフトウェアとフラッシュ書き込みプログラムが連携して処理していきます。カスタムプログラムを作成する上で必要となるエミュレータソフトウェアの動作について説明します。

##### 4.1 ワーク RAM 領域

エミュレータソフトウェアは、外部フラッシュメモリへのダウンロード要求が発生すると、外部フラッシュメモリ書き込みプログラム配置用のワーク領域を RAM 上に確保します。

尚、RAM 上の元データはエミュレータソフトウェアが事前にバックアップを行い、ダウンロード処理完了後に復元します。

ワークRAM領域の構成を、

図 4-1に示します。

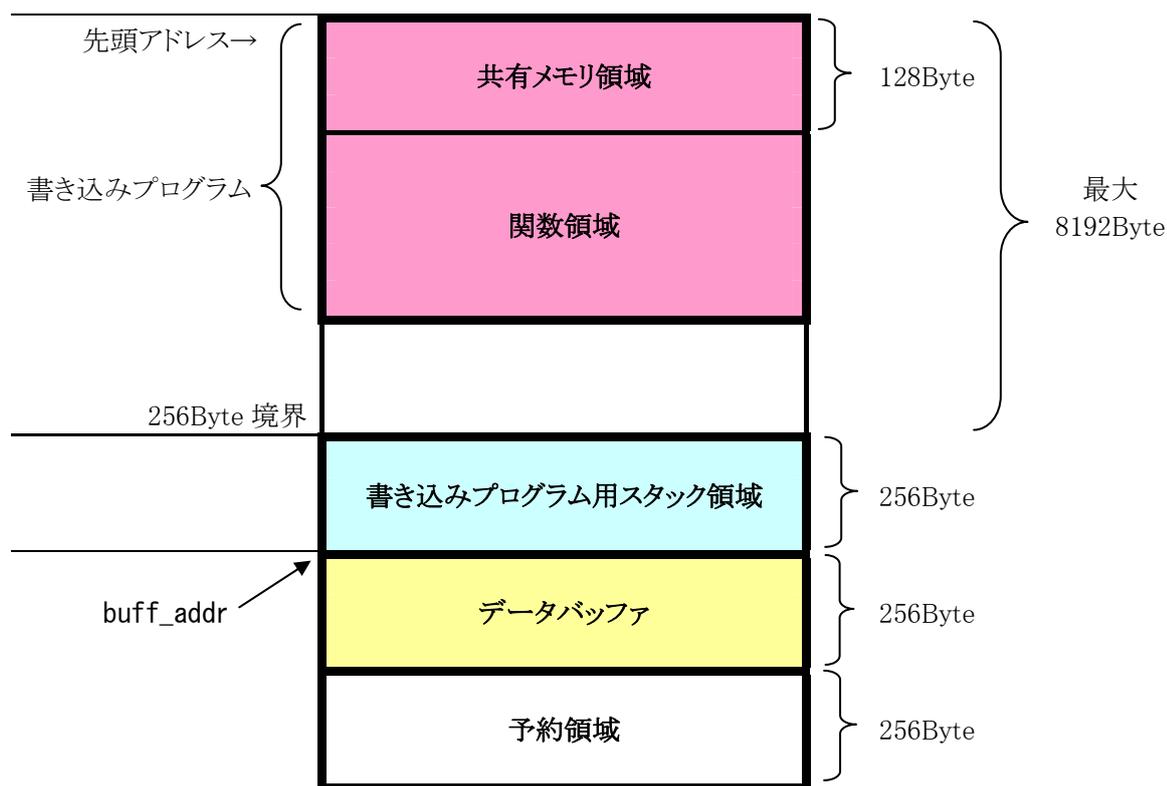


図 4-1 ワーク RAM 領域

## 4.2 パラメータの受け渡し

### 4.2.1 エミュレータソフトウェアから書き込みプログラム

エミュレータソフトウェアは、書き込みプログラムの関数を実行する前に使用する制御パラメータをワーク RAM 上の共有メモリ領域に格納します。

その後エミュレータソフトウェアが書き込みプログラムの実行を開始すると、ライブラリ内の `Set_Parameter()` という関数が実行され、この関数によって共有メモリ領域のパラメータが書き込みプログラムのグローバル変数へと受け渡されます。

