

この度は、EEPROM エミュレーション・ライブラリ Pack02 パッケージ Ver.2.00 をご使用いただきまして、誠にありがとうございます。

この添付資料では、EEPROM エミュレーション・ライブラリ Pack02 パッケージ Ver.2.00 をお使いいただく上での制限事項および注意事項等を記載しております。ご使用前に、必ずお読みくださいますようお願い申し上げます。

目次

第1章 対象製品	2
第2章 ユーザーズマニュアル	2
第3章 バージョンアップ内容	3
第4章 対応ツール	3
第5章 インストール	4
5.1 インストール方法	4
5.2 アンインストール方法	4
5.3 ファイル構成	5
第6章 ビルド方法	6
6.1 使用するソフトウェア	6
6.2 CS+(旧 CubeSuite+)でのビルド方法	6
6.2.1 C言語の場合	6
6.2.2 アセンブリ言語の場合	9
6.3 ビルド時の注意事項	12
6.3.1 CA78K0R コンパイラ使用時	12
6.3.2 CC-RL コンパイラ使用時	13
第7章 デバッグ方法	14
7.1 デバッグ時の注意事項	14
第8章 サンプルプログラム	15
8.1 サンプルプログラムの初期設定	15
8.2 オプション・バイトとオンチップ・デバッグの設定について	16
8.3 内蔵RAM領域の定義	17
8.3.1 CA78K0R コンパイラ使用時	17
8.3.2 CC-RL コンパイラ使用時	18

第1章 対象製品

EEPROM エミュレーション・ライブラリ Pack02 パッケージ Ver.2.00 は、「RL78 ファミリ CA78K0R コンパイラ用 EEPROM エミュレーション・ライブラリ Pack02 Ver.1.01 (JP_R_EEL_RL78_P02_V1.01_B_J.zip)」と「RL78 ファミリ CC-RL コンパイラ用 EEPROM エミュレーション・ライブラリ Pack02 Ver.1.01 (JP_R_EEL_RL78_P02_V1.01_CCRL_B_J.zip)」の統合パッケージです。

本リリースノートの対象製品を示します。

製品名	Ver.	インストーラ ファイル名	Ver.
RL78 ファミリ CA78K0R コンパイラ用 EEPROM エミュレーション・ライブラリ Pack02	V1.01	RENESAS_RL78_EEL-FDL_T02_PACK02_ 2V00.exe	V2.00
RL78 ファミリ CC-RL コンパイラ用 EEPROM エミュレーション・ライブラリ Pack02	V1.01		

第2章 ユーザーズマニュアル

本バージョンは下記のユーザーズマニュアルに対応しています。

使用コンパイラ	マニュアル名	資料番号
RL78 ファミリ CA78K0R コンパイラ	RL78 ファミリ EEPROM エミュレーション・ライブラリ Pack02 ユーザーズマニュアル	R01US0068JJ0101
RL78 ファミリ CC-RL コンパイラ	RL78 ファミリ EEPROM エミュレーション・ライブラリ Pack02 ユーザーズマニュアル	R01US0068JJ0101
	RL78 ファミリ EEPROM エミュレーション・ライブラリ Pack02 CA78K0R コンパイラ版と CC-RL コンパイラ版の差分	R20UT3486JJ0100

第3章 バージョンアップ内容

本バージョンのバージョンアップ内容を示します。

No.	パッケージ Ver.	対象	内容
1	V2.00	CA78K0R コンパイラ用 EEPOM エミュレーション・ ライブラリ Pack02	ZIP ファイル(JP_R_EEL_RL78_P02_V1.01_B_J)からライブラリ本体の変更はありません。
		CC-RL コンパイラ用 EEPOM エミュレーション・ ライブラリ Pack02	ZIP ファイル(JP_R_EEL_RL78_P02_V1.01_CCRL_B_J)からライブラリ本体の変更はありません。
		CA78K0R コンパイラ用 EEPOM エミュレーション・ ライブラリ Pack02 ユーザーズマニュアル	・ユーザーズマニュアル(R01US0068JJ0101)から変更はありません。
		CC-RL コンパイラ用 EEPOM エミュレーション・ ライブラリ Pack02 ユーザーズマニュアル	・ユーザーズマニュアル(R01US0068JJ0101)から変更はありません。 ・CA78K0R コンパイラ版と CC-RL コンパイラ版の差分 (R20UT3486JJ0100)から変更はありません。

第4章 対応ツール

EEPROM エミュレーション・ライブラリ Pack02 を使用する場合には、下記のバージョンを使用してください。

対象ライブラリ	ツール名	バージョン
CA78K0R コンパイラ用 ライブラリ	統合開発環境 : CubeSuite+	V1.00.00 以降
	統合開発環境 : CS+	V3.00.00 以降
CC-RL コンパイラ用 ライブラリ	統合開発環境 : CS+	V3.01.00 以降

第5章 インストール

この章では、EEPROMエミュレーション・ライブラリ Pack02 パッケージ V2.00のインストールとアンインストールの手順について説明します。

5.1 インストール方法

EEPROMエミュレーション・ライブラリ Pack02のインストールは次の手順で行います。

- (1) Windowsを起動します。
- (2) EEPROMエミュレーション・ライブラリ Pack02 パッケージの圧縮ファイルを解凍、インストーラを実行します。
- (3) プルダウンメニューから"Asia/Oceania - Japanese"を選択します。
- (4) "OK"ボタンを押下し、以降、インストーラの指示に従い、実行します。



5.2 アンインストール方法

EEPROMエミュレーション・ライブラリ Pack02のアンインストールは次の手順で行います。

- (1) Windowsを起動します。
- (2) ユーザ任意の場所に配置したEEPROMエミュレーション・ライブラリが入っているフォルダを削除します。

5.3 ファイル構成

EEPROMエミュレーション・ライブラリ Pack02 パッケージのインストーラを実行することによって、展開されるファイルの構成は、次のとおりです。

配置先フォルダ	
r20ut2465jjxxxx_rl78.pdf	: リリースノート(本書)
support.txt	: EEL のサポート情報
CA78K0R_110 or CCRL_100	
doc	
r01us0068jjxxxx_rl78.pdf	: ユーザーズマニュアル
r20ut3486jjxxxx_rl78.pdf	: 差分マニュアル(CC-RL のみ)
lib	
eel.lib	: EEPROM エミュレーション・ライブラリ(EEL)
fdl.lib	: データ・フラッシュ・ライブラリ(FDL)
eel.h	: C 言語用 EEL ヘッダファイル
eel.inc	: アセンブラ用 EEL ヘッダファイル
eel_types.h	: C 言語用 EEL 定義設定ヘッダファイル
eel_types.inc	: アセンブラ用 EEL 定義設定ヘッダファイル
fdl.h	: C 言語用 FDL ヘッダファイル
fdl.inc	: アセンブラ用 FDL ヘッダファイル
fdl_types.h	: C 言語用 FDL 定義設定ヘッダファイル
Sample	
asm	
eel_descriptor.inc	: EEL ディスクリプタヘッダファイル
eel_descriptor.asm	: EEL ディスクリプタソースファイル
eel_sample_linker_file.dr	: EEL サンプルディレクティブファイル(CA78K0R 版のみ)
fdl_descriptor.inc	: FDL ディスクリプタヘッダファイル
fdl_descriptor.asm	: FDL ディスクリプタソースファイル
C	
eel_descriptor.h	: EEL ディスクリプタヘッダファイル
eel_descriptor.c	: EEL ディスクリプタソースファイル
eel_user_types.h	: EEL ユーザ定義ヘッダファイル
eel_sample_linker_file.dr	: EEL サンプルディレクティブファイル(CA78K0R 版のみ)
fdl_descriptor.h	: FDL ディスクリプタヘッダファイル
fdl_descriptor.c	: FDL ディスクリプタソースファイル
r_eel_sample.c.c	: C 言語サンプルプログラムファイル(CA78K0R 版のみ)
r_eel_sample.c.dr	: C 言語サンプルリンクディレクティブファイル(CA78K0R 版のみ)

注1. xはバージョン番号、またはRev番号の為、省略しています。

注2. CA78K0R用でサンプルプログラムを使用する場合は、プログラムファイル(*.c)とリンクディレクティブファイル(*.dr)を一緒に組み込んでください。

CC-RL用では、リンクディレクティブファイル(*.dr)[リンク情報の設定ファイル]は添付されていません。

CC-RL用サンプルのリンク情報は、CS+のリンクの設定画面で指定してください。

第6章 ビルド方法

この章では、EEPROMエミュレーション・ライブラリを用いたプログラムのビルドの手順を説明します。

6.1 使用するソフトウェア

EEPROM エミュレーション・ライブラリを用いたプログラムをビルドするには、次の統合開発環境が必要です。

- ・ CA78K0R コンパイラ用：統合開発環境 CS+ V3.00.00 以降、または CubeSuite+ V1.00.00 以降
- ・ CC-RL コンパイラ用：統合開発環境 CS+ V3.01.00 以降

6.2 CS+(旧 CubeSuite+)でのビルド方法

CS+を用いて EEPROM エミュレーション・ライブラリ Pack02 をユーザプログラムに組み込んで、ビルドする手順を説明します。

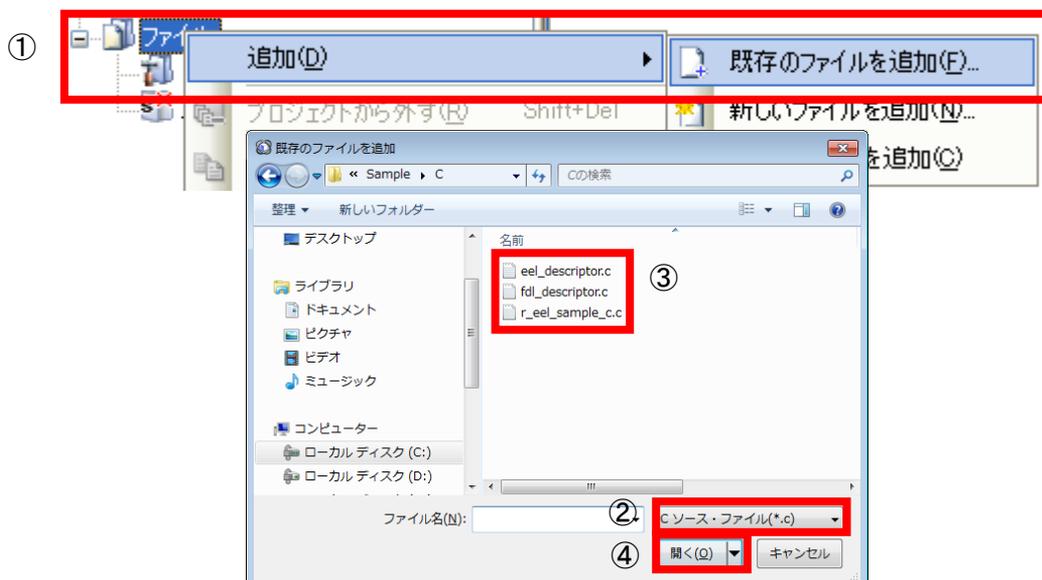
6.2.1 C言語の場合

(1) プロジェクトの作成とソースファイルの設定

CS+でプロジェクトを作成し、表示された画面の左側にある[プロジェクト・ツリー]から、[ファイル]を右クリックして表示されるリストの[追加]を選択し、[既存のファイルを追加]をクリックすると、[既存のファイルを追加]画面が表示されます（図 6-1）。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると、ファイルの種類の一覧が表示されるので、その中にある[C ソースファイル(*.c)]を選択し、ソースファイルとしてユーザプログラムのファイル(サンプルの場合は r_eel_sample.c.c)、及び EEPROM エミュレーション・ライブラリ、データ・フラッシュ・ライブラリのディスクリプタファイル (eel_descriptor.c, fdl_descriptor.c) を登録してください。

図 6-1 ユーザプログラムファイルの登録

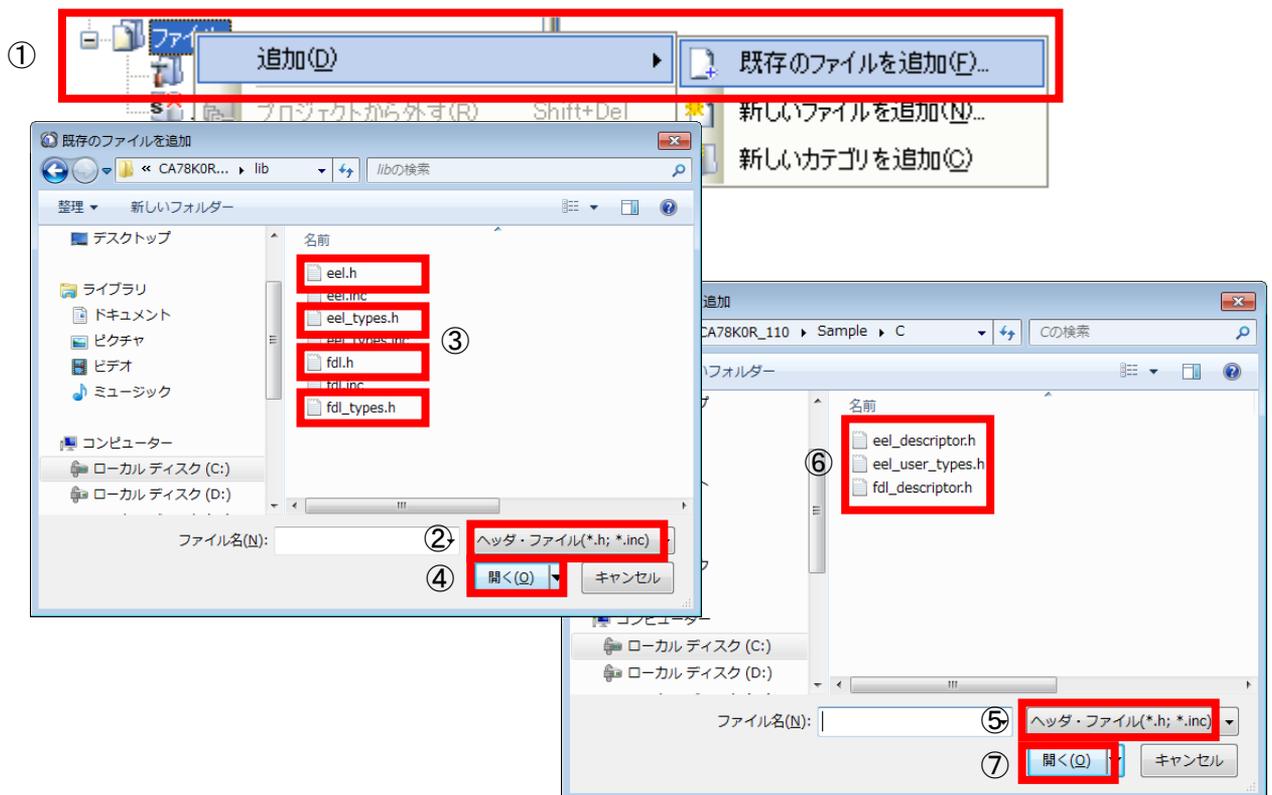


(2) インクルードファイルの設定

CS+の[プロジェクト・ツリー]にある、[ファイル]を右クリックして表示されるリストから[追加]を選択し、[既存のファイルを追加]をクリックすると、[既存のファイルを追加]画面が表示されます（図 6-2）。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると、ファイルの種類の一覧が表示されるので、その中にある[ヘッダファイル(*.h; *.inc)]を選択し、EEPROM エミュレーション・ライブラリ、データ・フラッシュ・ライブラリのヘッダファイル、及びディスクリプタ用ヘッダファイル (eel.h, eel_types.h, fdl.h, fdl_types.h, eel_descriptor.h, fdl_descriptor.h, eel_user_types.h) を登録してください。