### 4.2.2 書き込みプログラムからエミュレータソフトウェア

書き込みプログラムの各関数は `return` 前に `Return_Result` 関数を呼び出し、引数として実行ステータスを渡します。

`Return_Result` 関数は、受け取った引数を共有領域に格納します。

エミュレータソフトウェアは、その共有領域を読み出して実行ステータスを確認します。

## 4.3 エミュレータソフトウェアによる実行制御

フラッシュメモリ書き込み処理におけるエミュレータソフトウェアの制御フローを図 4-2に示します。

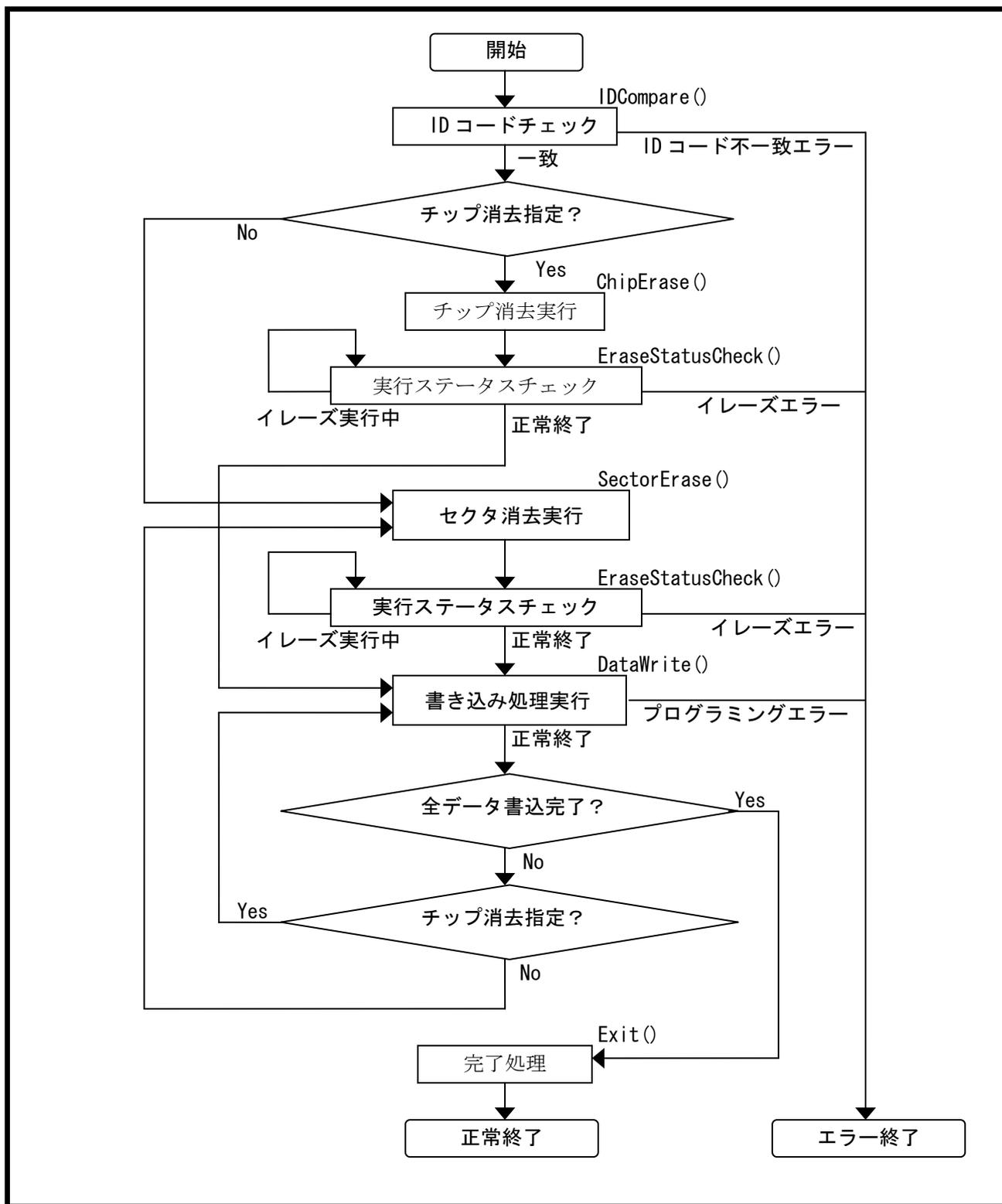


図 4-2 フラッシュメモリ書き込み制御フロー

## 5. 関数

### 5.1 関数一覧

カスタムプログラムで使用する関数の一覧を表 5.1に示します。

表 5.1 関数一覧

| No | 関数名                         | 機能                    |
|----|-----------------------------|-----------------------|
| 1  | void Return_Result(int)     | No. 2~No. 7 の実行結果の返送  |
| 2  | void IDCompare(void)        | メーカーID, デバイス ID のチェック |
| 3  | void ChipErase(void)        | チップ全体消去コマンドの発行        |
| 4  | void SectorErase(void)      | セクタ消去コマンドの発行          |
| 5  | void EraseStatusCheck(void) | 消去コマンド実行後のステータスレジスタ参照 |
| 6  | void Exit(void)             | 消去完了コマンドの発行           |
| 7  | void DataWrite(void)        | プログラミングコマンドの発行        |