図 6-2 インクルードファイルの登録

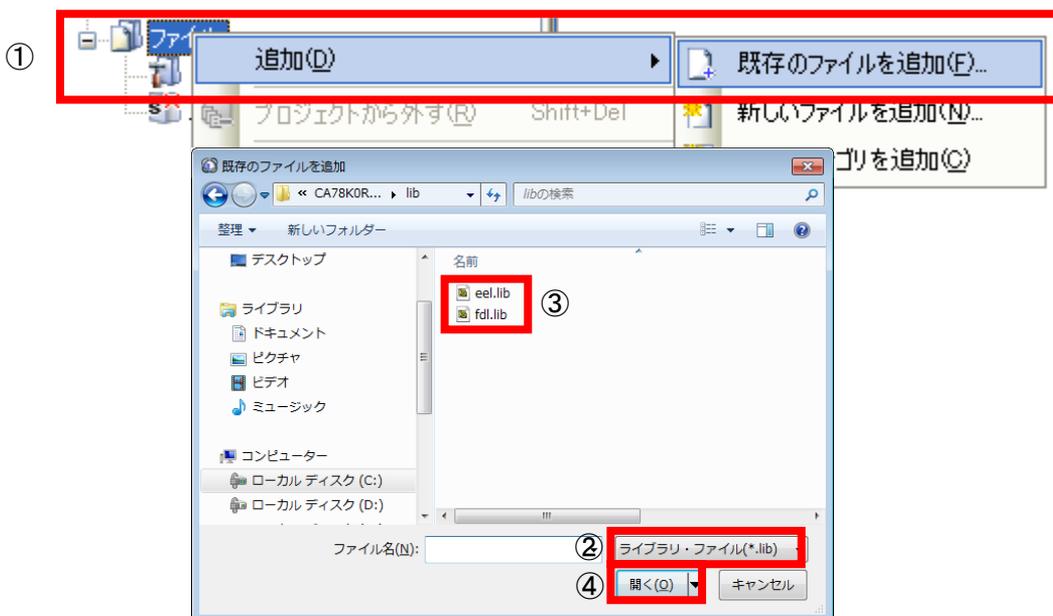


(3) ライブラリファイルの設定

CS+の[プロジェクト・ツリー]にある、[ファイル]を右クリックして表示されるリストから[追加]を選択し、[既存のファイルを追加]をクリックすると、[既存のファイルを追加]画面が表示されます（図 6-3）。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると、ファイルの種類の一覧が表示されるので、その中にある[ライブラリファイル(*.lib)]を選択し、EEPROM エミュレーション・ライブラリ、データ・フラッシュ・ライブラリ (eel.lib, fdl.lib) のファイルを登録してください。

図 6-3 ライブラリファイルの登録

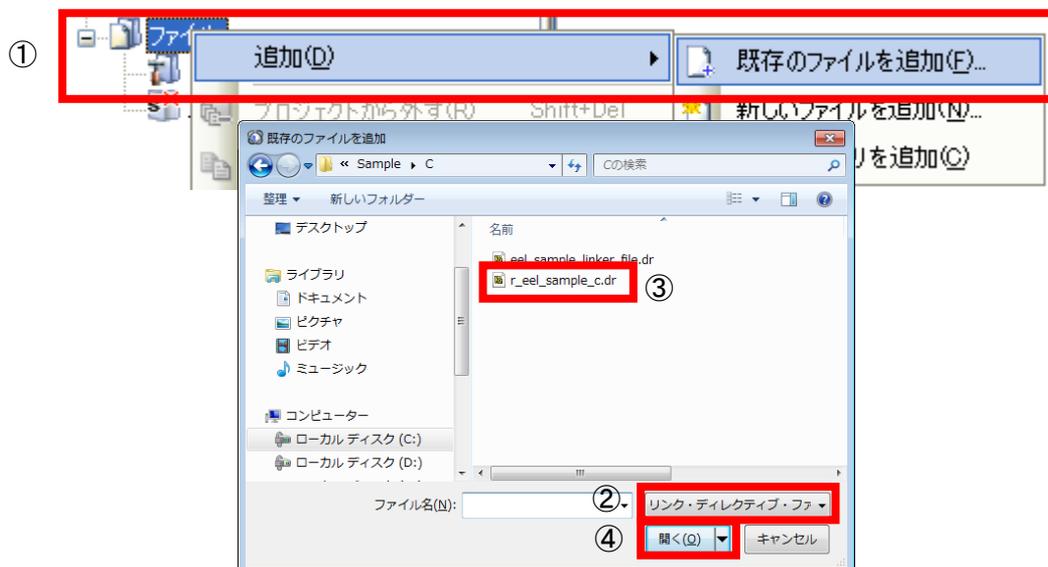


(4) リンクディレクティブの設定 (CA78K0R コンパイラ使用時のみ)

CS+の[プロジェクト・ツリー]にある、[ファイル]を右クリックして表示されるリストから[追加]を選択し、[既存のファイルを追加]をクリックすると、[既存のファイルを追加]画面が表示されます (図 6-4)。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると、ファイルの種類の一覧が表示されるので、その中にある[リンクディレクティブファイル(*.dr;*.dir)]を選択し、リンクディレクティブファイル名のユーザファイル(サンプルの場合は r_eel_sample_c.dr 注)を登録してください。

図 6-4 リンクディレクティブファイルの登録



注. 添付のサンプルディレクティブファイルを使用する場合は編集・修正が必要となる場合があります。

(5) ビルド

CS+のメニュー [ビルド] → [ビルド・プロジェクト] をクリックして、ビルドを実行します。

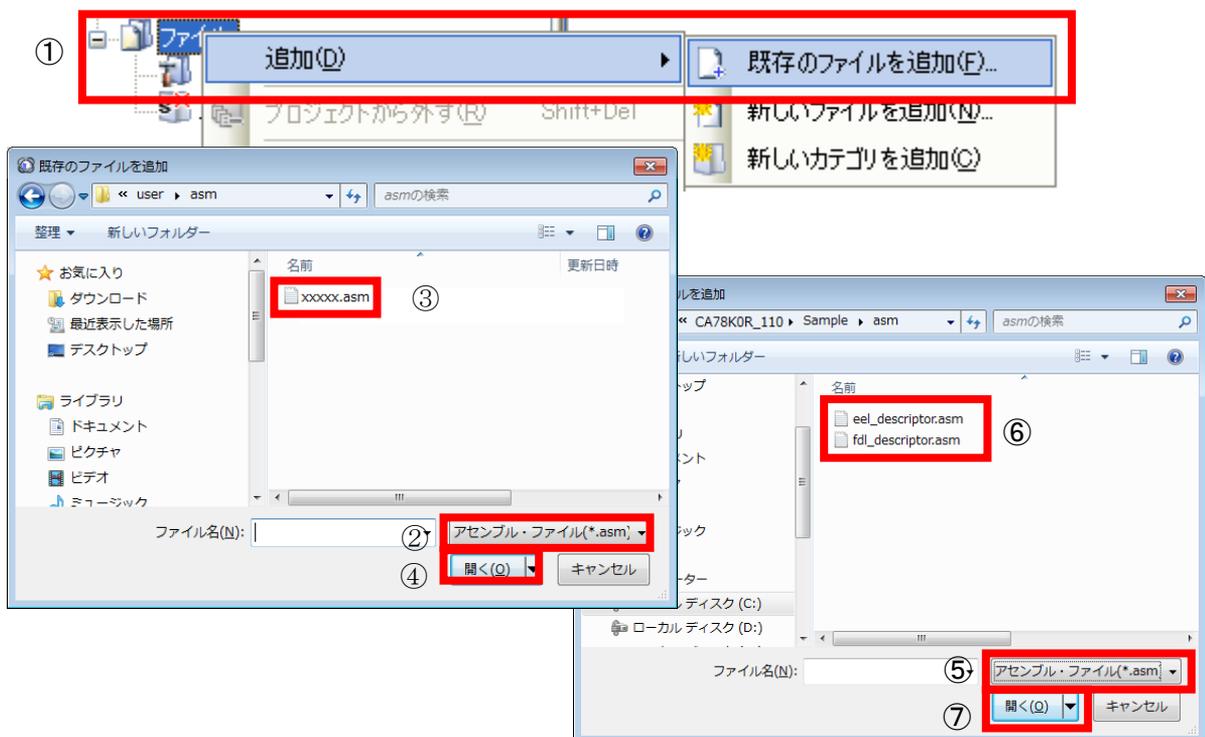
6.2.2 アセンブリ言語の場合

(1) プロジェクトの作成とソースファイルの設定

CS+でプロジェクトを作成し、表示された画面の左側にある[プロジェクト・ツリー]から、[ファイル] を右クリックして表示されるリストの[追加]を選択し、[既存のファイルを追加]をクリックすると、[既存のファイルを追加]画面が表示されます（図 6-5）。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると、ファイルの種類の一覧が表示されるので、その中にある[アセンブルファイル(*.asm)]を選択し、ソースファイルとしてユーザプログラムのファイル、及び EEPROM エミュレーション・ライブラリ、データ・フラッシュ・ライブラリのディスクリプタファイル（eel_descriptor.asm, fdl_descriptor.asm）を登録してください。

図 6-5 ユーザプログラムファイルの登録

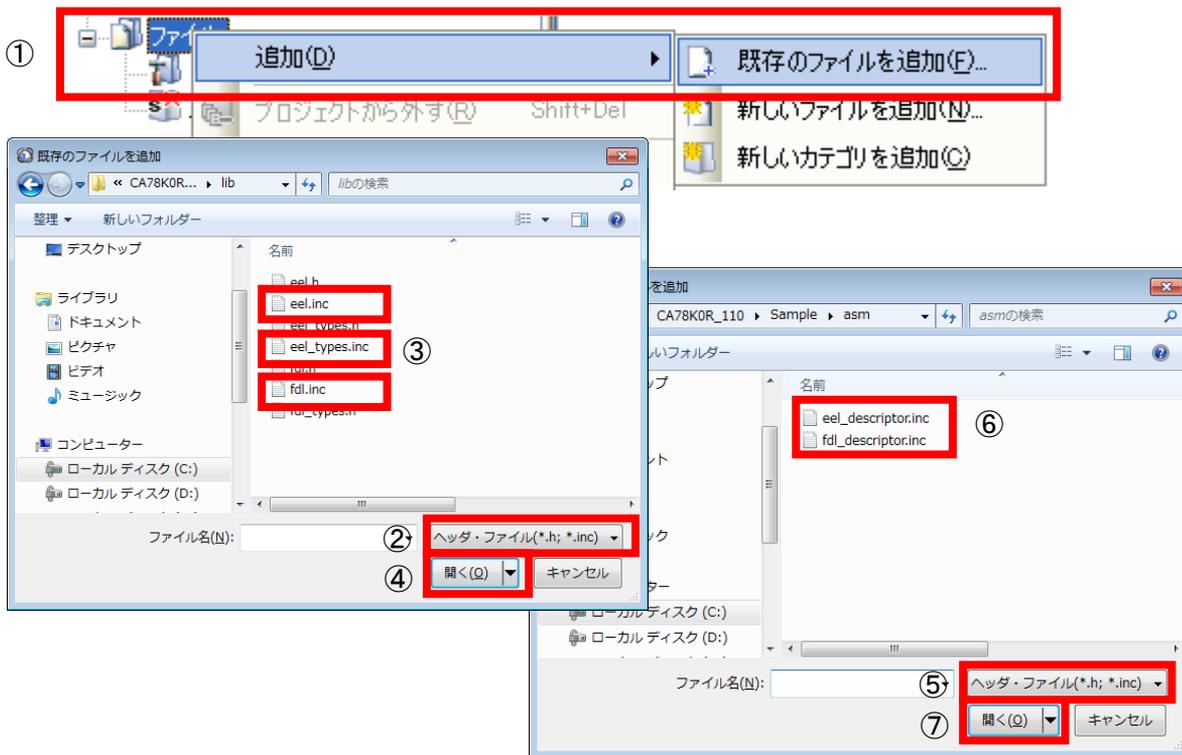


(2) インクルードファイルの設定

CS+の[プロジェクト・ツリー]にある、[ファイル] を右クリックして表示されるリストから[追加]を選択し、[既存のファイルを追加]をクリックすると、[既存のファイルを追加]画面が表示されます（図 6-6）。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると、ファイルの種類の一覧が表示されるので、その中にある[ヘッダファイル(*.h; *.inc)]を選択し、EEPROM エミュレーション・ライブラリ、データ・フラッシュ・ライブラリのヘッダファイル、及びディスクリプタ用ヘッダファイル（eel.inc, eel_types.inc, fdl.inc, eel_descriptor.inc, fdl_descriptor.inc）を登録してください。

図 6-6 インクルードファイルの登録

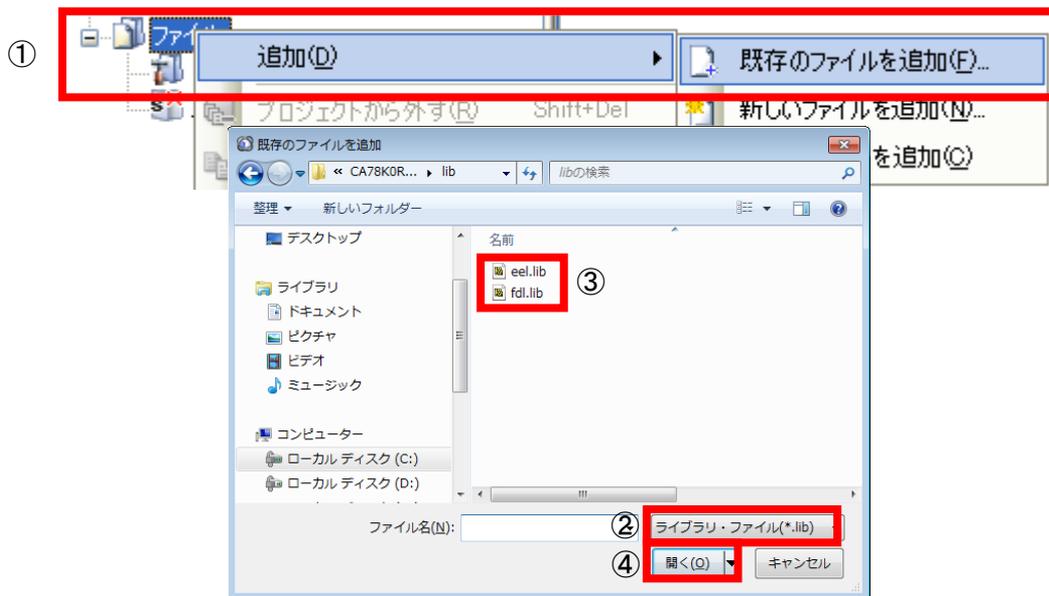


(3) ライブラリファイルの設定

CS+の[プロジェクト・ツリー]にある、[ファイル]を右クリックして表示されるリストから[追加]を選択し、[既存のファイルを追加]をクリックすると、[既存のファイルを追加]画面が表示されます（図 6-7）。