## 5.2 制御パラメータ

カスタムプログラムの関数で使用する制御パラメータについて説明します。

制御パラメータは、ヘッダファイル”EFE\_fwif.h”に define 文、あるいはグローバル変数として定義されています。

```
#define TRUE 1
#define FALSE 0

#define BUF_SIZE 0x100 // Data Buffer Size

#define MODE_32x1 0
#define MODE_16x2 1
#define MODE_8x4 2
#define MODE_16x1 3
#define MODE_8x2 4
#define MODE_8x1 5

#define CODE_OK 0x00 // Normal Complete Code
#define CODE_ERASE_EXE 0x00 // Erase Execute Code
#define CODE_ERASE_OK 0x01 // Erase Complete Code
#define CODE_ERASE_ERR 0x02 // Erase Error Code
#define CODE_PROGRAM_ERR 0x02 // Program Error Code
#define CODE_ID_ERR 0x03 // ID Error Code

extern volatile unsigned long* write_addr; // Write Address
extern unsigned long base_addr; // Sector Address
extern unsigned long buff_addr; // Buffer Address
extern unsigned long wbuf_size; // Write-Buffer size of Flash Memory
extern unsigned long connect; // Connection between MCU and External Flash Memory
extern unsigned long maker_id; // Manufacturer ID
extern unsigned long device_id; // Device ID
extern unsigned long com8bit_0; // JEDEC 1'st Command at 8bits-bus mode of Flash Memory
extern unsigned long com8bit_1; // JEDEC 2'nd Command at 8bits-bus mode of Flash Memory
extern unsigned long com16bit_0; // JEDEC 1'st Command at 16bits-bus mode of Flash Memory
extern unsigned long com16bit_1; // JEDEC 2'nd Command at 16bits-bus mode of Flash Memory
extern unsigned long com32bit_0; // JEDEC 1'st Command at 32bits-bus mode of Flash Memory
extern unsigned long com32bit_1; // JEDEC 2'nd Command at 32bits-bus mode of Flash Memory

extern void Return_Result(int);
```

図 5-1 EFE\_fwif.h

## 5.2.1 制御パラメータ（固定値）一覧

表 5.2 define 定義(固定値)の制御パラメータ

| No | パラメータ            | 値     | 分類                                 | 意味                                |
|----|------------------|-------|------------------------------------|-----------------------------------|
| 1  | TRUE             | 1     | 汎用                                 | 真                                 |
| 2  | FALSE            | 0     |                                    | 偽                                 |
| 3  | BUF_SIZE         | 0x100 | データバッファサイズ                         | ダウンロードデータの<br>一時格納領域サイズ           |
| 4  | MODE_32x1        | 0     | マイコンと外部フラッシュ<br>メモリ間のデータバス接続<br>形態 | マイコン 32bit<br><-> フラッシュメモリ 32bit  |
| 5  | MODE_16x2        | 1     |                                    | マイコン 32bit<br><->フラッシュメモリ 16bit×2 |
| 6  | MODE_8x4         | 2     |                                    | マイコン 32bit<br><->フラッシュメモリ 8bit×4  |
| 7  | MODE_16x1        | 3     |                                    | マイコン 16bit<br><->フラッシュメモリ 16bit   |
| 8  | MODE_8x2         | 4     |                                    | マイコン 16bit<br><->フラッシュメモリ 8bit×2  |
| 9  | MODE_8x1         | 5     |                                    | マイコン 8bit<br><->フラッシュメモリ 8bit     |
| 10 | CODE_OK          | 0x00  |                                    | 関数の実行ステータス                        |
| 11 | CODE_ERASE_EXE   | 0x00  | イレーズ実行中                            |                                   |
| 12 | CODE_ERASE_OK    | 0x01  | イレーズ正常終了                           |                                   |
| 13 | CODE_ERASE_ERR   | 0x02  | イレーズエラー                            |                                   |
| 14 | CODE_PROGRAM_ERR | 0x02  | プログラミングエラー                         |                                   |
| 15 | CODE_ID_ERR      | 0x03  | ID コード不一致エラー                       |                                   |

## 5.2.2 制御パラメータ（グローバル変数）一覧

表 5.3 グローバル変数定義の制御パラメータ

| No | パラメータ      | 型                           | 機能                             |
|----|------------|-----------------------------|--------------------------------|
| 1  | write_addr | volatile<br>unsigned long * | ライトアドレス                        |
| 2  | base_addr  | unsigned long               | 対象セクタの先頭アドレス                   |
| 3  | buff_addr  | unsigned long               | データバッファ先頭アドレス                  |
| 4  | wbuf_size  | unsigned long               | フラッシュメモリのライトバッファサイズ            |
| 5  | connect    | unsigned long               | マイコンと外部フラッシュメモリ間の<br>データバス接続形態 |
| 6  | maker_id   | unsigned long               | フラッシュメモリのメーカ ID                |
| 7  | device_id  | unsigned long               | フラッシュメモリのデバイス ID               |
| 8  | com8bit_0  | unsigned long               | JEDEC コマンドパターン                 |
| 9  | com8bit_1  | unsigned long               |                                |
| 10 | com16bit_0 | unsigned long               |                                |
| 11 | com16bit_1 | unsigned long               |                                |
| 12 | com32bit_0 | unsigned long               |                                |
| 13 | com32bit_1 | unsigned long               |                                |

## 5.2.3 グローバル変数の機能説明

## ■ write\_addr

データバッファに転送した 256 バイトデータの書き込み先頭アドレス。  
エミュレータソフトウェアが図 4-2の制御フローに沿って動的に制御します。

## ■ base\_addr

イレーズ、又は書き込み対象セクタの先頭アドレスが受け渡されます。  
エミュレータソフトウェアが図 4-2の制御フローに沿って動的に制御します。

## ■ buff\_addr

ダウンロードデータをバッファリングする領域の先頭アドレスが受け渡されます。

## ■ wbuf\_size

バッファ書き込み対応フラッシュメモリ用の制御パラメータです。  
フラッシュメモリ内蔵バッファのサイズが受け渡されます。  
(EFE の RFD タブで指定する[バッファサイズ]そのものです)  
バッファ書き込みモードを指定した場合に本パラメータを使用します。