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると、ファイルの種類の一覧が表示されるので、その中にある[ライブラリファイル(*.lib)]を選択し、EEPROM エミュレーション・ライブラリ、データ・フラッシュ・ライブラリ (eel.lib, fdl.lib) のファイルを登録してください。

図 6-7 ライブラリファイルの登録

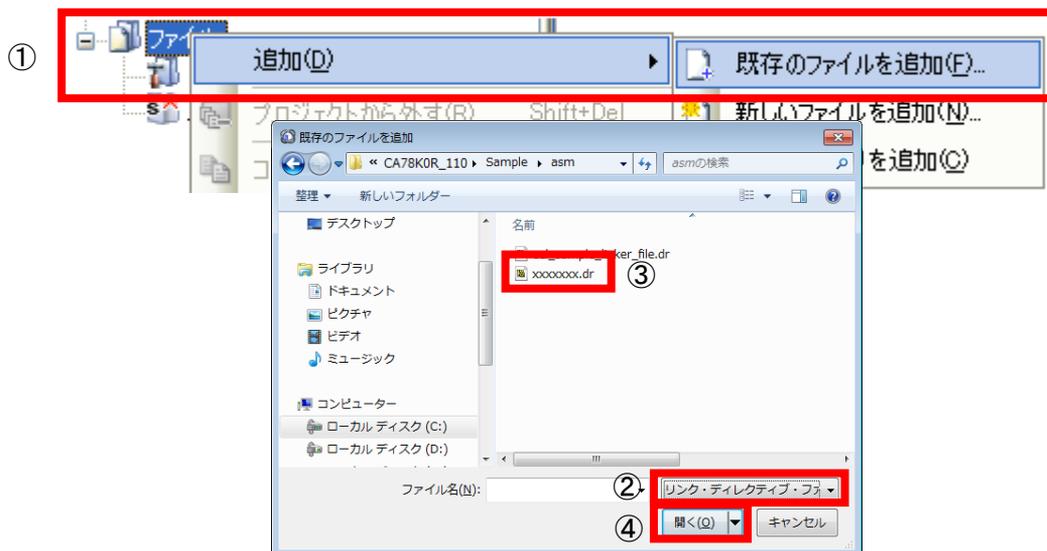


(4) リンクディレクティブの設定 (CA78K0R コンパイラ使用時のみ)

CS+の[プロジェクト・ツリー]にある、[ファイル]を右クリックして表示されるリストから[追加]を選択し、[既存のファイルを追加]をクリックすると、[既存のファイルを追加]画面が表示されます。(図 6-8)

次に画面中にある[ファイルの種類]のプルダウンメニューをクリックすると、ファイルの種類の一覧が表示されるので、その中にある[リンクディレクティブファイル(*.dr;*.dir)]を選択し、リンクディレクティブファイル名のユーザファイルを登録してください。

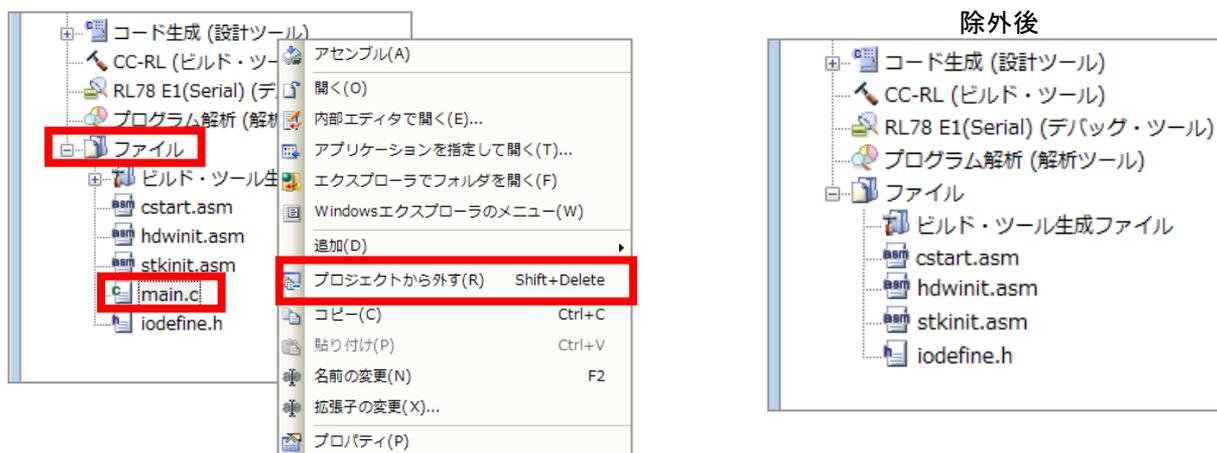
図 6-8 リンクディレクティブファイルの登録



(5) 自動生成されたファイルの除外(CC-RL コンパイラ使用時のみ)

CC-RL コンパイラ用 CS+では、[プロジェクト・ツリー]の[ファイル]にいくつかの自動生成されるファイルがあります。この中で、“main.c”の処理は、既に EEPROM エミュレーション・ライブラリのサンプルプログラムに含まれています。その為、このファイルをプロジェクトから除外する必要があります。

図 6-9 自動生成ファイルの除外



(6) ビルド

CS+のメニュー [ビルド] → [ビルド・プロジェクト] をクリックして、ビルドを実行します。

6.3 ビルド時の注意事項

6.3.1 CA78K0Rコンパイラ使用時

(1) オンチップ・デバッグ機能使用時

CS+でオンチップ・デバッグ機能を有効にした後、プログラムのビルドを行うと、以下のようなエラーが発生する場合があります。

```
RA78K0R error E3212: Default segment can't allocate to memory - ignored  
Segment '??OCDROM' at xxxxxH-200H
```

このエラーは、オンチップ・デバッグ機能で使用するモニター領域(OCDROM)用のセグメント配置ができないために発生していますので、エラーを回避するためには、プロジェクトに組み込んでいるリンクディレクティブファイル (*.dr)に、以下のような内容を追記し、セグメント配置用の領域を別途用意してください。

```
MEMORY OCD_ROM : ( 0xxxxxH, 00200H )
```

備考 1. xxxxx : エラーが発生した箇所の先頭アドレス

2. 領域名「OCD_ROM」は参考例としての表記となります。

6.3.2 CC-RLコンパイラ使用時

(1) オンチップ・デバッグ機能使用時

CS+でオンチップ・デバッグ機能を有効にした後、プログラムのビルドを行うと、以下のようなエラーが発生する場合があります。

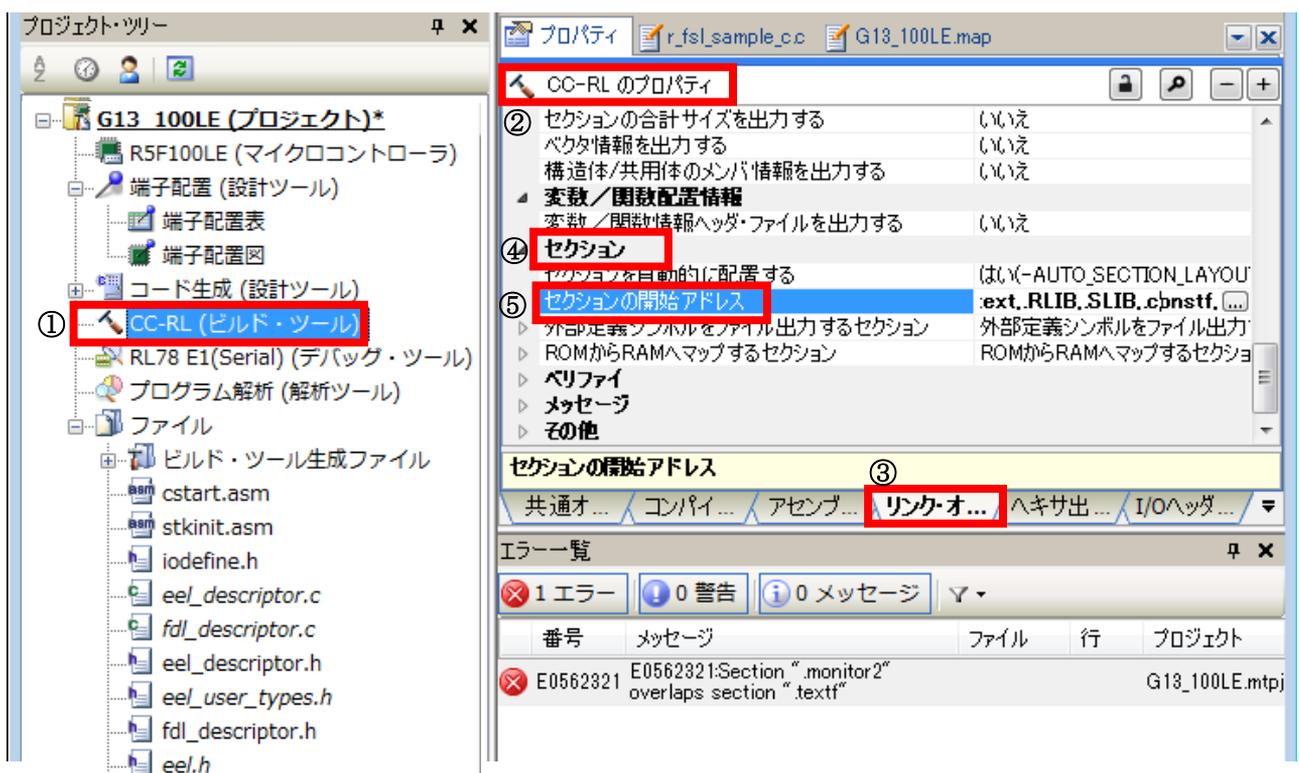
E0562321:Section ".monitor2" overlaps section "xxxxx"

このエラーは、オンチップ・デバッグ機能で使用するモニター領域(OCDROM)用のセクション配置ができないために発生しています。エラーを回避するためには、CS+の[プロジェクト・ツリー]から①[CC-RL(ビルド・ツール)]を選択、右クリックで表示されるプロパティを選択し、表示された②"CC-RLのプロパティ"の③"リンク・オプション"タブを選択します。④"セクション"項目内の⑤"セクションの開始アドレス"で、オンチップ・デバッグモニター用のセクション配置領域(monitor2 : R5F100LE では、初期アドレス 0xFE00-0xFFFF)に、他のセクションが重複しないよう配置を修正してください。(図 6-10)

セクションの設定の詳細については、CC-RL コンパイラ ユーザーズマニュアルをご確認ください。

備考 1. xxxxx : セクション名

図 6-10 セクション配置の変更



第7章 デバッグ方法

IECUBE、またはオンチップ・デバッグ・エミュレータ E1、及び E20 を使用してデバッグを行う場合につきましては、以下の資料を参照してください。

タイトル
CubeSuite+ 統合開発環境 ユーザーズマニュアル RL78 デバッグ編[CS+ for CA,CX] ^注
CS+ 統合開発環境 ユーザーズマニュアル RL78 デバッグ・ツール編[CS+ for CC] ^注

注. このドキュメントは、弊社HPの「統合開発環境 CS+ (旧 CubeSuite+)」のページからダウンロードしてください。

7.1 デバッグ時の注意事項

EEPROM エミュレーション・ライブラリ Pack02 を使用してオンチップ・デバッグ・エミュレータ E1、もしくは E20 を使用する場合の注意事項

- (1) CubeSuite+ Ver.1.01 より前のバージョンで、オンチップ・デバッグ・エミュレータ E1、もしくは E20 を使用して EEPROM エミュレーション・ライブラリ Pack02 のコマンドを実行した場合、シーケンサが完了した事を確認するまでブレイクを実行しないで下さい。シーケンサが正常に動作できなくなります。
- (2) フラッシュ・ライブラリのデバッグはシミュレータでは実行できません。

第8章 サンプルプログラム

添付のサンプルプログラム(r_eel_sample_c.c)は、R5F100LEA(RL78/G13)を対象に QB-R5F100LE-TB ボードで EEPROM エミュレーション・ライブラリ Pack02 の使用方法を簡単に確認することが可能なように用意しているプログラムです。あくまで参考例となりますので、必ずサンプルプログラム通りに作成する必要があるわけではありません。簡易的な動作確認用のプログラムとしてご使用ください。

また、CA78K0R コンパイラ用サンプルプログラムのリンクディレクティブファイル(r_eel_sample_c.dr)は、サンプルプログラムで使用するスタックやデータバッファ等を配置禁止領域^{注1}に配置しないように指定する事を目的としています。サンプルプログラムを使用する場合は、こちらのファイルも一緒に組み込んでください。^{注2}

CC-RL コンパイラ用サンプルプログラムには、リンクディレクティブファイル(r_eel_sample_c.dr)は不要ですが、CS+の画面上で、サンプルプログラムで使用するスタックやデータバッファ等を配置禁止領域^{注1}に配置しないように [リンク・オプション]で、[セクション]配置を設定する必要があります。

注 1. 詳細については、ユーザズマニュアルの「6.2 ソフトウェア・リソース」を参照してください。

- ご使用の環境やプログラムの変更によっては使用中のデータが意図しない領域へ配置される場合があります。実行モジュール作成後はマップファイルを確認し、プログラムやデータの配置状態を必ず確認してください。また、各コードやデータの定義方法や配置条件等については、CS+のユーザズマニュアルを参照してください。

8.1 サンプルプログラムの初期設定

サンプルプログラムは以下の初期設定で動作します。変更が必要な場合は、サンプルプログラムを修正してください。

- ・ CPU の動作周波数 : 高速オンチップ・オシレータ 32MHz
- ・ 電圧モード : フルスピード・モード

8.2 オプション・バイトとオンチップ・デバッグの設定について

オンチップ・デバッグを行う場合は「オンチップ・デバッグの許可/禁止をリンク・オプションで設定する」を「はい」に設定し、「オンチップ・デバッグ・オプション・バイト制御値」を「84」に指定してください。CC-RLコンパイラ用では、「デバッグ・モニタ領域を設定する」を「はい」に指定してください。