### ■connect

マイコンと外部フラッシュメモリ間の接続形態が受け渡されます。

(EFE の RFD タブで指定する[カスタムプログラムの接続形態]そのものです)

表 5.4 接続形態

| 値 | パラメータ     | マイコン外部バス設定 | 外部フラッシュメモリ |
|---|-----------|------------|------------|
| 0 | MODE_32x1 | 32bit      | 32bit×1    |
| 1 | MODE_16x2 | 32bit      | 16bit×2    |
| 2 | MODE_8x4  | 32bit      | 8bit×4     |
| 3 | MODE_16x1 | 16bit      | 16bit×1    |
| 4 | MODE_8x2  | 16bit      | 8bit×2     |
| 5 | MOED_8x1  | 8bit       | 8bit×1     |

### ■maker\_id

フラッシュメモリのメーカー ID が受け渡されます。

(EFE の RFD タブで指定する[メーカー ID]そのものです)

外部フラッシュメモリとの接続チェックに使用します。

### ■device\_id

フラッシュメモリのデバイス ID が受け渡されます。

(EFE の RFD タブで指定する[デバイス ID]そのものです)

外部フラッシュメモリとの接続チェックに使用します。

### ■com8bit\_0/com8bit\_1/com16bit\_0/com16bit\_1/com32bit\_0/com32bit\_1

JEDEC 方式のフラッシュメモリへの書き込みに使用するパラメータです。

(EFE の RFD タブで指定するバス幅ごとの[1st Address][2nd Address]そのものです)

JEDEC 方式の書き込みでは、まず初めに 2 回の Unlock サイクルを実行します。

1st Address を Unlock サイクル 1 回目、2nd Address を Unlock サイクル 2 回目で使用します。

実際に発行するコマンドアドレスは、外部フラッシュメモリの接続形態に応じて、以下のように演算します。

```

32bit×1: base_addr + com32bit0/1 << 2
16bit×2: base_addr + com16bit0/1 << 2
16bit×1: base_addr + com16bit0/1 << 1
8bit×4:  base_addr + com8bit0/1 << 2
8bit×2:  base_addr + com8bit0/1 << 1
8bit×1:  base_addr + com8bit_0/1

```

## 5.3 関数機能

カスタムプログラムで使用する関数の機能について説明します。

### 5.3.1 実行ステータスリターン関数

**【機能】**

5.3.2～5.3.7の関数の実行ステータスを、特定のアドレス領域を介してエミュレータシステムへ通知します。

関数の実体は、ライブラリファイルに含まれます。

**【関数名】**        **void    Return\_Result(int)**

**【引数】**            各関数で定義されている本関数への引数  
                      ※5.3.2～5.3.7の各関数で定義されている本関数への引数以外が返送された場合、  
                      正常に動作しません。

**【戻り値】**        なし

**【動作】**

5.3.2～5.3.7の関数処理の最後に、実行ステータスを引数として本関数を呼び出します。

本関数では、受け取った実行ステータスを特定のアドレス領域に格納します。

エミュレータソフトウェアは、実行ステータスを読み出し、以降の制御動作を決定します。

### 5.3.2 IDチェック関数

**【機能】**

外部フラッシュメモリに対する正常なアクセスが可能な状態であることをチェックするための関数です。ダウンロード処理の最初に実行されます。

**【関数名】**        **void    IDCompare(void)**

**【引数】**            なし

**【戻り値】**        なし

**【Return\_Result 関数への引数】**

**CODE\_OK**            …… 正常終了  
**CODE\_ID\_ERR** …… IDコード不一致エラー  
※上記以外のコードは返送しないでください。

**【動作】**

外部フラッシュメモリ内のレジスタへアクセスし、メーカーIDとデバイスIDを読み出します。読み出したID値と期待値（EFEのRFDタブで入力した値）の一致をチェックします。

値が一致した場合（正常終了の場合）、エミュレータソフトウェアは外部フラッシュメモリに対するアクセスにおいて接続設定（\*1）に問題がないと判断し、以降の処理に進みます。

値が不一致の場合（IDコード不一致エラーの場合）、エミュレータソフトウェアは以降の処理を中止します。ハードウェアでのマイコン-外部フラッシュメモリ間の接続、外部バスコントローラ設定、マイコンクロック設定、外部バスクロック設定等に誤り、漏れがないか確認ください。

### 5.3.3 セクタイレース関数

**【機能】**

1つのセクタを消去する関数です。セクタへのダウンロードデータ書き込み前に実行されます。

**【関数名】**        **void    ChipErase(void)**

**【引数】**            なし

**【戻り値】**        なし

**【Return\_Result 関数への引数】**

**CODE\_OK**    . . . 正常終了

※上記以外のコードは返送しないでください。

**【動作】**

セクタイレース（ブロックイレース）コマンドを発行します。

完了確認は行いません。（イレースステータスチェック関数にて確認します。）

イレースに先立ってロックビット解除（プロテクト解除）の必要がある場合には、イレースコマンド発行の前にロックビット解除処理を追加してください。

## 5.3.4 チップイレーズ関数

## 【機能】

全てのセクタを消去する関数です。

E1/E20 エミュレータデバッガの場合、[コンフィグレーションプロパティダイアログ>外部フラッシュメモリタブ>外部フラッシュの上書きタブ] で「全セクタ消去後に書き込む」を選択している場合 (図 5-2参照) に呼び出されます。