また、サンプルプログラムは高速オンチップ・オシレータを32MHzに設定する事で正常に動作します。リンク・オプションから、「ユーザ・オプション・バイトを設定する」を「はい」に設定した後、ユーザ・オプション・バイト値を「xxxxE8」に指定し、高速オンチップ・オシレータを32MHzに設定してください。

図8-1(a) オプション・バイトの設定(CA78K0Rコンパイラ)

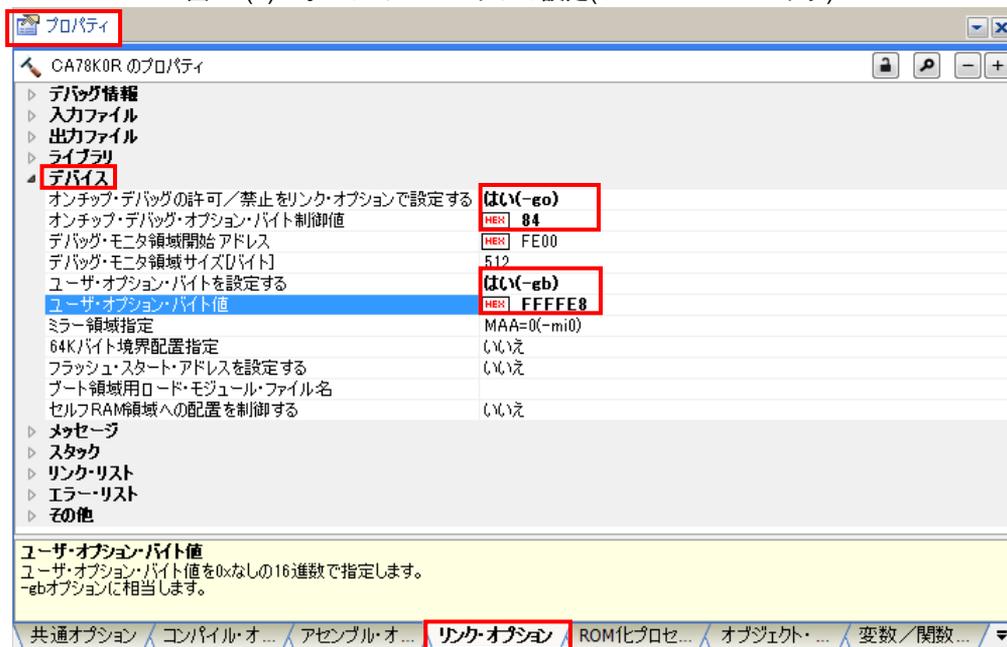
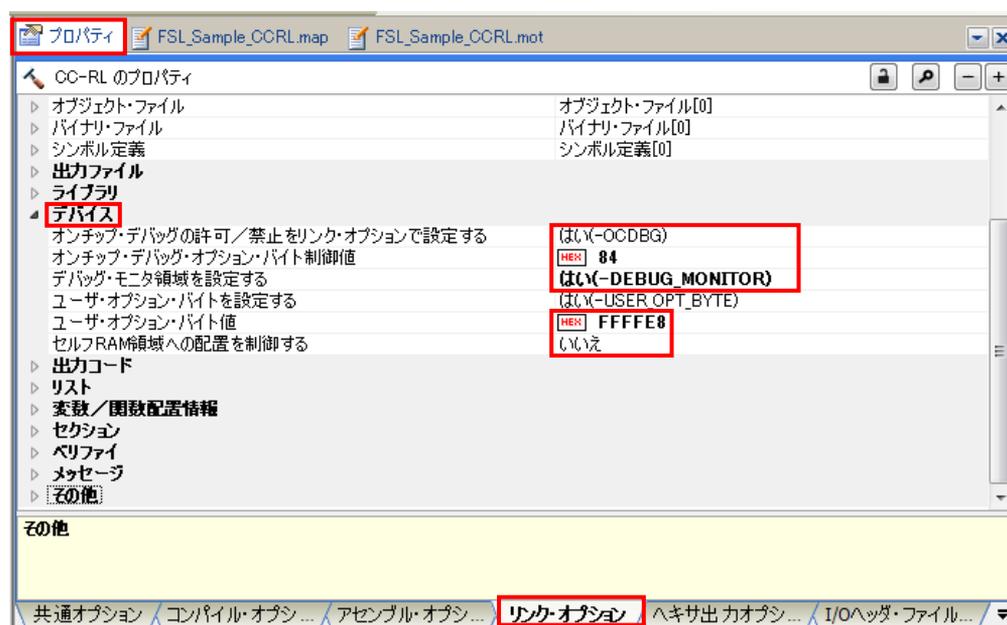


図8-1(b) オプション・バイトの設定(CC-RLコンパイラ)



8.3 内蔵RAM領域の定義

8.3.1 CA78K0Rコンパイラ使用時

CA78K0R コンパイラ使用時の初期状態では、内蔵 RAM 領域については全域が"RAM"という名称の領域として自動的に定義されています。特にリンクディレクティブファイル等で特に指定しない場合、内蔵 RAM 領域については全域が"RAM"という名称の領域として自動的に定義され、特に指定しない限り、スタック等がこの領域に配置^注される事になります。この場合、フラッシュ・ライブラリの使用禁止領域(セルフ RAM、FFE20H-FFEDFH の領域)にスタックやデータバッファ等が配置されてしまい、プログラムが正常に動作できなくなることがあります。

添付されているサンプル用リンクディレクティブファイルでは、解決策の一つとして"RAM"名の領域をフラッシュ・ライブラリの使用禁止領域を含めないように再定義し、スタック等が使用禁止領域に配置されないようにしています。

```
MEMORY RAM      : (0FF080H, 000DA0H)
```

上記は、"RAM"名の領域を、FF080H から DA0H バイトのサイズの領域 (FF080H-FFE1FH) ^注に再定義し、フラッシュ・ライブラリの使用禁止領域が"RAM"名の領域に含まれないようにしています。

ただし、この設定だけでは「FFE20H-FFEDFH」の領域が他の用途でも使用できなくなるため、別途以下の定義を追加する必要があります。この領域の名称に関しては特に制限等はありません。

```
MEMORY SADDR_RAM : (0FFE20H, 0001E0H)
```

また、セルフ RAM 領域がある場合は、その範囲を以下のように"SELFRAM"名の領域として定義する事により、この領域に対し、自動的に変数等が配置されないように制限する事が可能です。

```
MEMORY SELFRAM  : (0FEF00H, 000180H)
```

以下に、RL78/G13(RAM 4KB/ROM 64KB 製品)の場合の設定例を記載します。

(設定例では、RAM_SADDR に汎用レジスタ、SFR 領域を含めていますが、含めずに指定することも可能です)

```

: -----
: Define new memory entry for Self-RAM
: -----
MEMORY SELFRAM  : ( 0FEF00H, 000180H ) ← セルフ RAM 領域の定義
: -----
: Redefined default data segment RAM
: -----
MEMORY RAM      : ( 0FF080H, 000DA0H ) ← 標準的に使用する RAM 領域の定義
: -----
: Define new memory entry for saddr area
: -----
MEMORY RAM_SADDR : ( 0FFE20H, 0001E0H ) ← FFE20H-FFFFFH 領域の定義

```

注. CA78K0Rのリンクでは、配置先が指定されないデータ(セグメント・タイプDSEGおよびBSEG)はそのデータの再配置属性に従い、内蔵RAM領域に配置されます。そのため、状況によっては特定のデータが"RAM"名の領域に配置されない場合があります。