【関数名】        **void    ChipErase(void)**

【引数】         なし

【戻り値】       なし

【Return\_Result 関数への引数】

**CODE\_OK**    . . . 正常終了

※上記以外のコードは返送しないでください。

## 【動作】

JEDEC 方式の場合には、チップイレーズコマンドを発行します。

完了確認は行いません。(イレーズステータスチェック関数にて確認します。)

なお、CUI 方式の場合には、チップイレーズの定義がないため、セクタイレーズコマンドを全セクタ領域に対して繰り返し実行して、全体の消去を行います。

イレーズに先立ってロックビット解除 (プロテクト解除) の必要がある場合には、イレーズコマンド発行の前にロックビット解除処理を追加してください。

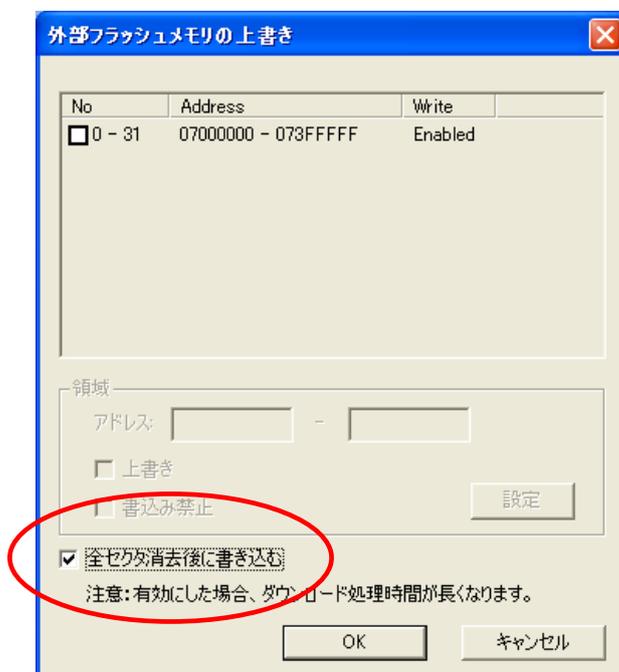


図 5-2 全セクタ消去後書き込みオプション

### 5.3.5 イレーズステータスチェック関数

**【機能】**

チップイレーズ関数、およびセクタイレーズ関数の処理ステータスをチェックする関数です。  
チップイレーズ関数、セクタイレーズ関数の後に実行されます。

**【関数名】**        **void EraseStatusCheck(void)**

**【引数】**         なし

**【戻り値】**        なし

**【Return\_Result 関数への引数】**

**CODE\_ERASE\_OK**        ・・・・正常終了  
**CODE\_ERASE\_EXE**      ・・・・イレーズ実行中  
**CODE\_ERASE\_ERR**      ・・・・イレーズエラー  
※上記以外のコードは返送しないでください。

**【動作】**

外部フラッシュメモリのステータスレジスタを参照します。  
レジスタのビット状態により、  
    [正常終了] [イレーズ実行中] [イレーズエラー]  
のいずれかを判断します。  
イレーズが正常終了した場合、エミュレータソフトウェアは以降の処理に進みます。  
イレーズ実行中の場合、エミュレータソフトウェアは本関数を再度実行します。  
イレーズエラーが発生した場合は、エミュレータソフトウェアは以降の処理を中止します。

### 5.3.6 シーケンスクリア関数

**【機能】**

コマンドシーケンスをリセットする関数です。

**【関数名】**        **void Exit(void)**

**【引数】**         なし

**【戻り値】**        なし

**【Return\_Result 関数への引数】**

**CODE\_OK**            ・・・・正常終了  
※上記以外のコードは返送しないでください。

**【動作】**

イレーズコマンド終了後に、コマンドシーケンスをクリアするために使用します。



## 6. サンプルプログラム構成

### 6.1 サンプルプログラムのフォルダ構成

サンプルプログラムのワークスペース以下のフォルダ構成を図 6-1に示します。

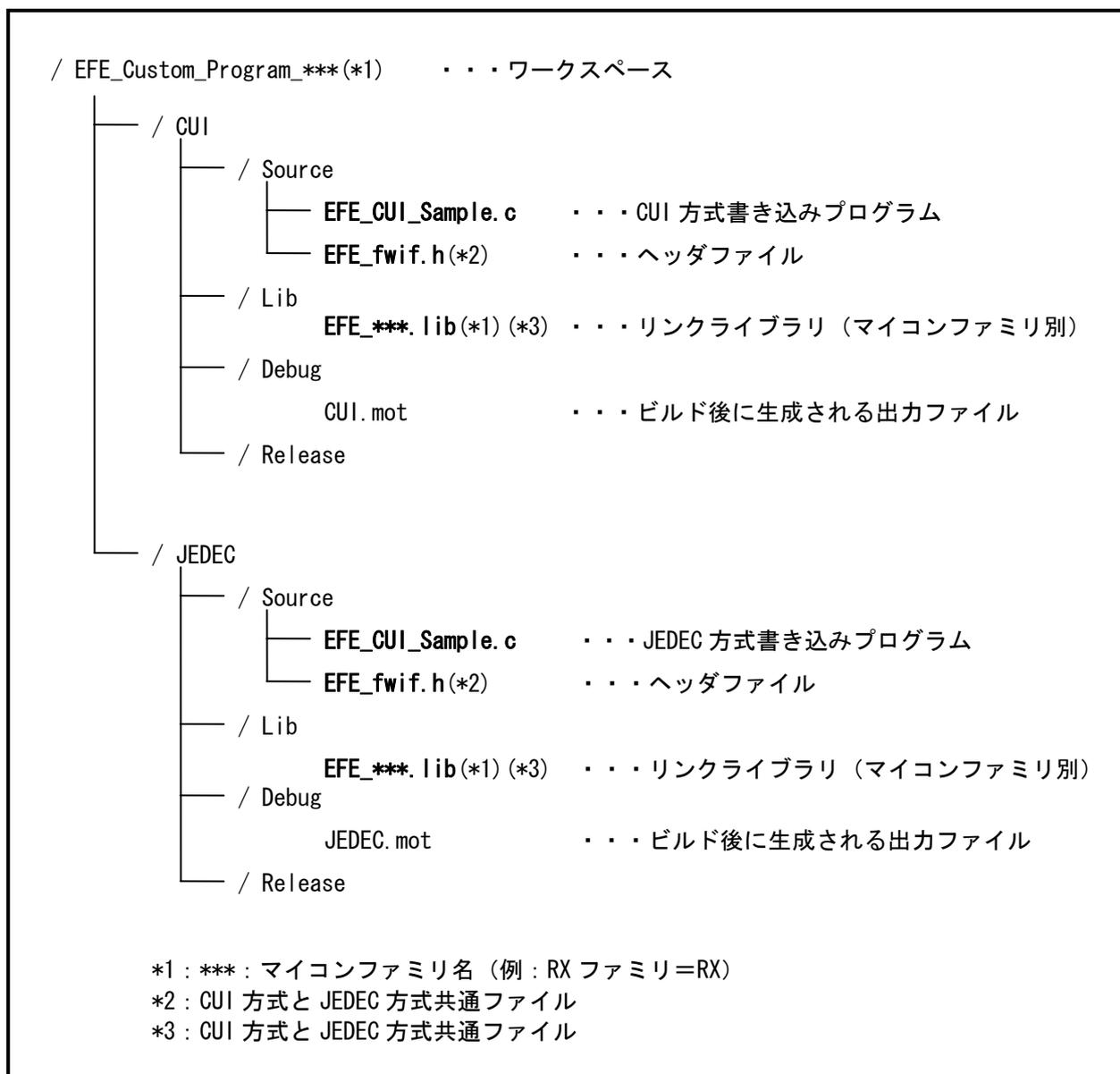


図 6-1 サンプルプログラムワークスペースのフォルダ構成

## 6.2 ワークスペースウィンドウ

サンプルプログラムのHEWワークスペースウィンドウ構成を図 6-2に示します。

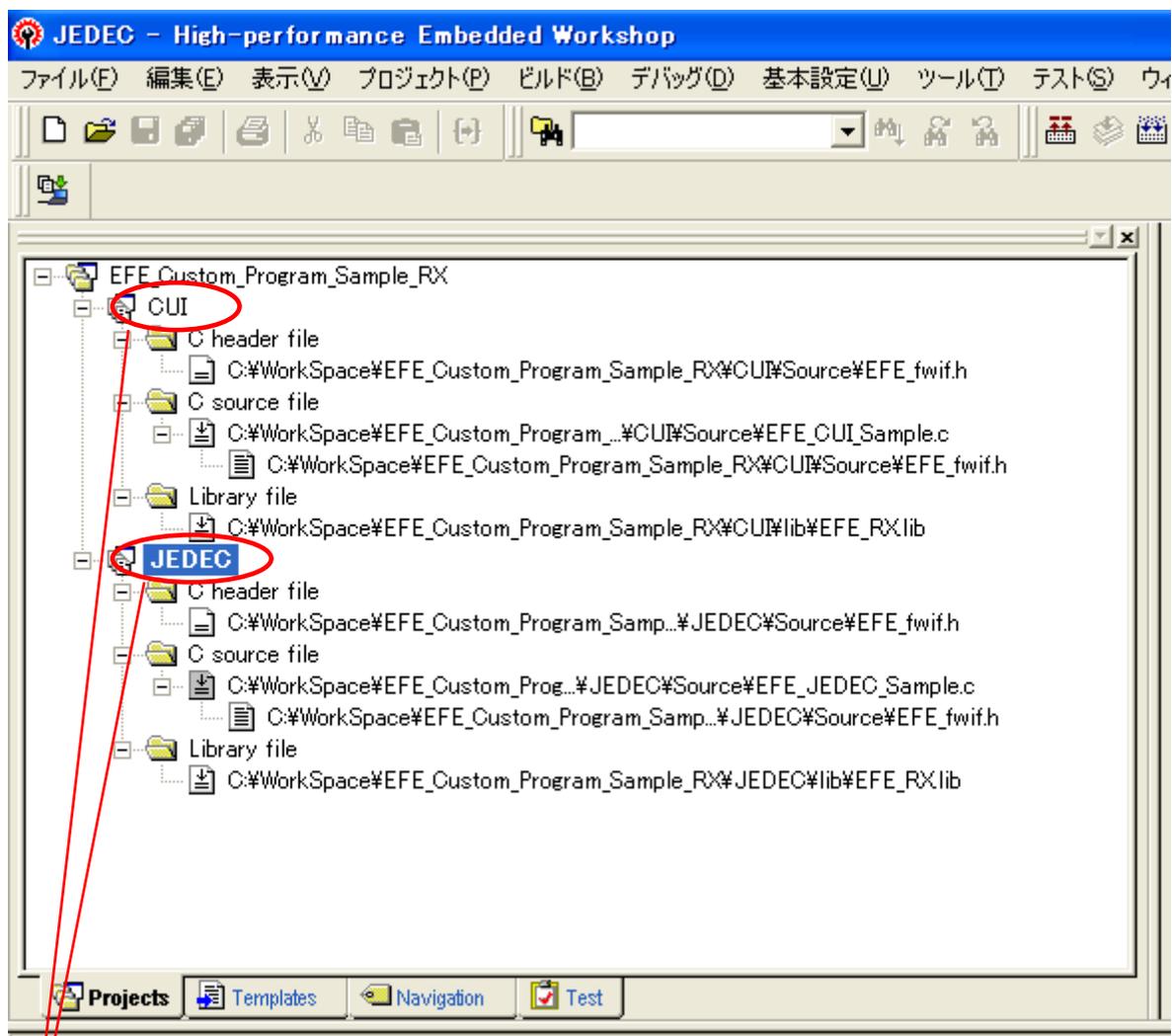


図 6-2 ワークスペースウィンドウ

### プロジェクト

作成対象の書き込み方式をアクティブ化して、ビルドを実行します。

(対象プロジェクト名で右クリック > “アクティブプロジェクトに設定” を選択後、ビルド実行)