各データの定義や配置方法等の詳細についてはCS+のユーザーズマニュアルを参照してください。

また、ビルド時に生成されるマップファイル(*.map)を必ず参照し、各データの配置状態を確認してください。

8.3.2 CC-RLコンパイラ使用時

(1) インクルード・パスの追加

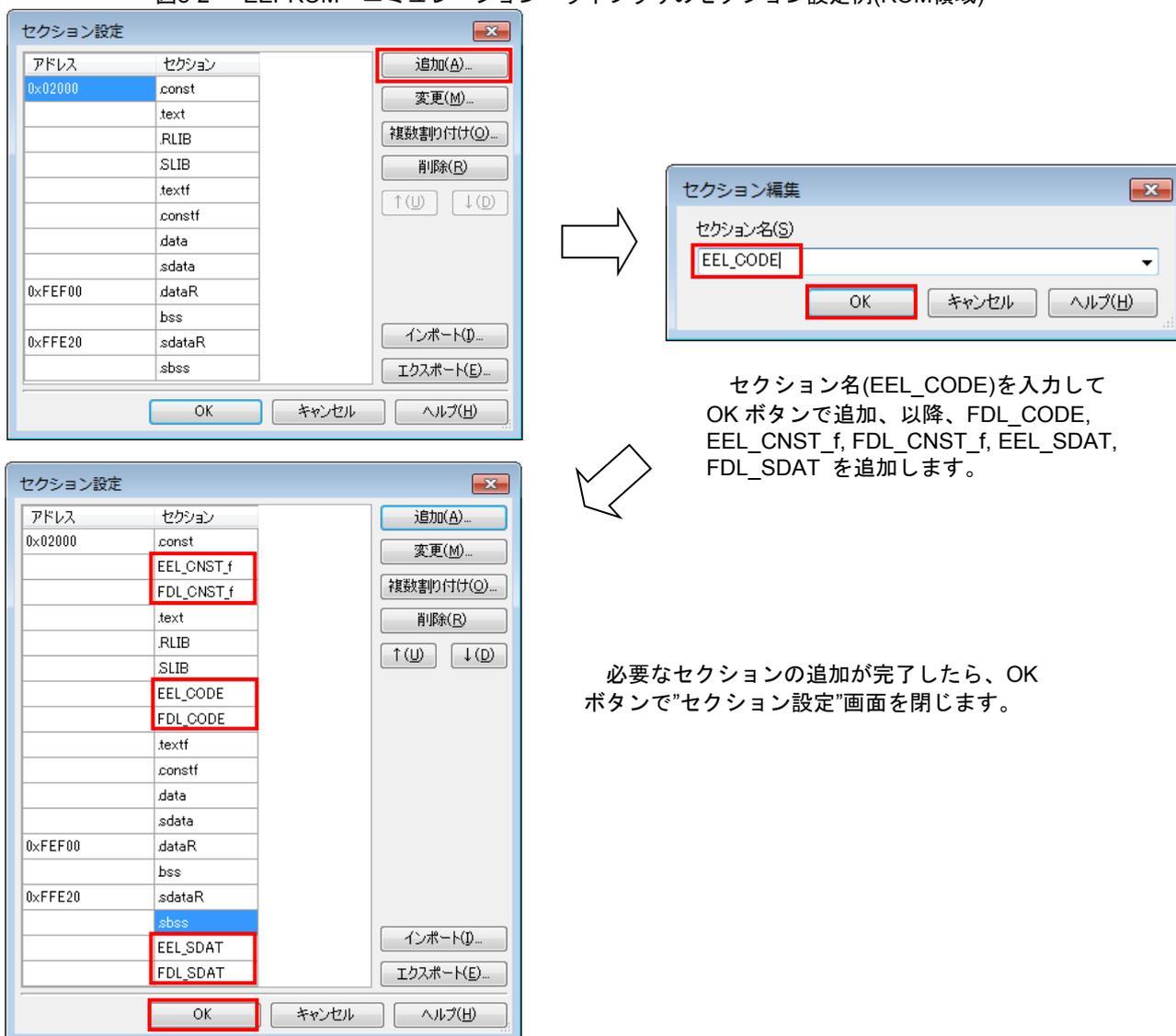
CC-RL コンパイラ用 CS+は、初期状態ではインクルード・パスは設定されていません。EEPROM エミュレーション・ライブラリで使用するヘッダファイルのインクルード・パスを追加する必要があります。EEPROM エミュレーション・ライブラリで使用するヘッダファイル

は、"eel_user_types.h"、"eel.h"、"fdl.h"、"eel_types.h"、"fdl_types.h"、"eel_descriptor.h"、"fdl_descriptor.h"、及び CS+が自動生成する"iodefine.h"です。[コンパイル・オプション]-[プリプロセス]-[追加のインクルード・パス]で、各ファイルが存在するインクルード・パスを追加してください。

(2) セクションの定義

CC-RL コンパイラ用 CS+ 使用時は、ROM 領域、RAM 領域で使用する各セクションを定義する必要があります。セクションの定義は、CS+の画面の[リンク・オプション]-[セクション]で指定することができます。[セクションを自動的に配置する]を"いいえ"に設定し、[セクションの開始アドレス]の設定画面を開き、ROM 領域に EEPROM エミュレーション・ライブラリのセクション(ここでは、本サンプル動作に必要なセクション、EEL_CODE, FDL_CODE, EEL_CNST_f, FDL_CNST_f, EEL_SDAT, FDL_SDAT)を追加します。(図 8-2)

図8-2 EEPROM・エミュレーション・ライブラリのセクション設定例(ROM領域)



(3) セルフ RAM 領域の確保

CC-RL コンパイラ用 CS+ でセクション設定の初期状態では、ユーザ RAM 領域が内蔵 RAM 領域の先頭(サンプルプログラム対象の R5F100LEA では 0xFEFE00)から確保されています。しかし、R5F100LEA では、EEPROM エミュレーション・ライブラリがセルフ RAM 領域として 0xFEFE00-0xFF07F を使用する為、この領域を避けてユーザ RAM 領域を設定する必要があります。ここでは、アドレス 0xFEFE00 から設定されているユーザデータの先頭アドレスを 0xFF080 に変更する例を示します。

図8-3 ユーザRAM領域範囲の変更例(RAM領域)

図8-3は、ユーザRAM領域範囲の変更例を示しています。図には、セクション設定画面とセクションのアドレス設定画面のスクリーンショットが示されています。

セクション設定画面の初期状態では、アドレス 0xFEFE00 の dataR セクションが選択されています。

セクションのアドレス設定画面で、アドレス(A)を FEF00 から FF080 に変更します。

変更後、OK ボタンでセクション設定画面を閉じます。

注意) セクションの設定で[セクションを自動的に配置する]を一度”いいえ”にして、ユーザ RAM 領域範囲の変更をした後に、”はい”に設定することで、ユーザが定義したセクションも含め、再自動配置されますが、自動配置では、ユーザが定義していない領域に対してもセクションを配置する可能性があり、意図していなかった領域にデータが配置される可能性があります。必ず、EEPROM エミュレーション・ライブラリで使用するソフトウェアリソース(特に RAM 上データ)が、配置可能領域に配置されていることをマップファイル(*.map)等で確認してください。

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・システムの設計において、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て、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。
2. 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。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
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、産業用ロボット等
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、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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いません。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、社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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て、当社は、一切その責任を負いません。
9. 本資料に記載されている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・システムに使用することはできません。また、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に使用しないでください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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禁じます。

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 エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■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/contact/>