### 6.3 リンカオプション

ビルド時に必要なリンカオプションについて説明します。

なお、サンプルプログラムのワークスペースはオプション適用済みです。

#### ■セクション

以下のオプションを指定してください。

`-start=DATA/03000, PROG,P,B/0x3080`

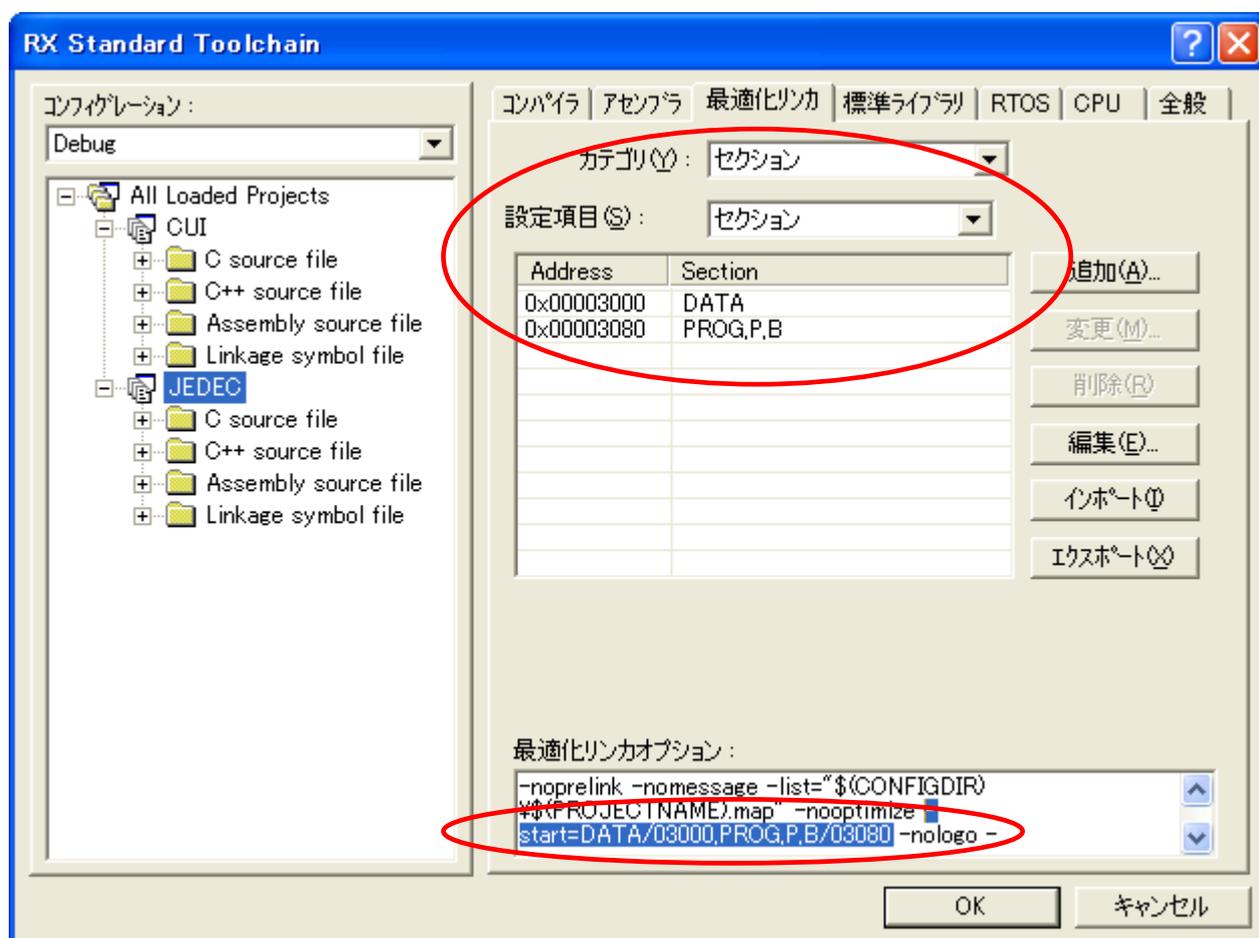


図 6-3 セクション指定

#### ■ROM 化支援オプション

ROM 化支援オプションは外して下さい。

`-rom=.....` ←削除

## ■ 出力ファイル指定

出力ファイル形式としてモトローラ S フォーマットを指定します。

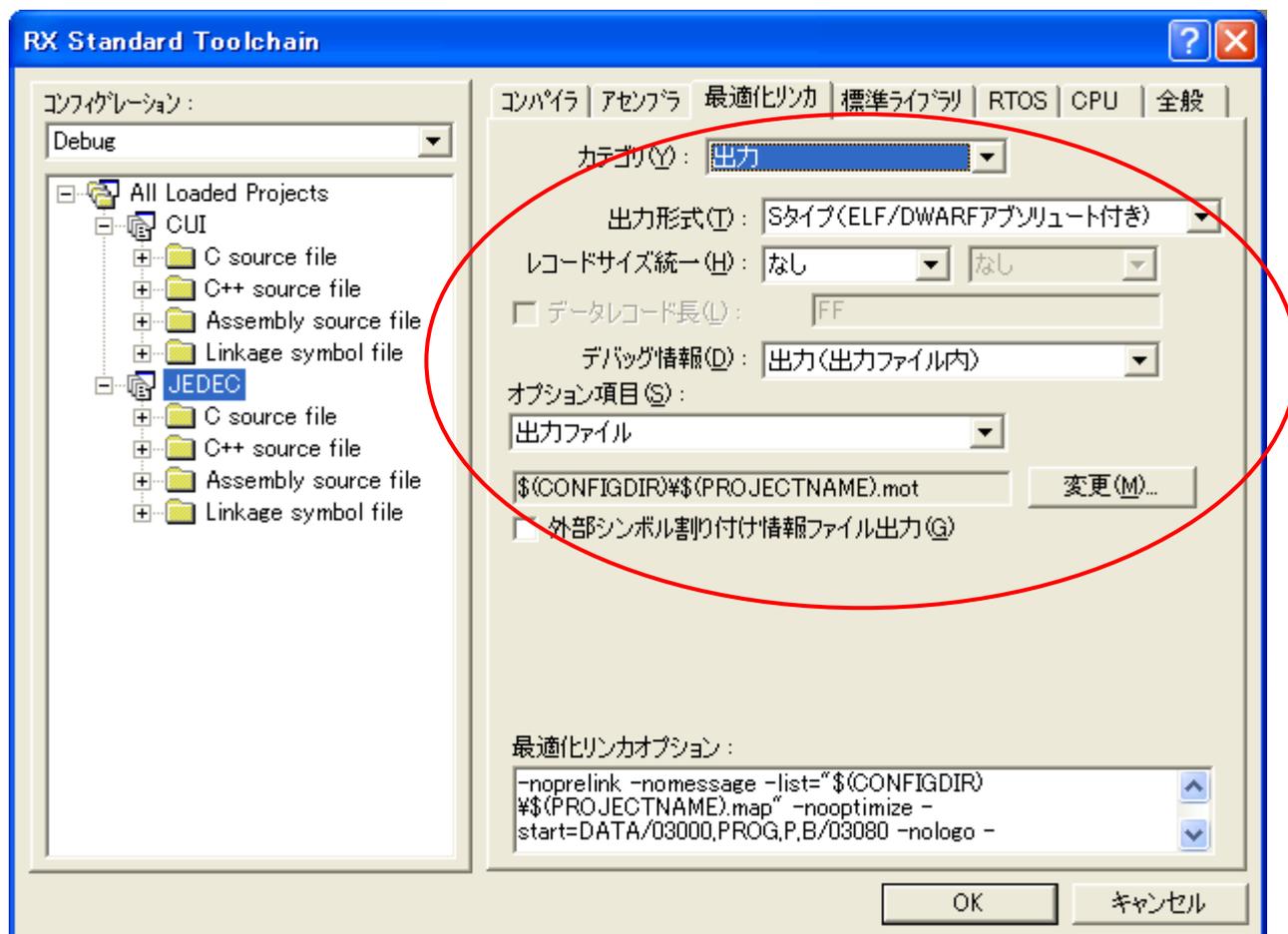


図 6-4 出力ファイルフォーマット

## ホームページとサポート窓口

ルネサス エレクトロニクス ツールホームページ

<http://japan.renesas.com/tools>

お問い合わせ先

<http://japan.renesas.com/inquiry>

## 改訂記録

| Rev. | 発行日        | 改訂内容 |      |
|------|------------|------|------|
|      |            | ページ  | ポイント |
| 1.00 | 2010.04.16 | —    | 初版発行 |

すべての商標および登録商標は、それぞれの所有者に帰属します。

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただけますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報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。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事情報の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<http://japan.renesas.com/inquiry>